INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Action MIMOSA

Migration et Mobilité: Sémantique et Applications

Sophia Antipolis

THÈME 1C

Table des matières

MIMOSA *est une action commune entre l'INRIA, le Centre de Mathématiques Appliquées (CMA) de l'Ecole des Mines de Paris et le Centre de Mathématiques et d'Informatique (CMI) de l'Université de Provence.*

1 Composition de l'équipe

Responsable Scientifique

Gérard Boudol [Directeur de Recherches, Inria]

Responsable Permanent

Ilaria Castellani [Chargée de Recherche, Inria]

Assistantes de Projet

Catherine Juncker [Inria]

Dominique Micollier [Armines]

Personnel Inria

Davide Sangiorgi [Chargé de Recherche, Inria]

Personnel CMA et CMI

Roberto Amadio [Professeur, université de Provence]

Frédéric Boussinot [Maître de Recherches, École des Mines de Paris]

Chercheurs Doctorants

Raul Acosta-Bermejo [Allocataire CONACYT, à partir de septembre]

Cédric Lhoussaine [Allocataire MENESR]

Massimo Merro [Boursier TMR, jusqu'à avril]

Jean-Ferdy Susini [Allocataire École des Mines de Paris]

Vincent Vanackere [Allocataire ENS Lyon, à partir de septembre]

Pascal Zimmer [Allocataire ENS Lyon, à partir de septembre]

Chercheurs Post-Doctorants

Josva Kleist [Projet RNRT MARVEL, à partir de septembre]

Antonio Ravara [Projet RNRT MARVEL, à partir de septembre]

Stagiaires

Raul Acosta-Bermejo [DEA RSD, Nice]

Julien Demaria [Maitrise, Nice]

Tri Nguyen-Huu [Mastère, ENS Lyon]

Andrea Valente [PhD, université de Turin]

Vincent Vanackere [DEA DIF, ENS Lyon]

Pascal Zimmer [DEA DIF, ENS Lyon]

2 Présentation et objectifs généraux

L'action MIMOSA est commune avec le Centre de Mathématiques Appliquées de l'ENSMP et le Centre de Mathématiques et d'Informatique de l'université de Provence. L'objectif général du projet est de concevoir et d'étudier des modèles de la programmation répartie et mobile, d'en tirer des primitives pour cette programmation, et d'élaborer des techniques de raisonnement et de vérification concernant spécifiquement les problèmes liés au code migrant. Nous développons deux approches : l'approche interactive, fondée sur l'idée de processus indépendants et communicants, et l'approche réactive, où les processus réagissent de concert à des événements diffusés. Nous visons particulièrement à intégrer les primitives de migration dans la programmation réactive, de façon à pouvoir programmer des "agents" participant à des "assemblées réactives" (jeux en réseau par exemple). Nos axes de recherche principaux sont les suivants :

- Modèles et langages pour la mobilité, approche interactive. Dans cet axe de recherche l'objectif est d'étudier les primitives pour la mobilité fondées en particulier sur le modèle du pi-calcul, et de ses extensions réparties.
- Modèles et langages pour la mobilité, approche réactive. Les objectifs sont de poursuivre le développement des outils de la programmation réactive et d'y intégrer des primitives pour la migration.
- Méthodes de raisonnement. Nous développons de telles méthodes pour les modèles de la migration, comme le calcul des Ambients, fondées sur la bisimulation et les logiques temporelles, ainsi que des méthodes de vérification pour des problèmes de sécurité.

– Types. Les systèmes de types sont utilisés, dans les modèles de la mobilité, à la fois pour éviter des erreurs, imposer des disciplines de sécurité, spécifier des comportements, mais aussi pour valider des transformations et des équivalences.

Relations internationales et industrielles :

– Working Group Esprit CONFER2, regroupant 14 partenaires, sur le thème de l'unification des modèles fonctionnels et distribués de la programmation.

– Relations contractuelles bilatérales avec l'université de Lisbonne, sur la sémantique des objets concurrents, avec l'université de Münich, sur les techniques de vérification pour les langages impératifs, parallèles et d'ordre supérieur, avec l'université de Pennsylvanie, sur l'utilisation des types dans le raisonnement à propos des systèmes concurrents.

– Projet RNRT MARVEL, avec France Télécom R&D et l'ENST, sur la définition d'un modèle de programmation répartie avec objets mobiles, et d'une machine virtuelle associée.

– Deux CTI France Télécom R&D, l'une sur les objets réactifs distribués, l'autre sur la modélisation et la vérification des objets migrants.

– Projet IST PING, regroupant sept partenaires, où nous intervenons sur l'utilisation des techniques réactives dans la programmation des jeux en réseau.

3 Fondements scientifiques

3.1 Sémantique de la mobilité

Mots clés : concurrence, sémantique, parallélisme asynchrone, typage, langage fonctionnel, objets concurrents, mobilité, lambda-calcul, π-calcul.

Participants : Roberto Amadio, Gérard Boudol, Ilaria Castellani, Cédric Lhoussaine, Massimo Merro, Davide Sangiorgi.

Résumé : *La* mobilité *est devenue l'un des aspects importants des systèmes informatiques, et particulièrement des systèmes distribués. Notre projet s'intéresse à la notion de «code mobile» – et donc à une notion de mobilité logique, plutôt que physique. Il s'agit de dégager les concepts utilisés dans les langages de programmation récemment proposés pour permettre la mobilité des calculs et d'en formaliser la sémantique.*

Les modèles sur lesquels nous développons principalement notre approche formelle de la mobilité sont le *π-calcul* de Robin Milner et le calcul des "*ambients*" de Luca Cardelli. Le premier est un modèle similaire à celui que le lambda-calcul offre pour la programmation séquentielle et fonctionnelle, mais dans lequel le parallélisme est pris en compte, ainsi qu'une certaine forme de mobilité puisque dans le π-calcul les processus se communiquent des noms de

canaux de communication. En fait, le π-calcul contient plus ou moins directement le λ-calcul, sa puissance d'expression est donc déjà bien établie de ce simple fait – ceci en regard de la programmation séquentielle.

Le second explore plutôt le concept de migration : il est fondé sur une notion très générale de "domaine", appelé "ambient", dans lequel des calculs peuvent s'effectuer, et qui possède une structure hierarchique. Dans le calcul des "ambients", la structure d'imbrication des domaines les uns dans les autres évolue dynamiquement, et en fait tout le calcul réside dans ce mouvement (et aussi dans la possible disparition des frontières). Ce mécanisme est en fait extrêmement général, puisqu'on peut l'utiliser pour simuler les machines de Turing.

Nos recherches sur ces formalismes, considérés en tant que modèles pour la programmation des systèmes distribués et mobiles, s'articulent autour de plusieurs axes. Nous étudions en particulier comment peut s'étendre la notion de type, et comment elle peut être utile dans le raisonnement à propos des programmes répartis ou migrants. Cette notion de type, familière dans la programmation séquentielle, paraît encore plus importante dans un cadre distribué, parce que les sources d'erreurs y sont bien plus nombreuses et subtiles. Plus généralement, nous étudions ce que peuvent être des méthodes de raisonnement relatives aux programmes répartis et migrants. Nous visons à appliquer ces méthodes tout spécialement aux problèmes de sécurité, par exemple dans la vérification de protocoles cryptographiques ou l'analyse de flux d'information dans les programmes.

3.2 Programmation réactive

Participants : Frédéric Boussinot, Jean-Ferdy Susini.

Mots clés : réactif, objet, script, parallélisme, événement, World Wide Web, Reactive-C, Java, SugarCubes, Icobj, Junior, distribution, migration.

Résumé : *L'approche réactive a pour but d'étudier divers modèles de programmation dans lesquels existe une horloge logique définissant des instants globaux. Elle a également pour objectif d'implémenter des langages de programmation construits sur ces modèles.*

Le traitement de la concurrence dans les langages de programmation usuels reste très empirique et fondé sur des concepts datés. Par exemple, Java propose les "threads" et les verrous pour la gestion de la concurrence. Nos travaux sur la programmation réactive, initiés avec le langage Reactive-C en 1988, visent à ajouter aux langages existants des primitives de programmation de haut niveau, fondées sur les paradigmes de la programmation synchrone : notion d'instant global, d'événement et de diffusion instantanée.

Plusieurs langages et formalismes réactifs ont été définis et implémentés en Reactive-C : réseaux de processus réactifs, langage synchrone SL, objets réactifs. Ils sont décrits dans un livre qui présente également Reactive-C en détail [7]. L'approche réactive au dessus de Java (plus particulièrement les SugarCubes) est décrite dans un livre [11].

Les SugarCubes sont un ensemble de classes Java pour la programmation réactive. Les SugarCubes proposent une communication de haut niveau à base d'événements diffusés instantanément, dans laquelle la réaction à l'absence est reportée à l'instant suivant. Les Scripts

Réactifs apportent toute la souplesse de l'interprétation à l'approche réactive. La présence des instants permet de définir une migration dans des états cohérents des scripts réactifs à travers le réseau. Nous avons implémenté en SugarCubes une version des scripts réactifs au dessus de Java. L'objectif de la programmation par Icobjs est de fournir une technique de programmation entièrement graphique, à base de combinaisons de comportements. Les comportements sont en fait des scripts réactifs, combinés de différentes façons par des manipulations d'icônes à l'écran.

Le formalisme Junior récemment introduit est un noyau de primitives pour la programmation réactive au dessus de Java. Junior est en quelque sorte le noyau des SugarCubes. Trois objectifs ont motivé la création de ce formalisme : prendre en compte le parallélisme asynchrone des systèmes distribués ; donner une sémantique formelle aux primitives réactives ; enfin, réaliser des implémentations efficaces et supportant un grand nombre d'événements diffusés. Nous avons récemment défini Jr [25], une API pour Junior, qui permet de masquer les diverses implementations que nous avons realisees, en collaboration avec France Telecom R&D.

Nous avons commencé à appliquer la programmation réactive à la conception de mondes virtuels, dans le cadre du projet européen PING. Ce projet cible plus particulièrement les jeux en réseau avec grand nombre de joueurs. Dans ce projet, nous utilisons Junior et les Icobjs pour les comportements d'objets virtuels.

4 Domaines d'applications

4.1 Télécommunications

Mots clés : télécommunications, programmation réactive, mobilité.

Le domaine d'applications "naturel" de nos travaux est celui des télécommunications. Nos contributions y portent à la fois sur la spécification et sur la programmation fiable à l'aide de formalismes adaptés de haut niveau.

4.1.1 Modélisation de systèmes mobiles

La spécification *formelle* des systèmes répartis, par exemple exprimés comme systèmes d'objets concurrents, reste encore très largement à faire, et les besoins dans ce domaine sont encore mal couverts. Nos travaux dans ce domaine ont fait l'objet d'une collaboration avec le FT-R&D, dans le cadre d'une CTI, et se poursuivent actuellement au sein du projet RNRT MARVEL, qui vise concrètement la conception d'une machine virtuelle répartie pour la mobilité.

4.1.2 IHM graphiques

L'utilisation des Icobjs apparaît naturelle pour les interfaces homme-machine télécom, en particulier lorsqu'il s'agit de représenter des animations d'icônes, comme par exemple dans la supervision de réseau. Une autre utilisation possible des Icobjs est la conception de systèmes permettant à l'utilisateur final de programmer ses propres services, par des combinaisons graphiques d'Icobjs. Enfin, le domaine des jeux, en particulier des jeux en réseaux, semble être un

domaine dans lequel les Icobjs peuvent être utilisés, par exemple pour la programmation des avatars de joueurs.

4.1.3 Plateformes distribuées d'exécution réactive

Un besoin existe de plateformes d'exécution répartie (DPE) utilisant ou implémentant l'approche réactive. Nous travaillons en collaboration avec le FT-R&D pour l'étude et la réalisation de ce type d'infrastructure d'exécution. Un des objectifs est la mise en place de mécanismes liés à la « qualité de service » (QoS) et utilisant l'approche réactive. Un second objectif est le « passage à l'échelle » de systèmes dans lesquels intervient un très grand nombre de composants parallèles et d'événements diffusés.

5 Logiciels

5.1 Reactive C, SugarCubes, Icobjs, Junior

Participants : Frédéric Boussinot [correspondant], Jean-Ferdy Susini.

Mots clés : programmation réactive, Java, Icobj, SugarCubes, Junior.

Résumé : *L'ensemble des logiciels développés autour de l'approche réactive est disponible librement sur le Web. Cet ensemble va de Reactive-C aux SugarCubes en passant par Junior et les Icobjs.*

Les SugarCubes sont un ensemble de classes Java pour la programmation réactive. La version 2 des SugarCubes est disponible librement sur le Web en `http://www-sop.inria.fr/mimosa/rp/SugarCubes`. Une comparaison entre les threads Java et les SugarCubes est disponible en `http://www-sop.inria.fr/mimosa/rp/SugarCubes/ComparisonThreadsSugarCubes`. Elle est parue dans un papier [16].

5.1.1 Scripts réactifs

L'interpréteur de Scripts réactifs au dessus de Tcl/Tk implémenté en Reactive-C est disponible librement sur le Web en `http://www-sop.inria.fr/mimosa/rp/ReactiveC/index.html`. L'interpréteur de Scripts Réactifs au dessus de Java est disponible en `http://www-sop.inria.fr/mimosa/rp/SugarCubes`. Il s'agit fondamentalement d'un traducteur qui transforme « à la volée » les scripts en SugarCubes. Des mécanismes de migration ont été introduits dans cet interpréteur.

5.1.2 Icobjs

Deux systèmes de programmation par Icobjs sont disponibles. Le premier, réalisé en Scripts Réactifs au dessus de Tcl/Tk est implémenté en Reactive-C. Le second se présente sous forme d'une série d'applets Java disponibles en `http://www-sop.inria.fr/mimosa/rp/ReactiveApplets`.

6 Résultats nouveaux

6.1 Sémantique de la mobilité

Participants : Roberto Amadio, Gérard Boudol, Ilaria Castellani, Cédric Lhoussaine, Massimo Merro, Davide Sangiorgi, Vincent Vanackere, Pascal Zimmer.

6.1.1 Travaux sur le π-calcul

Les travaux sur le π-calcul ont été pour l'essentiel ceux de M. Merro, qu'il a rassemblés dans sa thèse [12]. Il a en particulier exploité ses travaux précédents sur la théorie du π-calcul "local" – où l'on ne peut acquérir dynamiquement le nom d'un canal récepteur –, dans deux directions différentes : il montre d'abord [21] que dans le π-calcul local, la possibilité de transmettre comme valeur un vecteur de noms plutôt qu'un simple nom n'est pas primitive. Ce fait était en réalité déjà connu en ce qui concerne le π-calcul, mais la nouveauté est que dans le π-calcul local, le codage de la version "polyadique" dans la version "monadique" a de très bonnes propriétés, ce qui n'était pas le cas dans le π-calcul "ordinaire". En effet le codage présenté par Merro n'introduit pas de divergence, et est complètement adéquat par rapport à la congruence à barbes.

Dans un autre travail [19], M. Merro, poursuivant une recherche menée en collaboration avec J. Kleist et U. Nestmann de l'université d'Ålborg, étudie le mécanisme de migration du langage Obliq de Cardelli. Dans ce langage, basé sur la notion d'objet, la migration d'un objet consiste en la création d'un clone distant, et en la transformation de l'objet originel en un alias de ce clone. Ce mécanisme est implémenté, mais sa sémantique n'est pas formellement décrite par Cardelli. Merro avait précédemment montré que l'implémentation ne satisfaisait pas la propriété attendue de transparence de la migration – un programme utilisant un objet migrant ne doit pas pouvoir observer ses mouvements. L'article [19] donne une sémantique formelle pour un fragment d'Obliq, par le biais d'une traduction en π-calcul local, et cette traduction est utilisée pour démontrer que la sémantique de la migration est cette fois "correcte". Ceci est également développé dans la thèse de M. Merro [12].

Mentionnons également qu'une version complète du travail sur le π-calcul réceptif et distribué (décrit dans le rapport d'activité du projet MEIJE de 1999) a été soumise pour publication dans un journal. C. Lhoussaine poursuit actuellement ce travail, avec la définition et l'implémentation d'un algorithme d'inférence de type pour ce calcul.

6.1.2 Travaux sur le calcul des "ambients"

Trouver de bonnes méthodes de raisonnement en ce qui concerne les Ambients est un problème encore peu abordé, et qui semble difficile. En effet dans ce modèle il y a peu de moyens de contrôler le calcul, qui se fait par migration et "dissolution de domaines" : alors qu'on pense avoir écrit un "programme" qui déplace un "ambient" vers un autre par exemple, il n'y a le plus souvent aucun moyen d'empêcher l'environnement de venir perturber cette intention. De ce fait, il est très difficile de se convaincre que le codage en Ambients de telle ou telle spécification est correct. Dans une collaboration avec F. Levi (U. Pise), nous avons

étudié les phénomènes d'interférence entre les processus de ce modèle. Nous avons isolé des formes d'interférence (les interférences "graves") qui peuvent rendre la programmation et la vérification très compliquées. Nous avons proposé une modification des primitives de migration et un système de typage qui éliminent ces interférences. De cette façon on obtient un nouveau modèle, appelé "mobile safe ambients" [20], dans lequel il devient possible de prouver des équivalences intéressantes. Il semble aussi que cette modification du langage ouvre de nouvelles perspectives d'implémentation. Des travaux en collaboration avec A. Valente (U. Turin) à ce sujet sont en cours.

D'autre part, nous avons étudié la "logique des ambients" récemment proposée par Cardelli et Gordon. Il s'agit d'une logique modale qui permet de raisonner à la fois sur les aspects temporels et de répartition spatiale en ce qui concerne les ambients. Pour comprendre le pouvoir expressif de cette logique, nous avons étudié plus particulièrement l'équivalence – satisfaire les mêmes formules – qu'elle induit sur le modèle. Nous avons ainsi montré [30] que, de façon assez inattendue, cette équivalence est très "intensionnelle" : la logique permet en effet d'exprimer des propriétés révélant de manière fine la structure interne des ambients. L'équivalence logique est donc très proche de l'équivalence "structurelle", qui, dans tous les calculs de processus, est considérée comme l'égalité "minimale". L'article [30] a été accepté à la conférence POPL qui aura lieu en janvier 2001.

Dans un autre travail [22], P. Zimmer étudie le pouvoir expressif du calcul des ambients, et plus spécifiquement de sa version "pure", sans communication, où la seule façon de calculer est de faire bouger et dissoudre des ambients. Cardelli et Gordon avaient déjà suggéré qu'on pouvait simuler les machines de Turing dans ce fragment, mais pour représenter le π-calcul par exemple, où des valeurs sont échangées d'un processus à l'autre, la communication de valeur à l'intérieur d'un ambient était utilisée. P. Zimmer a montré dans [22] que ce n'est en fait pas nécessaire : le π-calcul peut être simulé dans le calcul des ambients pur. Etant donné que ce fragment ne possède aucune forme de communication ni de substitution, il faut montrer comment ces notions peuvent être simulées par la mobilité et des modifications de la structure hiérarchique des ambients. Ceci s'avère évidemment plus délicat que dans le calcul avec communication, et pour rendre le codage et les preuves plus aisés, P. Zimmer introduit un modèle intermédiaire, qui est un π-calcul avec substitutions explicites.

6.1.3 Analyse de protocoles cryptographiques

Il existe de nombreuses techniques de spécification des protocoles cryptographiques (et des propriétés qu'ils sont censés vérifier) dans des logiques variées (logique du premier ordre, logique linéaire, algèbres de processus, logique équationnelle,...). Un certain nombre d'expériences montrent que les spécifications peuvent s'exprimer naturellement dans des fragments de la logique du premier ordre. Notre objectif principal dans cet axe de recherche est de développer des méthodes de décision (exactes ou approchées) pour ces fragments et de les évaluer sur des protocoles réels. Un deuxième objectif est d'étudier la pertinence du modèle logique choisi en le mettant en relation avec d'autres modèles où les capacités des attaquants sont définies en termes de probabilité et de complexité de calcul.

Dans notre approche la vérification consiste à s'assurer que certains programmes parallèles qui modélisent le protocole et la spécification ne peuvent pas atteindre un état erroné tout en

interagissant avec l'adversaire. Pour des programmes finis, nous avons présenté dans [18] une procédure simple de décision pour le problème d'accessibilité qui est fondée sur un système de réduction symbolique. Vincent Vanackere a montré comment étendre notre méthode aux clefs publiques et aux fonctions de hachage. Il a également programmé un prototype de vérificateur qui a donné des résultats encourageants [32, 27].

Dans un travail en cours de publication nous avons étudié la complexité du problème de l'accessibilité pour des processus finis et itérés. Pour les processus finis, le problème est NP-dur et NP-complet pour un fragment non trivial. Pour les processus itérés le problème est indecidable en général et décidable en temps polynomial pour un fragment qui n'utilise pas les couples.

6.1.4 Ouvrages de Référence

Dans le cadre du projet RNRT MARVEL, nous avons mené à bien, en collaboration avec des chercheurs de FT-R&D, une revue de l'existant [24]. Nous y étudions dix modèles et langages pour le code mobile, sous divers aspects, et notamment la modélisation et l'expression de la répartition, de la mobilité, et la mise en œuvre de politiques de sécurité. Ce travail de revue était un préliminaire à l'élaboration d'un modèle de programmation pour la mobilité, ce qui constitue l'une des tâches du projet MARVEL (voir [28]).

I. Castellani a écrit un chapitre pour le "Handbook of Process Algebra" [29], qui présente les travaux sur la notion de localité, et plus particulièrement ceux effectués dans le cadre du projet MEIJE. D. Sangiorgi a terminé, avec D. Walker de l'université d'Oxford, l'écriture d'un livre sur le π-calcul [31], à paraître prochainement, publié par Cambridge University Press. Cet ouvrage a été conçu pour être une référence pour la théorie du π-calcul et aussi une démonstration de l'utilisation du π-calcul pour décrire des langages ou systèmes, et leurs propriétés.

6.2 Programmation réactive distribuée

Participants : Frédéric Boussinot, Jean-Ferdy Susini, Raul Acosta-Bermejo.

Mots clés : distribution, réactif, Java, SugarCubes, applets, Junior, threads.

6.2.1 SugarCubes v3

La version 2 des SugarCubes est maintenant disponible sur le Web avec ses sources. Elle introduit plusieurs notions nouvelles pour réaliser un interfaçage plus intégré avec Java, pour implémenter la migration, et également pour faciliter l'ajout dynamique de comportements parallèles. La version 3 est en cours de finalisation. Elle introduit de nouvelles constructions augmentant la puissance expressive du langage.

6.2.2 Junior et Jr

Nous avons dégagé un noyau de programmation réactive en Java que nous avons appelé Junior (Jr, J pour Java, r pour réactif), pour lequel nous avons défini une sémantique formelle sous forme de règles de réécriture conditionnelles. Junior peut être vu comme le noyau des

SugarCubes. Nous avons donné plusieurs implémentations de Junior, l'une d'entre elles étant capable de traiter un grand nombre de composants parallèles et d'événements (de l'ordre de plusieurs milliers). Nous avons défini une API de programmation en Junior. Ce travail est décrit dans un rapport de recherche [25].

6.2.3 Applets réactives

Nous avons illustré la puissance de l'approche réactive en implémentant un ensemble d'applets réactives au dessus de Junior. Ces applets sont disponibles sur le Web. Ces applets nous ont servi pour mener une comparaison entre les SugarCubes et les threads Java dans l'expression de la concurrence au niveau langage. Cette comparaison est publiée dans une revue [16].

6.2.4 ROS-REJO

Nous avons poursuivi l'étude de la notion d'agent réactif migrant commencée avec le prototype RAMA, développé par Navid Nikaein. Nous avons développé une plate-forme ROS permettant d'exécuter des programmes réactifs migrants, appelés REJO. Les REJOs sont compilés en SugarCubes (ou en Junior, au choix). REJO et ROS sont décrits sur le Web en http://www-sop.inria.fr/mimosa/rp/ROS. Ils ont été présentés à la conférence NOTERE2000[17].

6.2.5 Production d'automates

Nous avons étudié la production de machines à états (automates) à partir de programmes Junior. La technique proposée permet en particulier de définir des automates partiels qui peuvent être exécutés bien que tous les états n'aient pas été produits ou analysés. Ce travail est décrit dans un rapport de recherche [26].

7 Contrats industriels (nationaux, européens et internationaux)

7.1 Projet IST PING

Le projet PING se propose de réaliser une plateforme pour des mondes virtuels possédant un grand nombre de participants. Une des cibles du projet est celle des jeux en réseau. Notre tâche dans le cadre de PING est plus particulièrement de fournir un environnement de programmation réactive pour la conception des objets peuplant le monde. Nous étudions également comment les techniques réactives peuvent être utilisées pour préserver la cohérence des diverses visions distribuées du monde virtuel.

7.2 Projet RNRT MARVEL

Participants : Roberto Amadio, Gérard Boudol, Ilaria Castellani, Davide Sangiorgi.

Ce projet RNRT, dont l'intitulé complet est "Machine répartie virtuelle et langage pour objets mobiles", d'une durée de 3 ans, a débuté en décembre 1998. Il inclut comme parte-

naires France Télécom CNET (coordinateur), devenu depuis France Télécom R&D, l'INRIA de Rocquencourt, l'INRIA de Sophia Antipolis et l'ENST de Paris.

Le projet Marvel se propose de jeter les bases d'une programmation d'objets logiciels mobiles à grande échelle ainsi que les bases d'une nouvelle infrastructure logicielle répartie mettant en œuvre ce modèle de programmation.

7.3 CTI FT-R&D : Objets réactifs distribués

Participants : Frédéric Boussinot, Jean-Ferdy Susini.

L'objectif de ce contrat est l'étude des objets réactifs dans leurs aspects liés à la distribution à travers un réseau de communication. Les principaux axes de recherche sont les suivants :

- Étude de l'intégration de la notion d'*Object Request Broker* (ORB) dans le modèle des objets réactifs,
- Introduction de la notion de « qualité de service » dans les objets réactifs, plus particulièrement dans les Scripts Réactifs ; cette étude est liée à la réflexivité : comment un script peut-il influer sur l'interpréteur chargé de l'exécuter, dans l'objectif de respecter une qualité de service ?
- Poursuite des travaux liés à la migration des Scripts Réactifs à travers Internet et le Web.

7.4 CTI FT-R&D : Objets Migrants

Participants : Roberto Amadio, Gérard Boudol, Frédéric Boussinot, Cédric Lhoussaine.

Les objectifs de ce contrat sont, d'une part, de développer des techniques de modélisation, de spécification et de vérification en ce qui concerne les objets migrants, et d'autre part d'expérimenter la programmation et la validation d'agents mobiles, en se fondant sur le travail déjà accompli en ce qui concerne les objets et Scripts Réactifs. Nous portons un effort particulier sur les aspects de typage, aussi bien en ce qui concerne le modèle objet lui-même que pour ce qui touche aux aspects de migration.

8 Actions régionales, nationales et internationales

8.1 Actions nationales

8.1.1 Action concertée incitative Cryptologie VERNAM

Participants : Roberto Amadio, Vincent Vanackere.

Nous coordonnons l'Action concertée incitative Cryptologie VERNAM sur la Vérification automatique de protocoles cryptographiques. L'action débute à la fin de l'année pour une durée de trois ans et comprend le projet PROTHÉO de l'INRIA-Lorraine et l'ENS-Cachan. Toujours dans le cadre de l'ACI Cryptologie, nous participons à un seminaire PACA sur la cryptographie qui comprend des équipes de Toulon et de l'IML.

8.2 Actions européennes

8.2.1 Working Group Esprit LTR Confer2

Participants : Roberto Amadio, Gérard Boudol, Ilaria Castellani, Massimo Merro, Davide Sangiorgi.

Ce groupe de travail Esprit, qui regroupe une quinzaine de sites européens, a pour thème de recherche l'unification, en théorie et en pratique, des modèles fonctionnel et distribué de la programmation. Il expire fin 2000, mais son renouvellement est envisagé.

8.2.2 PROCOPE

Participant : Davide Sangiorgi.

Dans le cadre du programme franco-allemand PROCOPE nous avons une collaboration avec l'université Technique de Munich sur le thème « techniques de vérification pour des langages impératifs et parallèles d'ordre supérieur ».

8.2.3 Coopération Franco-Portugaise

Participants : Gérard Boudol, Ilaria Castellani, Cédric Lhoussaine.

Dans le cadre du programme de coopération luso-française de l'ambassade de France et du ICCTI, nous avons un projet de recherche avec l'université de Lisbonne portant sur la sémantique des objets concurrents.

8.3 Actions internationales

8.3.1 NSF/INRIA

Participant : Davide Sangiorgi.

Le thème de cette collaboration avec l'université d'Indiana, intitulée «Static Types for Reasoning about Concurrent Systems», porte sur les modèles de processus mobiles, le typage dans ces formalismes ainsi que leur pouvoir expressif dans la représentation comportementale des objets concurrents.

8.3.2 Projet franco-indien CEFIPRA 1502-1

Participants : Roberto Amadio, Ilaria Castellani, Gérard Boudol, Cédric Lhoussaine.

Ce projet, intitulé « Spécification et vérification de systèmes distribués : systèmes à états finis et d'ordre supérieur », a été reconduit jusqu'en mai 2000. Il nous lie aux équipes de P.S. Thiagarajan (SMI et IMS Madras) et de S. Prasad (IIT Delhi).

8.4 Visites, et invitations de chercheurs

Nous avons eu la visite d'A. Valente, doctorant de l'université de Turin, pour 6 mois de Janvier à Juin. Il a travaillé avec D. Sangiorgi sur l'implémentation du calcul des Ambients. M. Buscemi, doctorante de l'université de Catania, est en visite pour 3 mois, d'octobre à décembre. Nous avons eu également les visites de V. Sassone, professeur à l'université de Catania, pour un mois (Octobre), de M. Hennessy, professeur à l'université de Sussex, pour 3 semaines (novembre), et, pour des périodes plus courtes, de M. Pouzet (université de Paris 6), A. Ravara et V. Vasconcelos (université de Lisbonne).

9 Diffusion de résultats

9.1 Animation de la communauté scientifique

R. Amadio a présenté ses travaux à la conférence CONCUR (State College), aux workshops Infinite State Systems (Dagstuhl) et CONFER2 (Stockholm) et au Séminaire de l'ENS-Cachan.

G. Boudol était membre du comité de programme de la conférence ESOP, et a participé à cette conférence, tenue à Berlin dans le cadre d'ETAPS (Berlin), ainsi qu'à ECOOP (Cannes). Ilaria Castellani a été membre du comité de programme de la conférence CONCUR. Elle a participé à la conférence ECOOP. Elle a aussi participé à la réunion du projet CONFER2 à Cambridge, en septembre, où elle a donné une présentation. A cette occasion elle a également visité le laboratoire de recherche de Microsoft à Cambridge, où elle a fait un séminaire, et l'université de Sussex.

D. Sangiorgi a fait un exposé invité au workshop Mathematical Foundations of Programming Semantics (MFPS, New York), au cours d'une session spéciale en l'honneur de R. Milner. Il était membre des Comités de programme de la conférence Theoretical Computer Science (TCS, Sendai), du workshop Expressiveness in Concurrency (EXPRESS, Pennsylvania State University), et du workshop Higher Order Operational Techniques in Semantics (HOOTS, Montréal).

Massimo Merro a présenté ses résultats aux conférences ETAPS et TCS. P. Zimmer a participé aux conférences ETAPS, CONCUR et EXPRESS, où il a présenté ses travaux.

G. Boudol a été rapporteur de la thèse présentée par A. Ravara à l'université de Lisbonne. F. Boussinot et D. Sangiorgi ont été examinateurs de la thèse de M. Merro [12] (dirigée par Sangiorgi).

9.2 Enseignements

R. Amadio est responsable de la Licence d'Informatique de l'université de Provence et du cours "Typage et Vérification" du DEA d'Informatique de Marseille. Avec Denis Lugiez, il a encadré le stage de DEA de Vincent Vanackere.

G. Boudol a co-dirigé avec F. Boussinot le stage de mastère de Tri Van Huu. F. Boussinot a encadré le stage de maîtrise de J. Demaria, et le stage de DEA de R. Acosta-Bermejo. D. Sangiorgi a dirigé le stage de DEA de P. Zimmer, ainsi que le stage d'A. Valente. Il enseigne dans le DEA de Programmation à Paris. I. Castellani a assuré le cours de présentation d'option au DEA MDFI de Marseille.

C. Lhoussaine encadre des TP en informatique à l'université de Provence. P. Zimmer assure les TP du cours "Informatique et programmation (en Java)" du DEUG Tronc Commun première année de la Faculté des Sciences de l'université de Nice-Sophia Antipolis.

10 Bibliographie

Ouvrages et articles de référence de l'équipe

[1] R. Amadio, I. Castellani, D. Sangiorgi, «On bisimulations for the asynchronous π-calculus», *Journal of Theoretical Computer Science 195*, 1998.

[2] R. Amadio, P.-L. Curien, *Domains and Lambda-Calculi*, Cambridge University Press, 1998.

[3] G. Berry, G. Boudol, «The chemical abstract machine», *Theoretical Computer Science 96*, 1992.

[4] G. Boudol, «Asynchrony and the π-calculus», *rapport de recherche n° 1702*, INRIA, May 1992.

[5] G. Boudol, «The π-calculus in direct style», *Higher-Order and Symbolic Computation 11*, 1998.

[6] F. Boussinot, R. de Simone, «The SL Synchronous Language», *IEEE Transactions on Software Engineering 22*, 1996.

[7] F. Boussinot, *La programmation réactive*, Masson, 1996.

[8] D. Sangiorgi, B. Pierce, «Typing and subtyping for mobile processes», *MSCS 6*, 1996.

[9] D. Sangiorgi, «Interpreting functions as π-calculus processes: a tutorial», *rapport de recherche n° 3470*, INRIA, août 1998.

[10] D. Sangiorgi, «On the bisimulation proof method», *MSCS 8*, 1998.

Livres et monographies

[11] F. Boussinot, *Objets réactifs en Java*, Collection Scientifique et Technique des Telecommunications, PPUR, 2000.

Thèses et habilitations à diriger des recherches

[12] M. Merro, *Localité dans le π-calcul et applications aux objets distribués*, thèse de doctorat, Ecole des Mines de Paris, octobre 2000.

Articles et chapitres de livre

[13] R. Amadio, S. Prasad, «Modelling IP mobility», *Formal Methods of System Design 17*, 2000.

[14] R. Amadio, «On modelling mobility», *Theoretical Computer Science 240*, 2000.

[15] G. Boudol, «On the semantics of the call-by-name CPS transform», *Theoretical Computer Science 234*, 2000.

[16] F. Boussinot, J.-F. Susini, «Java threads and SugarCubes», *Software Practice & Experience 30*, 2000.

Communications à des congrès, colloques, etc.

[17] R. ACOSTA-BERMEJO, «Reactive operating system», *in : NOTERE'2000*, 2000.

[18] R. AMADIO, D. LUGIEZ, « On the reachability problem in cryptographic protocols », *in : CONCUR'00*, *LNCS*, *1877*, 2000. Paru comme rapport INRIA 3915, `http://www.inria.fr/rrrt/rr-3915.html`.

[19] J. KLEIST, M. MERRO, U. NESTMANN, «Local π-calculus at work: mobile objects as mobile processes», *in : TCS'2000*, *LNCS*, 2000.

[20] F. LEVI, D. SANGIORGI, «Controlling Interference in Ambients», *in : Proc. 27th POPL*, ACM Press, 2000.

[21] M. MERRO, «Locality and polyadicity in asynchronous name-passing calculi», *in : FoSSaCS'00*, *LNCS*, *1784*, Springer, 2000.

[22] P. ZIMMER, «On the Expressiveness of Pure Mobile Ambients», *in : EXPRESS'00*, *BRICS Notes Series*, *NS-00-2*, 2000. Electronic Notes in Theoretical Computer Science, volume 39, Elsevier.

[23] P. ZIMMER, «Subtyping and Typing Algorithms for Mobile Ambients», *in : FoSSaCS'00*, *LNCS*, *1784*, Springer, 2000.

Rapports de recherche et publications internes

[24] G. BOUDOL, F. GERMAIN, M. LACOSTE, « Analyse des modèles et langages de la mobilité », *rapport de recherche n° 3930*, INRIA, avril 2000, `http://www.inria.fr/rrrt/rr-3930.html`.

[25] F. BOUSSINOT, L. HAZARD, J.-F. SUSINI, «Programming with Junior», *rapport de recherche n° 4027*, INRIA, octobre 2000, `http://www.inria.fr/rrrt/rr-4027.html`.

[26] F. BOUSSINOT, «Junior Automata», *rapport de recherche n° 4031*, INRIA, octobre 2000, `http://www.inria.fr/rrrt/rr-4031.html`.

Divers

[27] R. AMADIO, D. LUGIEZ, V. VANACKERE, « On the reachability problem in cryptographic protocols», présenté au Workshop on Issues in the Theory of Security, Genève, juillet 2000.

[28] G. BOUDOL, A. SCHMITT, J.-B. STEFANI, «Towards the MARVEL programming model», Draft, Juillet 2000.

[29] I. CASTELLANI, «Process algebras with localities», à paraître.

[30] D. SANGIORGI, «Extensionality and intensionality of the Ambient logics», à paraître.

[31] D. SANGIORGI, D. WALKER, «The pi-calculus: a theory of mobile processes», à paraître.

[32] V. VANACKERE, « Protocoles Cryptographiques et Calcul Symbolique », Mémoire de DEA, ENS-Lyon, Juin 2000, 2000.