

# *Projet ECOO*

*Environnements et COOpération*

*Nancy*

THÈME 3A



*R*apport  
*d'Activité*

2000



## Table des matières

<b>1</b>	<b>Composition de l'équipe</b>	<b>3</b>
<b>2</b>	<b>Présentation et objectifs généraux</b>	<b>4</b>
<b>3</b>	<b>Fondements scientifiques</b>	<b>6</b>
3.1	Modèles de coordination de tâches . . . . .	6
3.1.1	Généralités . . . . .	6
3.1.2	ECOO et la coordination de tâches . . . . .	6
3.2	Transactions coopératives . . . . .	7
3.2.1	Généralités . . . . .	7
3.2.2	ECOO et les transactions coopératives . . . . .	7
3.3	Gestion d'espace d'objets partagés . . . . .	7
3.3.1	Généralités . . . . .	7
3.3.2	ECOO et la gestion d'espace d'objets partagés . . . . .	8
3.4	Analyse des usages . . . . .	8
3.4.1	Généralités . . . . .	8
3.4.2	ECOO et les usages . . . . .	8
3.5	Analyse globale et évaluation partielle . . . . .	8
3.5.1	Généralités . . . . .	8
3.5.2	Adaptation au type du receveur . . . . .	9
<b>4</b>	<b>Domaines d'applications</b>	<b>9</b>
<b>5</b>	<b>Logiciels</b>	<b>9</b>
5.1	Introduction . . . . .	9
5.2	TuaMotu . . . . .	10
5.3	Corvette . . . . .	10
5.4	DOTS . . . . .	10
5.5	Discoo . . . . .	11
5.6	SmallEiffel: the GNU Eiffel Compiler . . . . .	11
<b>6</b>	<b>Résultats nouveaux</b>	<b>12</b>
6.1	Coordination explicite . . . . .	12
6.1.1	Introduction . . . . .	12
6.1.2	Interprétation flexible de procédés – Workflows coopératifs . . . . .	12
6.1.3	Modélisation de procédés inter-services et/ou organisations . . . . .	13
6.2	Coordination implicite . . . . .	14
6.2.1	Introduction . . . . .	14
6.2.2	Métrique de Divergence . . . . .	14
6.3	Collaboration . . . . .	16
6.3.1	Argumentation . . . . .	16
6.3.2	Négociation . . . . .	17
6.4	Relations entre différentes dimensions de la coopération . . . . .	18

---

6.4.1	Introduction . . . . .	18
6.4.2	Convergence synchrone/asynchrone dans les équipes virtuelles . . . . .	18
6.4.3	Description des contraintes d'interaction . . . . .	19
6.5	Le Projet SmallEiffel . . . . .	20
6.6	Systèmes Formels à Prototypes pour le Langages à Objets . . . . .	21
<b>7</b>	<b>Contrats industriels (nationaux, européens et internationaux)</b>	<b>21</b>
7.1	AEE : Architecture pour l'Electronique Embarquée . . . . .	21
7.2	Bio-Expert . . . . .	22
7.3	Cacao . . . . .	22
7.4	Corvette . . . . .	23
7.5	Définition d'un compilateur sous contraintes de ressource pour FGPA – Collaboration avec Microlor . . . . .	23
<b>8</b>	<b>Actions régionales, nationales et internationales</b>	<b>24</b>
8.1	Actions régionales . . . . .	24
8.2	Actions nationales . . . . .	24
8.3	Actions Internationales . . . . .	24
8.3.1	Thèse en co-tutelle . . . . .	24
8.3.2	Participation à des comités de programme . . . . .	24
8.3.3	Participation à des conférences . . . . .	25
8.4	Visites, et invitations de chercheurs . . . . .	25
8.4.1	Equipes et entreprises virtuelles . . . . .	25
<b>9</b>	<b>Diffusion de résultats</b>	<b>25</b>
9.1	Animation de la Communauté scientifique . . . . .	25
9.2	Enseignement . . . . .	25
<b>10</b>	<b>Bibliographie</b>	<b>25</b>

---

*ECO*O 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, jusqu'au 01/10/00]

Antoinette Courier [Personnel CNRS, depuis le 01/10/00]

### 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 01/09/1998 jusqu'au 30/08/2000), 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 01/09/1999 jusqu'au 30/08/2001)]

François Charoy [Maître de conférences, Université Nancy 2, en délégation à l'INRIA (depuis le 01/09/2000 jusqu'au 30/08/2002)]

Luigi Liquori [Maître de conférences, INPL]

Jacques Lonchamp [Professeur, Université Nancy 2]

Pascal Molli [Maître de conférences, UHP, en délégation à l'INRIA (depuis le 01/09/2000 jusqu'au 30/08/2002)]

Olivier Perrin [Maître de conférences, Université Nancy 2]

Hala Skaf [Maître de conférences, UHP]

**Chercheurs doctorants**

Karim Baïna [boursier MENRT]

Jacques Besse [Thèse CNAM, Fonds de formation]

Julia Bitcheva [boursière CNET]

Abdelmajid Bouazza [boursier UHP]

Christophe Bouthier [boursier BDI/Région]

Daniela Grigori [boursière CNET]

Olivier Malcurat [boursier CNET, thèse en architecture au CRAI]

Fabrice Muller [Thèse CNAM, Fonds de formation]

Benoît Sonntag [boursier CCH]

Samir Tata [boursier du gouvernement français, ATER Université Nancy 2 depuis le 01/09/2000]

Olivier Zendra [ATER UHP, post-doc INRIA à l'Université de Mac Gill depuis le 01/12/2000 jusqu'au 30/11/2001]

**Ingénieurs experts**

Philippe Canalda [depuis le 01/09/1999 jusqu'au 30/08/2000]

Jean-Marc Humbert [depuis le 15/08/2000 jusqu'au 15/03/2001]

Marc Patten [depuis le 01/09/2000 jusqu'au 30/08/2001, sur poste Accueil Jeunes]

**Professeur invité**

Pradeep Ray [septembre 2000]

**2 Présentation et objectifs généraux**

**Mots clés** : coopération, coordination, communication, équipe/entreprise virtuelle, Internet, interopérabilité, intertravaillabilité, techniques objets, transaction.

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 spécialisé ou un réseau général 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 dans les applications de type co-conception et/ou ingénierie simultanée se déployant sur Internet. Il est organisé autour de quatre axes principaux :

- la coordination explicite basée sur la possibilité, de formaliser des procédés décrivant explicitement le travail à réaliser, et de les contrôler sur le terrain,
- la coordination implicite basée sur l'hypothèse qu'une bonne conscience de ce que font les autres (« awareness » en anglais), par la communication qu'elle induit, peut assurer une auto-coordination suffisante,
- une étude des procédés collaboratifs de co-décision (résolution collective de question, argumentation, négociation ...).
- l'étude des relations entre ces différentes dimensions.

**Coordination explicite.** L'hypothèse fondamentale de cette approche est que l'on est capable de décrire par un modèle de procédé «enactable» (partiellement exécutable) les activités et l'enchaînement « subtil » des activités caractéristiques des applications créatives qui nous intéressent. Si nous pensons que les modèles de workflows actuels [WFM95] sont un bon point de départ, nous savons aussi qu'ils ne fournissent pas la flexibilité nécessaire aux applications qui nous intéressent. On s'intéresse donc à l'utilisation au mieux des modèles de workflow existants et surtout à leur évolution vers plus de souplesse pour mieux répondre aux besoins des applications de co-conception et de co-ingénierie (cf. 6.1).

**Coordination implicite.** Si la modélisation des procédés peut être un élément important dans la mise en œuvre de la coopération dans les équipes et entreprises virtuelles, il est irréaliste de tout vouloir décrire et contrôler par ce biais, de tout vouloir formaliser. Une approche originale, et nous pensons complémentaire, développée dans le domaine du travail coopératif (CSCW) est basée sur le constat suivant: si les membres d'une équipe virtuelle sont maintenus conscients, de manière pertinente, de l'état et de l'activité du projet, alors ils seront amenés à communiquer et à s'auto-coordonner. C'est cette démarche que nous appelons « coordination implicite », ou encore, coordination basée sur la conscience de groupe (traduction de « awareness ») (cf 6.2).

**Procédés collaboratifs de co-décision** Nous nous intéressons à la formalisation et l'implantation de modèles génériques de négociation et d'argumentation au sein des applications coopératives. L'objectif est (cf. 6.3) :

- de développer un environnement générique de support aux tâches en collaboration à forte composante décisionnelle, où l'argumentation sert à préparer les prises de décision,

---

[WFM95] WPMC, « Workflow Management Coalition », *rapport de recherche*, AIIM, 1995.

- de construire un modèle de négociation qui soit indépendant de tout domaine d'application particulier, utilisable 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.

**Etude des relations entre les différentes dimensions de la coopération** Comme nous venons de l'introduire, nous pensons que les partenaires d'une équipe ou d'une entreprise virtuelle se coordonnent en combinant coordination explicite et coordination implicite. Ces dimensions ne sont pas orthogonales et l'étude de leurs relations est nécessaire. Elle se décline de plusieurs points de vue : coopération asynchrone vs. coopération synchrone, coordination formalisée vs. coopération non formalisée ... En particulier, dans ce contexte, l'étude de la bascule d'un mode vers un autre nous semble tout particulièrement pertinente. Plus généralement, l'étude des relations entre coordination et communication est un axe de recherche neuf dans la communauté scientifique et un verrou difficile pour les applications coopératives qui mêlent travail asynchrone et travail synchrone, procédés formalisés et procédés non formalisés, interactions indirectes et interactions directes (cf. 6.4).

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 importante 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. En particulier, elle a produit le compilateur SmallEiffel sélectionné comme compilateur Eiffel de l'environnement GNU de la Free Software Foundation. SmallEiffel est distribué à des milliers d'exemplaires.

## 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 <sup>[WFM95]</sup> sont trop rigides pour bien modéliser la subtilité des interactions caractéristiques des applications coopératives.

#### 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

---

[WFM95] WfMC, « Workflow Management Coalition », *rapport de recherche*, AIIM, 1995.



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

---

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

simultanément et par conséquent insuffisamment intégrées. Soulignons aussi que les meilleures solutions sont souvent trop coûteuses pour bon nombre des applications visées.

### 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 CSCW<sup>1</sup>.

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.

---

1. CSCW : Computer-Supported Cooperative Work

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 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 effectivementinstanciées (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 Cocoa (cf. 7.3)), les procédés logiciels (collaboration avec Hitachi (cf. 7.4), projet AEE (cf. 7.1)), la biologie (Bio-Expert (cf. 7.2)).

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. TuaMotu n'est pas encore diffusé, mais est

utilisé dans plusieurs contrats industriels.

## 5.2 TuaMotu

**Participants** : Pascal Molli (responsable), Abdelmajid Bouazza, Christophe Bouthier, François Charoy, Saad El-Hadri, Claude Godart, Marc Patten, Hala Skaf.

Le prototype TuaMotu permet à une équipe virtuelle de partager des données tout en maintenant une conscience de groupe. Il se situe dans la lignée d'outils comme BSCW, TeamScope ou SourceForge.

Il intègre un gestionnaire de données partagées avec des systèmes de conscience de groupe. Le système Motu se démarque par une conscience de groupe plus riche : sa perception des actions des différents utilisateurs est plus fine et plus active. « Tuamotu » est une infrastructure logicielle pour le support d'environnements de travail coopératif en réseau de grande taille (internet). Ce support se fait principalement au travers du partage d'un système de fichiers versionnés. Écrit en Java, il est pour l'instant à l'état 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, gérant des branches et les espaces de travail des clients. Le serveur intègre un mécanisme de gestion des droits d'accès ainsi qu'un mécanisme de notification d'événements aux clients. Le protocole des COO-transactions ainsi qu'une couche workflow coopératif est en cours de développement et devrait aboutir à court terme.

Tuamotu est utilisé dans les projets industriels Cocoa et Corvette.

Pour plus d'information : <http://motu.sourceforge.net/>

## 5.3 Corvette

**Participants** : François Charoy (responsable), Karim Baina, Claude Godart, Daniela Grigori, Saad el Hadri, Hala Skaf.

Le logiciel Corvette est développé dans le cadre du projet Corvette (cf. 7.4). Il s'agit d'un système de workflow coopératif obtenu en couplant le logiciel de workflow WorkCoordinator de chez Hitachi au modèle de transactions coopératives des TuaMotu. L'objectif est de finir le prototype pour mars 2001.

## 5.4 DOTS

**Participants** : Jacques Lonchamp (responsable), Fabrice Muller.

DOTS (DOTS pour "Decision Oriented Task Support") est un environnemnt qui met en oeuvre les procédés d'aide à la prise de décisions (argumentation, résolution collective de questions ...) développés dans 6.3.1. Il est entièrement développé en java. Les dernières évolutions concernent l'intégration de réunions virtuelles synchrones aux tâches asynchrones. Dans cette approche, les participants peuvent choisir de manière opportuniste de passer du mode asynchrone à des sessions en temps réel, en conservant le contexte du travail en cours. Le modèle de tâche peut prescrire une démarche pour organiser les sessions temps réel. Un tableau de

bord pour chaque question débattue permet d'aider à la décision de basculement au mode synchrone, souvent motivé par une difficulté à trouver un consensus entre les participants.

Pour plus d'information : <http://www.loria.fr/equipes/ecoo/dots>

## 5.5 Discoo

**Participants** : Khalid Benali (responsable), Karim Baina.

COO [2] est un environnemnt de coopération basé sur le protocole COO qui met en œuvre la COO-sérialisabilité [1]. Après une version centralisée (qui n'est plus disponible parce que basée sur une architecture PCTE), nous en avons développé une version distribuée (Discoo). Celle-ci permet la mise en œuvre de la coopération en deux étapes. Dans la première, chaque partenaire définit avec ses voisins sa table de coopération en associant objets et patrons de coopération. Dans la seconde, les partenaires coopèrent en s'échangeant des objets. Les contrôles de cohérence se suffisent d'informations locales. DisCOO est mis en œuvre au dessus d'un bus CORBA.

<http://www.loria.fr/equipes/ecoo/discoo>

## 5.6 SmallEiffel: the GNU Eiffel Compiler

**Participants** : Dominique Colnet (responsable), 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 classique 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<sub>E</sub>X 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 liste 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 grand 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

siteSmallEiffel).

## 6 Résultats nouveaux

### 6.1 Coordination explicite

#### 6.1.1 Introduction

Quand un processus coopératif est complexe, il devient important explicite pouvant servir de guide et exécution. Cette constatation nous a gestion de flot de travail (Workflow décrire, contrôler et exécuter des processus de ces systèmes pour ce type de processus sont : la description des processus et pour des utilisateur engagés dans un activités de conception. Nous permettant de rendre plus de ces processus. La l'interprétation faite flexible et adaptable approche plus déclarative. On coordination et les enchainements de contrainst dans lequel les interactions place. La coordination y est Parallèlement à ce travail nous abordons des questions plus générales de coordination de processus concurrents. En effet, si on commence à savoir maitriser l'exécution de processus simples (de production et même de coopération), la coordination de processus entre eux représente un enjeux encore plus important dans le contexte d'entreprise virtuelles. On se rapproche ici des problématiques de compositions de services (de e-services en particulier) et donc d'interfaces entre ces services.

#### 6.1.2 Interprétation flexible de procédés – Workflows coopératifs

**Participants :** Karim Baina, François Charoy, Claude Godart, Daniela Grigori, Saad El Hadri, Olivier Perrin, Hala Skaf.

Les modèles de descriptions de processus utilisés dans les systèmes de workflow actuels ont un avantage important : ils sont simples à comprendre et facilement interprétables par des utilisateurs. Ils permettent en particulier de représenter des interactions, des synchronisations et des enchainements simples d'activités permettant d'atteindre un objectif. Le problème principal de cette représentation est que l'interprétation qui en est faite par les systèmes de workflow actuels est orientée vers les application administratives et de production. En d'autres termes, la mise en oeuvre des modèles de workflow faite par les systèmes de workflow actuels est trop rigide pour supporter les interactions subtiles telles qu'elles existent dans les applications créatives du type co-conception ou co-ingénierie.

Notre objectif est donc de développer une interprétation des modèles de workflow actuels qui supportent les activités coopératives mises en oeuvre dans les applications qui nous intéressent sans (trop) perturber les habitudes des partenaires de l'application.

**Principe d'anticipation** Le principe de cette interprétation se base sur la possibilité d'anticiper le démarrage d'une activité par rapport à son moment « normal » d'activation [16]: démarrage d'une activité avec des entrées incomplètes, des résultats provisoires, ... L'anticipation permet ainsi de rendre plus flexible l'exécution d'un procédé en conservant l'ordre de terminaison des activités.

**Workflow coopératif** Nous définissons un système de workflow coopératif comme un système de workflow mettant en oeuvre le principe d'anticipation, permettant ainsi une inter-

prétation flexible des modèles de workflow. Nous étudions une mise en oeuvre de ce concept couplant la gestion du flot de contrôle traditionnel au modèle des COO-transactions [1], qui permet la gestion cohérente d'échange de données intermédiaires (incomplètes, provisoires ...) entre activités coopérantes (à l'opposé, on peut considérer que les modèles de workflow classiques couplent la gestion du flot de contrôle à un modèle de transaction ACID).

D'autre part, nous montrons dans [16] comment les flots de contrôle peuvent être utilisés pour optimiser le protocole COO associé aux COO-transactions. Dans [15], nous montrons aussi que l'on peut mettre en oeuvre le concept d'anticipation comme un nouvel opérateur combinable aux opérateurs traditionnels de workflow.

**Expérimentation** Une mise en oeuvre de ce concept est faite dans le projet Corvette (cf. 5.3) en couplant le logiciel de workflow WorkCoordinator de Hitachi avec le modèle de transaction des TuaMotu (cf. 5.2).

### 6.1.3 Modélisation de procédés inter-services et/ou organisations

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

La modélisation de procédés inter-services inter-organisations pose des problèmes spécifiques qui nécessitent une adaptation et une évolution des modèles actuels. Parmi ces problèmes, soulignons l'hétérogénéité des modèles et des systèmes et surtout les problèmes de confidentialité et de sécurité. En particulier, les problèmes de confidentialité font qu'il est utopique de vouloir connaître dans le détail les procédés à combiner et qu'il faut en général se suffir d'un couplage faible des procédés. Soulignons aussi que les organisations peuvent être au sein de la même entreprise ou dans des entreprises différentes. Le fait qu'elles soient dans des entreprises différentes ne fait qu'amplifier ces difficultés. Au cours de cette année, nous avons essentiellement mené des expériences de modélisation de procédés dans le domaine de l'électronique embarquée et d'interopérabilité de systèmes de workflow.

**Modèle de procédé logiciel pour l'électronique embarquée** Il s'agit de développer un modèle de procédé logiciel intégrant, comme tout procédé logiciel, différents points de vue (cycle de vie du logiciel, gestion de versions et de configurations, plan qualité, capitalisation ...) concernant souvent différents services, mais aussi différentes organisations : le procédé est distribué chez différents partenaires (constructeur et équipementiers) jouant des rôles différents et ayant des objectifs différents. Cette étude se fait dans le cadre du projet AEE (cf 7.1) dans lequel nous avons fait émerger le concept central de « point de synchronisation » pour permettre une gestion flexible du procédé. L'objectif d'un point de synchronisation est de contrôler l'avancement des procédés des différents points de vue introduits ci-dessus, de décider des ajustements à réaliser et éventuellement de planifier les actions correctives à réaliser.

**Interopérabilité de systèmes de workflow.** Comme introduit dans 6.1.3, notre objectif à terme est de développer un système dans la veine des systèmes de workflows actuels mais adaptés à la nature des procédés conceptuels qui nous intéressent. Dans un contexte inter-services et inter-organisationnels, ces systèmes devront inter-opérer. Nous menons des expériences dans ce sens en nous appuyant sur les standards MIME et XML de la WfMC.

## 6.2 Coordination implicite

### 6.2.1 Introduction

Une approche originale développée dans le domaine du travail coopératif (CSCW), pour coordonner une équipe virtuelle, est basée sur le constat suivant: si les membres d'une équipe virtuelle sont maintenus conscients, de manière pertinente, de l'état et de l'activité du projet, alors ils seront amenés à communiquer et à s'auto-coordonner. C'est cette démarche que nous appelons "coordination implicite", ou encore, coordination basée sur la conscience de groupe<sup>2</sup>.

Nous avons observé que le travail d'une équipe virtuelle est rythmé par des phases de divergence suivies de phases de convergence qui se répètent à l'infini. Les membres divergent quand ils travaillent en isolation dans leurs espaces de travail respectifs et convergent lors des phases d'intégration. Aussi, une difficulté importante inhérente à ce modèle de cycle de vie est la phase de convergence. Notre idée est qu'une mesure en temps réel la divergence, dans une unité qui reste à découvrir, peut permettre de limiter la difficulté de la convergence en limitant la divergence.

La Coo-sérialisabilité[1] est un résultat important de l'équipe ECOO. Elle tolère les phases de divergence, donne des garde-fous pour converger et n'est vérifiée en fait qu'au point de synchronisation. La Coo-sérialisabilité n'incite pas les gens à coopérer, elle n'est là que pour garantir un niveau de cohérence des données. Cependant, de manière intuitive, chacun est à même de se rendre compte que plus la divergence est élevée, plus la phase de convergence risque d'être longue. L'équipe virtuelle prend alors le risque de passer plus de temps à converger qu'à travailler.

Cette observation simple nous a donné des directions de recherche:

1. la divergence doit pouvoir être mesurée et visualisée. Elle doit contribuer à tenir informés les membres d'un équipe virtuelle sur le degré de divergence existant dans leurs projets.
2. la phase de convergence doit être accélérée ou assistée de manière à la rendre la plus courte possible.

### 6.2.2 Métrique de Divergence

**Participants :** Abdelmajid Bouazza, Pascal Molli, Hala Skaf.

Supposons une mémoire commune contenant des objets. Dans un premier temps, plusieurs processus démarrent et prennent chacun une copie des objets partagés. Puis chaque processus applique des opérations quelconques sur ses propres copies. On peut mesurer la divergence en estimant le nombre et l'importance des conflits qu'il faudra résoudre pour que tous les processus aient à terme, la même valeur pour chaque objet partagé<sup>[BSMM99]</sup>. C'est bien sur

---

2. traduction de "awariness"

---

[BSMM99] 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.



beaucoup plus facile à dire qu'à faire<sup>[Dou95,Dou96]</sup>.

On trouve dans la littérature des approches basées sur la divergence dans les domaines suivants :

**Gestion de configurations** Les gestionnaires de configuration permettent à plusieurs utilisateurs de travailler en parallèle sur des données communes, puis d'intégrer, de fusionner *a posteriori* leurs modifications avant de terminer. Le problème principal de ces techniques est que le coût d'intégration peut être très élevé lorsque la divergence initiale est trop importante.

**Transformations opérationnelles** Ces techniques se basent sur des matrices de transformation pour fusionner les opérations concurrentes de sites différents. Mais, d'une part la complexité de l'application peut conduire rapidement à une explosion combinatoire de la matrice de transformation, d'autre part la divergence autorisée est relativement faible en volume et en durée.

**Bases de données répliquées** Ces techniques se fondent essentiellement sur des modèles de transactions avancés. Elles essaient de trouver un ordre d'exécution qui, soit prévient les conflits en se basant sur la sérialisabilité, soit les empêche de trop diverger en leur associant des métriques comme le fait l'Epsilon-sérialisabilité<sup>[PHK<sup>+</sup>95]</sup>. Le problème essentiel de cette approche est qu'au delà de la limite de divergence tolérée, les transactions divergentes sont annulées, ce qui n'est pas réaliste dans des applications de longue durée qui nous concernent.

Ces différentes approches se sont développées indépendamment les unes des autres. Notre approche tente de les concilier <sup>[BSMM99]</sup> : les gestionnaires de configuration offrent un modèle simple pour converger, mais n'essaient ni d'estimer le coût de convergence ni de proposer des algorithmes pour converger. L'Epsilon-sérialisabilité montre que la divergence peut-être mesurée et tolérée dans une certaine mesure mais utilise un moyen trop radical pour retrouver un état cohérent. Nous proposons de suivre une approche « gestion de configuration », mais d'utiliser des métriques pour mesurer la divergence entre les configurations de travail des différents utilisateurs. Si la divergence devient trop élevée, alors des algorithmes de consensus, par exemple basés sur la COO-sérialisabilité, peuvent être utilisés pour resynchroniser les différentes répliquations.

Aussi, nous approfondissons actuellement les pistes suivantes <sup>[BSMM99]</sup> :

**Métriques basées sur le calcul des différences.** Nous proposons d'utiliser une approche multi-versionnée. Une métrique simple peut reposer sur le calcul des différences entre les différentes versions des participants. Aussi, nous utilisons des techniques bien connues de calcul de différences entre deux fichiers par exemple. Nous utilisons aussi utiliser des techniques de différences plus spécialisées, par exemple celles basées sur XML.

**Métriques basées sur le nombre de lectures impropres.** Cette métrique vient en fait

- 
- [Dou95] P. DOURISH, «The Parting of the Ways: Divergence, Data Management and Collaborative Work», *in: 4th European Conference on Computer Supported Cooperative Work*, K. S. Hans Marmolin, Yngve Sundblad (éditeur), Kluwer Academic, p. 215–230, Stockholm, Sweden, 1995.
  - [Dou96] P. DOURISH, «Consistency Guarantees: Exploiting Application Semantics for Consistency Management in a Collaboration Toolkit», *in: Computer Supported Cooperative Work*, M. S. Ackerman (éditeur), ACM Press, p. 268–277, 1996.
  - [PHK<sup>+</sup>95] C. PU, W. HSEUSH, G. E. KAISER, W. K.L., P. S. YU, «Divergence Control for Distributed Database Systems», *in: Distributed and Parallel Databases*, Kluwer Academic, 1995.

de l'Epsilon-sérialisabilité. En évaluant au fur et à mesure l'ensemble des lectures impropres réalisées par rapport à une version de référence, il est possible d'évaluer le degré d'incohérence d'un ensemble de réplicats. Dans l'Epsilon-sérialisabilité, seules des transactions de type « lectures seules » peuvent faire des lectures impropres. Notre approche est donc de généraliser cela à tous les types de transactions.

**Métriques basées sur le nombre d'opérations non ordonnées.** Ces métriques viennent des systèmes distribués. En collectant le nombre de conflits potentiels d'ordonnement par la simple analyse d'un journal des modifications, il est possible d'évaluer *a priori* la difficulté des ré-ordonnements futurs qui se produiront durant la phase de convergence.

**Combiner les métriques.** Bien sûr, toutes ces métriques peuvent être combinées pour obtenir une moyenne et corriger ainsi les résultats peut être non significatif d'une des métriques.

### 6.3 Collaboration

L'explosion actuelle des applications distribuées, telles le télétravail, le travail coopératif (CSCW), la téléconférence, le commerce électronique, etc., a donné une importance accrue aux systèmes de décision assistée par ordinateur. Dans ce cadre, nous étudions plus particulièrement la formalisation et l'implantation de modèles génériques de négociation et d'argumentation au sein des applications distribuées coopératives. En effet, pour étudier les alternatives possibles de leurs décisions collectives, les utilisateurs d'un environnement de travail coopératif, potentiellement distribués dans l'espace et dans le temps, ont besoin de mécanismes d'aide à la négociation et à l'argumentation.

#### 6.3.1 Argumentation

**Participants :** Jacques Lonchamp, Fabrice Muller.

L'objectif du thème argumentation est de développer un environnement générique de support aux tâches en collaboration à forte composante décisionnelle, où l'argumentation sert à préparer les prises de décision (ex: co-conception, évaluation multi-métiers de solutions en ingénierie concurrente, revue de code,...). L'intérêt principal est l'intégration de concepts et techniques provenant à la fois des environnements paramétrés par des modèles (type workflow), des collecticiels et des systèmes d'aide à la décision, en visant à les enrichir mutuellement : par exemple, lier l'évolution dynamique du modèle à une décision argumentée, ou encore, développer de la conscience de groupe dans un contexte décisionnel.

L'approche suivie a consisté tout d'abord à intégrer au sein d'un modèle de procédé classique, la description précise des types de questions à résoudre, des types de produits et des rôles concernés par ces résolutions (concept de modélisation à grains fins orientée décision). Puis, à intégrer des protocoles d'argumentation évolués (incluant par exemple des contraintes qualitatives d'importance relative des arguments) et des procédures d'assistance fondées en particulier sur l'analyse des arbres argumentatifs (niveau de participation, niveau d'activité, niveau de conflit). Enfin, à réexplorer dans ce contexte particulier, deux questions fondamentales

pour tout système coopératif : l'adaptabilité (y compris dynamique) du système aux décisions du groupe selon les circonstances, et aussi la bascule entre les modes de travail asynchrone et synchrone.

Le prototype actuel (DOTS pour "Decision Oriented Task Support") est entièrement développé en java. Les résultats récents concernent l'intégration de réunions virtuelles synchrones aux tâches asynchrones. Dans cette approche, les participants peuvent choisir de manière opportuniste de passer du mode asynchrone à des sessions en temps réel, en conservant le contexte du travail en cours. Le modèle de tâche peut prescrire une démarche pour organiser les sessions temps réel. Un tableau de bord pour chaque question débattue permet d'aider à la décision de basculement au mode synchrone, souvent motivé par une difficulté à trouver un consensus entre les participants.

Deux domaines d'application ont été explorés récemment : la fusion collective de graphes de concepts et la co-conception en ingénierie concourante [17]. Un projet d'évaluation en vraie grandeur des concepts et du prototype DOTS est en cours de définition avec des ergonomes cogniticiens et des industriels.

### 6.3.2 Négociation

**Participants :** Karim Baina, Khalid Benali.

L'objectif du thème négociation est de construire un modèle de négociation qui soit indépendant de tout 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 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,...).

Pour atteindre cet objectif, nous avons choisi d'utiliser une approche transactionnelle basée sur des actes du langage : une négociation est vue comme étant une transaction possédant un début (solicit), une fin (confirm ou kill), et devant respecter certains critères. Nous avons journalisé un certain nombre d'événements (notamment des actes du langage via l'invocation d'opérations *assert(speech\_act)* au niveau de chaque agent (notion d'histoire locale), puis avons défini des propriétés qui devront être vérifiées sur ces histoires (notion de critère de correction). Certaines de ces propriétés seront dédiées à la coordination des événements entre deux agents (protocole de négociation), alors que d'autres concerneront le contrôle des décisions prises par un agent (tactiques mises en oeuvre par cet agent). 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". Pour reprendre le vocabulaire des systèmes transactionnels, on parle de critères de correction des négociations. Ces travaux ont été décrits dans CoopIS [19] et ont été implanté au sein de DisCOO notre environnement de travail coopératif permettant à différents agents (des activités pilotées par des opérateurs humains) géographiquement éloignés de coopérer via le partage de ressources (essentiellement des documents) NIS [8].

Nous intéressant prioritairement à des agents humains, nous avons quelque peu négligé les tactiques de négociation en supposant que l'agent humain se chargera, seul, de cet aspect de la négociation. Néanmoins la prise en compte de cet aspect de la négociation et son implantation nous permettra, dans un futur proche, de programmer des agents réactifs qui pourront assister les négociateurs humains dans la mesure où leur tactique aura été formellement décrite. Une autre direction de recherche dans laquelle nous avons commencé à exploiter nos résultats dans la modélisation de la négociation est le domaine des workflows coopératifs. Afin de permettre la coopération et, de là, la coordination de différents workflows, diverses négociations devront être menées entre des activités de workflows différents (rendez-vous, contrats, résultats attendus,...) afin d'échanger des services.

## 6.4 Relations entre différentes dimensions de la coopération

### 6.4.1 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 ...

### 6.4.2 Convergence synchrone/asynchrone dans les équipes virtuelles

**Participants :** Abdelmajid Bouazza, Claude Godart, Pascal Molli.

La convergence est un des points faibles des protocoles permettant aujourd'hui à une équipe virtuelle de partager des données. Les protocoles comme le copier-modifier-fusionner <sup>[Fei91,Cou89]</sup> ou le protocole COO [1] sont conçus pour des interactions essentiellement asynchrones. Dans le cadre des équipes virtuelles, la recherche de consensus de manière asynchrone peut avoir comme objectif « fixer une date de réunions (virtuelles) » ou « fournir une nouvelle révision du logiciel pour la fin de la semaine ». Un protocole complètement asynchrone de convergence est alors tout à fait inacceptable quand une équipe virtuelle doit finir un rapport avant une date butoir ou corriger un bogue de dernière minute avant une démonstration. Nous sommes

---

[Fei91] P. H. FEILER, « Configuration Management Models in Commercial Environments », *rapport de recherche n° CMU/SEI-91-TR-7 ESD-9-TR-7*, Software Engineering Institute, Carnegie Mellon University, March 1991.

[Cou89] W. COURINGTON, « The Network Software Environment », *rapport de recherche n° FE197-0*, Sun Microsystems Inc., février 1989.

convaincu que ces phases de convergence ou le temps de convergence est un facteur important doivent pouvoir être améliorées en tirant parti d'un mode de travail plus synchrone.

Notre approche consiste donc à pouvoir passer d'un mode de travail asynchrone caractéristique des phases de divergence à un mode de travail synchrone ou partiellement synchrone pour les phases de convergence. Cela signifie que nous devons pouvoir passer d'un mode de travail où les changements sont publiés de manières différés (comme dans les transactions) à un mode de travail où les changements sont visibles immédiatement par tous.

La communauté CSCW a développé des algorithmes dans le domaine des éditeurs synchrones permettant de produire des changements en parallèle tout en maintenant la cohérence des copies de chaque utilisateur<sup>[EG89,BLK92,SJZ<sup>+</sup>98,VCFS00]</sup>. En observant les algorithmes utilisés par les systèmes asynchrones et synchrones nous avons pu constater de nombreuses analogies. Nous avons pu établir un modèle unifié [10] supportant aussi bien le mode de travail synchrone et asynchrone.

Ce modèle est prometteur dans le sens où il pourrait permettre de construire des environnements où il ne faut plus différencier les fonctionnalités synchrones et asynchrones puisqu'il est possible de passer de l'un à l'autre par un simple ajustement de paramètre. Si tel est le cas, ce serait une avancée majeure.

### 6.4.3 Description des contraintes d'interaction

**Participants :** Gérôme Canals, Claude Godart, Samir Tata.

Le modèle d'interactions que nous développons [5] adresse une large gamme d'applications coopératives. En effet, il est, à un certain niveau d'abstraction, indépendant de la granularité des objets partagés et du mode de travail adopté par l'équipe virtuelle (synchrone, asynchrone ...). C'est au moment de sa mise en oeuvre que l'on s'oriente vers un modèle de coopération ou un autre. Pour cela, nous avons étudié différents mécanismes de mise en oeuvre pour différents types d'applications coopératives.

Notre modèle distingue trois composantes : un modèle de contrôle d'accès, un modèle de

Le modèle de contrôle d'accès définit les rôles en terme de droits d'accès. Il décrit les relations entre les participants accédant aux objets partagés qui peuvent être composites. Nous avons défini des règles d'inférence suivant plusieurs dimensions dont celle des objets composites

Le modèle de synchronisation décrit la coordination des activités des différents participants d'une équipe virtuelle. Il se caractérise principalement par le support des objets composites et la

- 
- [EG89] C. A. ELLIS, S. J. GIBBS, «Concurrency Control in Groupware Systems», *in: SIGMOD Conference, 18*, ACM Press, p. 399–407, Portland, OR, 1989.
- [BLK92] M. BEAUDOUIN-LAFON, A. KARSENTY, «Transparency and Awareness in a Real-time Groupware System», *in: 5th Annual ACM Symposium on User Interface Software and Technology*, ACM Press, 1992.
- [SJZ<sup>+</sup>98] C. SUN, X. JIA, Y. ZHANG, Y. YANG, D. CHEN, «Achieving Convergence, Causality-preservation and Intention-preservation in real-time Cooperative Editing Systems», *in: ACM Transactions on Computer-Human Interactions, 5*, p. 63–108, 1998.
- [VCFS00] N. VIDOT, M. CART, J. FERRIÉ, M. SULEIMAN, «Copies convergence in a distributed real-time collaborative environment», *in: Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW-00)*, ACM Press, Philadelphia, Pennsylvania, USA, December 2000.

description des relations de synchronisation entre des activités coopératives. Ce modèle permet de définir les contraintes de synchronisation que les interactions doivent satisfaire. Cela nous permet de ne pas décrire explicitement comment les activités doivent travailler pour pouvoir coopérer, mais simplement de définir de quelle manière doivent se dérouler les interactions entre elles.

Le modèle d'interaction est défini sous la forme de l'intégration du modèle de contrôle d'accès et du modèle de synchronisation. Ainsi, il supporte la gestion des rôles et la coordination des activités. Grâce aux modèles qui le constituent, il prend en compte le caractère imprévisible des certaines interactions et la complexité des objets partagés. Il définit les interactions en terme de politiques de coopération qui décrivent les règles de travail établies dans une équipe virtuelle. Celles-ci sont définies de manière à tenir compte de l'aspect cohérence et dynamique de la coopération dans les équipes virtuelles.

Une mise en oeuvre particulière, pour une coopération asynchrone et des objets partagés à gros grain, a été réalisée sur le système Motu.

## 6.5 Le Projet SmallEiffel

**Participants** : Dominique Colnet [Resp], Olivier Zendra, Luigi Liquori.

L'aliasing est le fait de *référencer une même donnée via deux variables différentes*, ou plus généralement, deux chemins différents, les *alias*. La donnée en question peut être aussi simple qu'un booléen, ou aussi compliquée qu'une grande structure de données ou un objet complexe. L'aliasing est permis dans la plupart des langages informatiques, langages applicatifs dont les langages à objets ainsi que dans les langages fonctionnels.

Dans le domaine de la compilation et dans le but d'optimiser l'implantation de SmallEiffel, nous avons travaillé sur l'utilisation de l'aliasing dans les langages à objets [22, 9, 6]. L'originalité de nos travaux par rapport à ceux du domaine de l'aliasing tient à l'approche « positive » envers l'aliasing que nous avons adoptée, et qui, à notre connaissance, a rarement été suivie.

Nos travaux actuels de recherche concernent également la sécurité du typage et plus précisément les problèmes liés à la covariance [14]. En effet, la définition actuelle du langage Eiffel autorise la redéfinition covariante des primitives: le type des arguments d'une primitive peut évoluer dans le même sens que la notion de sous-type. L'avantage de cette décision est d'augmenter le pouvoir expressif en laissant une très grande liberté dans les redéfinitions. L'inconvénient majeur de cette décision est de rendre le système de typage non-valide: la détection des incohérences de type n'est pas toujours possible statiquement. Nous pensons qu'il est possible de supprimer la règle de covariance dans la définition du langage Eiffel sans perdre trop d'expressivité grâce à l'ajout d'un mécanisme complémentaire sûr: le type *match*. Cette nouvelle indication de type (*match* Current ou *match* Type) permet au concepteur d'indiquer de manière ancrée le type d'une entité sans laisser la possibilité de covariance des types ancrés *like* actuels. Ce nouveau mécanisme est en cours d'expérimentation afin de vérifier d'une part que la limitation d'expressivité est acceptable et que la nouvelle définition du système de types correspondant est sûre. Le résultat de ce travail pourrait bien entendu s'appliquer à n'importe quel autre langage à base de classes.

## 6.6 Systèmes Formels à Prototypes pour le Langages à Objets

**Participants :** Luigi Liquori [Resp], Dominique Colnet, Olivier Zendra.

Parmi les langages à objets, les langages à prototypes (ou langages à délégation) constituent un axe de recherches important. Plus particulièrement, on se place dans un contexte de calculs à objets où la création des nouveaux objets est déléguée aux objets mêmes (calculs à délégation ou *delegation-based*). Ces calculs paraissent à ce jour comme les plus prometteurs pour une immersion dans un environnement de processus parallèles afin d'obtenir un langage à objets concurrents.

Parmi ces calculs, le Lambda Calculus of Objects (LCO) défini par Fisher, Honsell et Mitchell [FHM94] à l'Université de Stanford, et le Object Calculus (OC) défini par Abadi et Cardelli [AC96] au Centre de Recherche Digital de Palo Alto en sont les principaux représentants. En particulier, les aspects de typage et d'implantation des dits calculs ont été développés.

Dans les calculs à délégation, les objets sont définis directement à partir d'autres objets, en utilisant ces derniers comme prototypes. Les seules opérations sur les objets sont l'extension d'un objet avec une variable ou une méthode (*object extension*), et la surcharge d'une variable ou d'une méthode (*object override*). L'objet modifié hérite de toutes les propriétés du prototype. Plusieurs modèles fonctionnels et impératifs pour ces langages ont été présentés dans la littérature scientifique.

Les calculs LCO et OC conduisent à divers systèmes de types qui empêchent l'erreur `message-not-found` à l'exécution, qui apparaît quand un objet reçoit un message qui n'est pas présent dans son interface. Les systèmes de types pour LCO et OC sont très puissants: en particulier ils permettent la *mytype method specialization*, c'est à dire la possibilité de spécialiser les types des méthodes héritées (une telle possibilité est déjà offerte par le langage Eiffel, qui malheureusement a un système de types qui ne prévient pas l'erreur `message-not-found`). Les langages Obliq, Self, Kevo, Emerald, Cecil, Omega sont les plus connus.

Néanmoins, les langages à prototypes sont moins utilisés que les langages fondés sur les classes. Par contre, ils peuvent être utilisés comme langages cibles pour implanter et étudier des propriétés formelles des langages à classes, puisque les classes peuvent être vues comme des objets capables de recevoir le message `new` de création d'un objet.

Cette veine de recherche a permis la rédaction des articles [24, 23, 25].

## 7 Contrats industriels (nationaux, européens et internationaux)

### 7.1 AEE : Architecture pour l'Electronique Embarquée

**Participants :** Olivier Perrin (responsable), Nacer Boudjlida, Philippe Canalda, Claude Godart.

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

---

[FHM94] K. FISHER, F. HONSELL, J. C. MITCHELL, « A Lambda Calculus of Objects and Method Specialization », *Nordic Journal of Computing* 1, 1, 1994, p. 3–37.

[AC96] M. ABADI, L. CARDELLI, *A Theory of Objects*, 1996.

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.

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.

Pour plus d'information : <http://aee.inria.fr/>

## 7.2 Bio-Expert

**Participants** : Karim Baina (responsable).

Le portail thématique et communautaire [bio-expert.com](http://bio-expert.com) permettra de rendre accessible à la communauté des biologistes une base de connaissances de haute teneur dans le domaine du biomédical pour la formation professionnelle. Le développement de la téléplateforme de formation en-ligne de bio-expert s'appuie sur quatre fondements scientifiques et technologiques à savoir la biologie médicale, l'imagerie, la gestion des connaissances et le travail coopératif. De plus, le projet de startup bio-expert, porté par Karim Baina, Ingénieur doctorant dans l'équipe ECOO, est soutenu par des coéquipiers ayant des compétences complémentaires technologiques et organisationnelles. Le projet bio-expert, de part son objectif et son équipe fondatrice, a été élu lauréat du Concours des Entreprises Innovantes 2000 du Ministère de la Recherche, dans la catégorie émergence. Le projet est accompagné par la mission d'essaimage de France Telecom, par l'INRIA et les Universités lorraines au travers de l'incubateur Lorrain et par d'autres partenaires industriels. Le projet bio-expert est en phase de recherche de financements, parmi ses contacts privilégiés, il y a ISource (le fonds d'amorçage de l'INRIA) et INNOVACOM (la société de capital-risque créée par France Telecom).

## 7.3 Cacao

**Participants** : Pascal Molli (responsable), Christophe Bouthier, Claude Godart, Olivier Malcurat, Marc Patten.

La coopération principale, démarré en 1998 se fait dans le cadre du projet Cacao (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).

Pour plus d'information : <http://www.loria.fr/equipes/ecoo/cnet/>

Une nouvelle collaboration avec le CNET à Caen démarre sur le thème « Procédés inter-entreprises ».



## 7.4 Corvette

**Participants** : François Charoy (responsable), Karim Baina, Claude Godart, Daniela Grigori, Saad el Hadri, Hala Skaf.

Ce travail se fait en collaboration avec le laboratoire Hitachi SDL à Kawasaki. 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 administratif 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.

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

**Participants** : Dominique Colnet (responsable), Jacques Besse, 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).

Le compilateur a été livré en septembre 2000.

## 8 Actions régionales, nationales et internationales

### 8.1 Actions régionales

Ecoo participe au CPER « Intelligence Logicielle » dans les axes « Qualité et sûreté Logicielle », « Téléopérations et assistants intelligents », « Bio-informatique », « Calcul à haute performance ».

Les membres d'ECOO ont fait des conférences dans le cadre du LoriaTech, thèmes « Vers l'entreprise Virtuelle », « De Merise à UML » et « Les clusters de PC ».

Nous collaborons avec le CRAI (Centre de Recherche en Architecture et Ingénierie) de Nancy dans le cadre du projet Cacao.

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

### 8.2 Actions nationales

Gérôme 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 est membre de la conférence Bases de Données Avancées (BDA'2001).

Khalid Benali est membre élu du CNU. Il a été et est membre du comité de programme de Inforsid'2000, Inforsid'2001.

### 8.3 Actions Internationales

#### 8.3.1 Thèse en co-tutelle

Une thèse en co-tutelle entre le l'EMI (Ecole Mohamedia d'Ingenieurs) et le LORIA démarre. Co-encadrement de D. Chiadmi (EMI) et K. Benali (LORIA).

#### 8.3.2 Participation à des comités de programme

Nacer Boudjlida a été membre des comités de programme de Maghrebien Confrence on Software Engineering and Atificial Intelligence à Fès, et du workshop ICSE CoSET (Constructing Software Engineering Tools) 2000 à Limerik.

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

Claude Godart a été et est membre des comités de programme : - de la conférence ICDE (International Conference on Data Engineering) 2001 à Heidelberg, - de la conférence SEKE (Software Engineering and Knowledge Engineering) 2001 à Buenos Aires, - de la conférence ICPADS (International Conference on Parallel and Distributed Systems) 2000 à Iwate, 2001 à Kyongju City, - du workshop VLDB TES (Technologies for Electronic Services) 2000 au Caire, 2001 à Rome, - du workshop ITVE (Information Technology for Virtual Enterprises) 2001 à Gold Coast, - du workshop Healthcom2000 à Sydney, des comités de lecture : - de TCS (Theoretical Computer Science Journal), special issue on dependability, - de VLDB Journal, special issue on e\_services integration.

### 8.3.3 Participation à des conférences

Des membres du projet ont participé aux conférences ou colloques suivants: CAISE, CE, CoopIS, CSCW, HealthComm, HPCN, ICPADS, LMO, NOTERE, TOOLS'EUROPE, TOOL'S PACIFIC.

## 8.4 Visites, et invitations de chercheurs

### 8.4.1 Equipes et entreprises virtuelles

Gérôme Canals a effectué un séjour d'un an au CRIM (Centre de Recherche en Informatique de Montréal). Au cours de ce séjour, il a également coopéré avec des membres de l'université de Montréal et de l'université de Concordia.

Daniela Grigori a fait un séjour de trois mois au laboratoire Hewlet Packard à Palo Alto. Francois Charoy y a fait un séminaire sur «l'anticipation dans les workflows».

Pascal Molli et Hala Skaf ont fait un séjour d'un mois à Sydney (University of Western Sydney et University of New South Wales). Claude Godart a fait un séjour de deux semaines à Sydney (Séminaire «Coordination d'une entreprise virtuelle» à University of Western Sydney et University of New South Wales). Pradeep Ray (University of New South Wales) a fait un séjour d'un mois au LORIA. Vijay Varadjaravan (University of Western Sydney) a fait un séminaire à Nancy en février 2000 («La sécurité dans les entreprises virtuelles»).

Khalid Benali a fait un séminaire à l'Ecole Mohamedia d'Ingénieurs («Workflow et travail coopératif»).

## 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.

### 9.2 Enseignement

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

## 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.

### Thèses et habilitations à diriger des recherches

- [3] J. BESSE, *Environnement de développement et simulateur pour les micro-processeurs PROMIC*, Mémoire diplôme d'ingénieur, CNAM - Nancy, novembre 2000.
- [4] D. COLNET, *Compilation, langages à objets et programmation par contrats. Expérience acquise dans le cadre du projet "SmallEiffel The GNU Eiffel Compiler"*, Habilitation à diriger des recherches, décembre 2000.
- [5] S. TATA, *Outils pour la description et la mise en oeuvre des interactions coopératives dans les équipes virtuelles*, Thèse d'université, Université Henri Poincaré - Nancy I, 2000.
- [6] O. ZENDRA, *Traduction et optimisation globale dans les langages de classes*, Thèse d'université, LORIA - UHP Nancy 1, octobre 2000.

### Articles et chapitres de livre

- [7] M. BUGLIESI, G. DELZANNO, M. MARTELLI, L. LIQUORI, «Object Calculi in Linear Logic», *Journal of Logic and Computation* 10, 1, février 2000, p. 75–104.
- [8] M. MUNIER, K. BENALI, C. GODART, «DisCOO, a really distributed system for cooperation», *Networking and Information Systems Journal*, 2000.
- [9] O. ZENDRA, D. COLNET, «Coping with aliasing in the GNU Eiffel Compiler implementation», *Software - Practice and Experience*, 2000.

### Communications à des congrès, colloques, etc.

- [10] A. BOUAZZA, P. MOLLI, «Unifying coupled and uncoupled collaborative work in virtual teams», *in: ACM CSCW'2000 workshop on collaborative editing systems, Philadelphia, Pennsylvania, USA*, décembre 2000.
- [11] N. BOUDJLIDA, M. SMAIL-TABBONE, M.-D. DEVIGNES, «Services for a Bioinformatics Open Distributed Environment», *in: XEWA '2000: IEEE Workshop on XML-Enabled Wide Area Search in Bioinformatics, League City, Texas*, décembre 2000.
- [12] N. BOUDJLIDA, «Databases: Elements of their Story (A Tutorial)», *in: 10th International Database Symposium, Hong Kong*, Hong Kong Web Society and City University of Hong Kong, juin 2000.
- [13] P. CANALDA, C. GODART, «Coopetitive Business to business modeling – Experiment report», *in: CAISE Workshop on Practical Business Process Modeling (PBPM\*00)*, juin 2000.
- [14] D. COLNET, L. LIQUORI, «Match-O, a Dialect of Eiffel with Match-Types», *in: Proc of IEEE TOOLS, Sydnnet (AU)*, IEEE (éditeur), novembre 2000, <http://www.loria.fr/publications/2000/A00-R-270/A00-R-270.ps>.

- [15] C. GODART, F. CHAROY, O. PERRIN, H. SKAF, « Cooperative Workflows to Coordinate Asynchronous Cooperative Applications in a Simple Way », *in: Parallel and Distributed Systems (PADS)*, I. Press (éditeur), IEEE Computer Society, p. 409–416, juillet 2000, <http://www.loria.fr/publications/2000/A00-R-195/A00-R-195.ps>.
- [16] D. GRIGORI, H. SKAF-MOLLI, F. CHAROY, « Adding Flexibility in a Cooperative Workflow Execution Engine », *in: HPCN Europe 2000*, mai 2000.
- [17] J. LONCHAMP, « A Generic Computer Support for Concurrent Design », *in: Concurrent Engineering 2000, Lyon*, CERA, TECHNOMIC Pub.Co, Inc, Lancaster, USA, juillet 2000, <http://www.loria.fr/publications/2000/A00-R-196/A00-R-196.ps>.
- [18] M. MUNIER, J.-M. ANDREOLI, S. CASTELLANI, « AllianceNet : Information Sharing, Negotiation and Decision-Making for Distributed Organizations », *in: EC-Web'2000 (First International Conference on Electronic Commerce and Web)*, London, Greenwich, DEXA 2000, 2000.
- [19] M. MUNIER, K. BAINA, K. BENALI, « A Negotiation Model for CSCW », *in: Cooperative Information Systems (CoopIS'2000)*, LNCS 1901, p. 224–235, septembre 2000, <http://www.loria.fr/publications/2000/A00-R-319/A00-R-319.ps>.
- [20] M. POIRIER, J.-M. NG, Y. GUO, G. CANALS, R. DSSOULI, « Une Architecture pour la Gestion de la Qualité de Service », *in: NOTERE, Paris*, novembre 2000.
- [21] B. SONNTAG, A. VIGNIER, « Un algorithme polynomial pour le problème ouvert du "coupled tasks" », *in: ROADED'2000*, p. 57–58, janvier 2000.
- [22] O. ZENDRA, D. COLNET, « Vers un usage plus sûr de l'aliasing avec Eiffel », *in: 5ème Colloque Langages et Modèles à Objets (LMO'2000)*, Mont Saint-Hilaire, Québec, Canada, Hermes, p. 183–194, janvier 2000, <http://www.loria.fr/publications/2000/A00-R-022/A00-R-022.ps>.

## Rapports de recherche et publications internes

- [23] H. CIRSTEA, C. KIRCHNER, L. LIQUORI, « A Rho Cube », *Rapport de recherche*, novembre 2000.
- [24] H. CIRSTEA, C. KIRCHNER, L. LIQUORI, « Matching Power », *Rapport de recherche*, novembre 2000.
- [25] F. LANG, P. LESCANNE, L. LIQUORI, K. R. DOUGHERTY DAN, « A Generic Object-Calculus Based on Addressed Term Rewriting Systems », *Rapport de recherche*, novembre 2000, Ajouter Kristoffer Rose, IBM T. J. Watson Research Center.