INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Projet MEIJE

Parallélisme, Synchronisation et Temps-Réel

Sophia Antipolis

THÈME 1C

Table des matières

MEIJE est un projet commun entre l'INRIA et le Centre de Mathématiques Appliquées (CMA) de l'Ecole des Mines de Paris. Le projet doit arriver à son terme fin 1999.

1 Composition de l'équipe

Responsable scientifique

Robert de Simone [directeur de recherches, Inria]

Responsable permanent

Gérard Boudol [directeur de recherches, Inria]

Assistantes de projet

Françoise Martin-Trucas [Inria, partiellement dans le projet]
Dominique Micollier [Armines]

Personnel Inria

Amar Bouali [chargé de recherche]
Ilaria Castellani [chargée de recherche]
Annie Ressouche [chargée de recherche]
Davide Sangiorgi [chargé de recherche]

Personnel CMA

Gérard Berry [directeur de recherches, École des Mines de Paris]
Frédéric Boussinot [maître de recherches, École des Mines de Paris]
Xavier Fornari [ingénieur Armines]
Valérie Roy [chargée de recherche École des Mines de Paris]

Chercheurs doctorants

Michel Bourdellès [allocataire MENESR, jusqu'à avril]
Yannis Bres [allocataire MENESR, depuis octobre]
Silvano Dal-Zilio [moniteur normalien, jusqu'à septembre]
Loïc Henry-Gréard [allocataire AMX MENESR]
Cédric Lhoussaine [allocataire MENESR]
Massimo Merro [boursier TMR]
Dumitru Potop-Butucaru [boursier Eiffel]
Jean-Ferdy Susini [allocataire École des Mines de Paris]

Chercheur post-doctorant

Conrado Daws [jusqu'en mai]

Collaborateurs extérieurs

Roberto Amadio [professeur à l'université de Provence]
Charles André [professeur à l'UNSA]
Ellen Sentovich [Cadence Berkeley Labs]

Stagiaires

Cédric Dégea [ESSI 2]
Eric Guillhaure [ESSI 2]
Navid Nikaien [DEA UNSA RSD]

2 Présentation et objectifs généraux

Le projet «Parallélisme, Synchronisation et Temps Réel» est un projet commun entre l'Inria et l'Ecole des Mines de Paris. Ses activités s'articulent autour de quatre thèmes :

- la modélisation mathématique des concepts fondamentaux liés aux systèmes et langages parallèles et répartis, dans des formalismes tels que le π-calcul et le calcul bleu ;
- l'étude des formalismes réactifs synchrones, avec principalement le développement du langage Esterel, de ses modèles sémantiques, de son implémentation efficace et de qualité industrielle, de son environnement de développement et d'analyse ;
- la conception de méthodes de vérification automatique par modèles des programmes parallèles ou réactifs, et le développement des outils Auto/Autograph et Fc2Tools pour les calculs de processus généraux, ainsi que le «model-checker» symbolique Xeve;
- l'étude de modèles de programmation réactive dynamique, autour du langage Reactive C, des classes Java SugarCubes, et de leurs extensions.

3 Fondements scientifiques

3.1 Sémantique de la mobilité

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

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

Cette activité est commune avec l'université de Provence à Marseille, où se trouvent physiquement Roberto Amadio et Cédric Lhoussaine.

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.*

Le modèle sur lequel nous développons principalement notre approche formelle de la mobilité est le *pi-calcul*. C'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 pi-calcul les processus se communiquent des noms de canaux de communication. En fait, le pi-calcul contient plus ou moins directement le lambda-calcul, sa puissance d'expression est donc déjà bien établie de ce simple fait – ceci en regard de la programmation séquentielle.

Nos recherches sur le pi-calcul en tant que modèle pour la programmation des systèmes distribués s'articulent autour de plusieurs axes. L'approche objet étant très largement dominante dans le domaine des systèmes répartis, l'étude formelle du modèle objet, dans ses relations avec le pi-calcul, est naturellement une composante importante de notre travail. Nous étudions aussi comment peut s'étendre la notion de type, et comment elle peut être utile dans le raisonnement à propos des programmes parallèles. Cette notion de type, familière dans la programmation séquentielle, paraît encore plus importante dans un cadre distribué, ne serait-ce que pour assurer une certaine cohérence dans les communications.

Enfin nous étudions la modélisation explicite de la migration, avec la notion de «localité» et celle, associée, de site défaillant, ou simplement temporairement déconnecté. C'est une direction qu'il est essentiel d'explorer, en particulier si l'on veut pouvoir aborder les questions de sécurité.

3.2 Systèmes réactifs synchrones et Esterel

Participants : Gérard Berry, Amar Bouali, Yannis Bres, Robert de Simone, Xavier Fornari, Loïc Henry-Gréard, Dumitru Potop, Annie Ressouche, Valérie Roy.

Cette activité est un thème de collaboration avec l'équipe SPORTS de l'I3S (CNRS/UNSA), dirigée par Charles André.

Mots clés : langage synchrone, réactivité, causalité, sémantique, temps réel, compilation, optimisation, circuit digital.

Résumé : *Le langage Esterel permet la programmation de systèmes réactifs synchrones. La syntaxe impérative du langage est adaptée aux systèmes dominés par le contrôle. Elle repose sur des primitives spécifiques de parallélisme et de préemption. La sémantique formelle permet la définition exacte des programmes, leur traduction vers des formats adaptés à la synthèse de logiciel ou de matériel, l'optimisation de cette synthèse, et la vérification de propriétés de programmes.*

On désigne comme *réactifs* les systèmes dont la caractéristique principale est d'interagir avec leur environnement extérieur au rythme de cet environnement. Les systèmes réactifs *synchrones* s'appuient sur les notions d'horloge globale, de diffusion instantanée d'informations, de parallélisme déterministe et de préemption pour fournir un modèle de programmation cohérent et adapté. Esterel propose les primitives correspondantes en complément d'un langage impératif traditionnel. Les opérations de manipulations de données sont reportées vers un langage hôte, par exemple C.

Les applications d'Esterel sont les contrôleurs temps-réel, les systèmes embarqués, les protocoles de communication, les interfaces homme-machine, et plus généralement les systèmes réactifs dominés par le contrôle.

Le comportement d'un programme Esterel est défini par une sémantique mathématique [1]. Un programme est compilé en un système d'équations booléennes avec mémoires, c'est-à-dire en un circuit synchrone. Cette traduction permet la synthèse directe de circuits électroniques ou la synthèse de programmes par tri des équations et traduction directe en C. Elle permet également de s'interfacer avec de nombreux systèmes de vérification formelle comme ceux développés dans le projet.

Les recherches développées autour du langage et de son modèle concernent :

- la définition d'une sémantique à causalité constructive, permettant de traiter des programmes comportant des cycles de dépendances instantanées d'objets pourvu que ces cycles soient dynamiquement sains,
- l'étude de l'algorithmique efficace pour le calcul de la causalité constructive,
- l'étude de l'optimisation des circuits et programmes engendrés,
- la validation formelle de la traduction du langage en circuits synchrones,
- le lien entre les langages synchrones dominés par le contrôle comme Esterel et les langages synchrones dominés par les données comme Lustre et Signal,
- le lien entre les langages synchrones et les nouvelles méthodes de synthèse de systèmes mixtes matériel / logiciel.

Les développements théoriques conduisent à des algorithmes implantés dans les différentes parties du compilateur Esterel v5. Des retours d'utilisateurs industriels ou universitaires viennent fréquemment susciter de nouvelles questions théoriques et pratiques concernant les méthodologies de conception et leurs besoins algorithmiques.

3.3 Vérification automatique

Participants : Amar Bouali, Michel Bourdellès, Annie Ressouche, Robert de Simone.

Mots clés : vérification, système distribué, parallélisme synchrone, parallélisme asynchrone, analyse par modèle, méthode symbolique.

Résumé :

Le vocable anglais de model-checking *recouvre la généralité des méthodes de vérification et d'analyse automatiques de systèmes parallèles communicants à base d'exploration exhaustive des espaces de configurations atteignables. Cette approche est maintenant largement développée, et elle donne lieu à une intense recherche sur les modes de représentation et les algorithmiques dédiées face au problème de l'explosion combinatoire. Notre objectif est de pouvoir valider des spécifications ou programmes de taille réelle, dans le cadre synchrone (Esterel) ou asynchrone (calculs de processus).*

Les systèmes parallèles posent généralement des problèmes de fiabilité accrus du fait des multiples flots de contrôle qui coexistent. Le phénomène de ce type sans doute le plus connu

est celui de l'interblocage de processus, en attente mutuelle. Le parallélisme peut provenir d'impératifs physiques (systèmes distribués), mais aussi être introduit comme paradigme de modularité logique, comme dans la programmation réactive. Nos travaux sur la modélisation formelle de tels systèmes nous ont permis de dégager ensuite des méthodes de vérification automatique centrées sur une analyse exhaustive des configurations d'états de contrôle atteignables, dans le cas fini (mais de complexité combinatoire élevée).

La problématique fondamentale des logiciels d'analyse par modèles (*model checkers*) est de combattre l'explosion combinatoire. Ceci peut être exploré par l'algorithmique, l'application de méthodes compositionnelles ou la réduction des problèmes par abstraction, ou encore l'introduction de types de données symboliques pour la représentation «en compréhension» des entités manipulées. D'un point de vue pratique on doit également s'interroger sur l'introduction «la plus harmonieuse possible» de ces méthodes dans un cycle de développement logiciel (par exemple, dans notre cas, pour les programmes Esterel).

3.4 Programmation réactive

Participants : Frédéric Boussinot, Jean-Ferdy Susini.

Mots clés : réactif, objet, script, parallélisme, événement, World Wide Web, Reactive-C, Java, SugarCubes, Icobj, 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 du parallélisme (« concurrency ») dans les langages de programmation usuels reste très empirique et fondé sur des concepts datés. Par exemple, Java ne propose que des sémaphores et moniteurs 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'évènement, 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 [2].

Les SugarCubes sont un ensemble de classes Java pour la programmation réactive [9]. 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 [22] 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.

4 Domaines d'applications

4.1 Télécommunications

Mots clés : télécommunications, programmation réactive, mobilité.

C'est naturellement un domaine important pour les systèmes distribués, les systèmes réactifs et la communication par messages. 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. Un langage de description serait utile, par exemple, pour servir de support formel, exempt d'ambiguïtés, à la spécification de standards normalisés, comme c'est actuellement le cas, pour des systèmes qui n'incluent pas de mobilité, avec SDL ou Lotos. L'objectif à terme est de contribuer à clarifier les modèles informels existants (CORBA, ODP), et de fournir la base d'un langage expérimental pour la mobilité. Nos travaux dans ce domaine ont fait l'objet d'une collaboration avec le CNET, 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 Programmation réactive synchrones de protocoles

Les téléphones ou postes de radio portables du futur vont devenir de véritables systèmes multimédia capables d'allier son, images, navigation Internet, consultation de base de données, et relais de communications simultanées sur différentes gammes d'ondes. Ils auront leurs protocoles dédiés, avec normalisations UMTS, et des algorithmes de cryptage et d'évasion de fréquence. Les protocoles devront être téléchargeables. Nous collaborons avec Thomson sur l'utilisation d'Esterel pour programmer des piles protocolaires radio. Le projet Rodéo de l'INRIA conduit aussi des expériences sur l'utilisation d'Esterel pour des protocoles de nature similaire.

4.1.3 IHM graphiques

L'utilisation des icobjs apparait 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.

4.1.4 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 Cnet pour l'étude et la réalisation de ce type de d'infrastructure d'exécution. Un des objectif 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.

4.2 Systèmes embarqués

Mots clés : système embarqué, transports, protocole, programmation réactive, contrôleur, avionique.

Les systèmes embarqués étant souvent critiques pour leur fiabilité, il est essentiel de les développer et de les valider avec des méthodes formelles de programmation et de vérification. Par leur parallélisme inhérent de programmation, les langages synchrones permettent de s'affranchir de la gestion dynamique de tâches parallèles telle qu'on la trouve dans les systèmes classiques comme les OS temps-réel et qui est difficile à maîtriser. Le déterminisme des programmes parallèles Esterel permet une mise au point et une vérification beaucoup plus simple. Nous collaborons activement avec Dassault Aviation sur ces thèmes [7], ce qui a constitué le moteur principal des améliorations du langage Esterel par retour d'expériences au cours des dix dernières années. Nous débutons également des collaborations avec Texas Instruments sur la conception de parties de microprocesseurs dédiés pour téléphones cellulaires.

4.3 Synthèse de circuits

Mots clés : circuit, contrôleur, matériel.

Les circuits matériels deviennent de plus en plus complexes, surtout en ce qui concerne le contrôle des chemins de données (pipeline, cohérence de caches, interfaces bus, etc.). Esterel est adapté à la synthèse efficace de contrôleurs matériels. Nous travaillons sur ce sujet avec la société américaine Synopsys, qui a acquis une licence source Esterel, et dont nous sommes en train d'acquérir la plate-forme de CAO électronique par le biais de l'opération européenne *EuroPractice*. Nous avons également entamé des collaborations avec Texas Instruments ou Motorola (Suisse) sur le développement de modèles de circuits DSP (Digital Signal Processors).

4.4 Conception conjointe matériel / logiciel

Mots clés : système embarqué, matériel.

Les systèmes embarqués sont souvent faits de composants mixtes matériel / logiciel dont la conception doit être conjointe. Les langages synchrones sont bien adaptés à ce problème, car ils peuvent être compilés indifféremment sur des cibles matérielles ou logicielles. Nous travaillons à l'utilisation d'Esterel dans les environnements de conception conjointe développés à U.C. Berkeley et chez la société Cadence Design Systems, qui a des actions très fortes dans ce domaine explosif. En particulier, nous avons réalisé avec Cadence des implantations distribuées d'Esterel sur le système POLIS, et cette compagnie prévoit d'utiliser dans des produits de codesign futurs des langages fortement inspirés d'Esterel, mais à la syntaxe influencée par les langages C (comme ECL, «Esterel C Language») ou Java (comme Jester, «Java-Esterel»). Ces actions se développeront principalement dans le contexte d'un nouveau projet RNRT nommé Syntel.

5 Logiciels

5.1 Esterel et son environnement

Participants : Gérard Berry, Amar Bouali, Yannis Bres, Xavier Fornari [correspondant], Loïc Henry-Gréard, Jean-Paul Marmorat, Dumitru Potop, Annie Ressouche, Valérie Roy.

Mots clés : Esterel, compilateur, synthèse, optimisation, causalité, interface graphique.

Résumé : *Le compilateur Esterel v5 traduit les programmes Esterel en circuits ou programmes C, en pleine conformité avec la sémantique constructive. L'environnement de développement comporte également un simulateur-débogueur Xes, des optimiseurs et un logiciel de vérification automatique de propriétés Xeve.*

Le compilateur Esterel v5 consiste en plusieurs processeurs permettant de produire des codes objets pour des cibles matérielles ou logicielles. L'environnement de programmation contient également un simulateur-débogueur graphique permettant de mettre au point les programmes, des optimiseurs spécialisés fondés sur des techniques de calcul booléen, et des interfaces vers des systèmes de vérification automatique de propriétés.

Le compilateur a été conçu de façon préindustrielle pour offrir de bonnes performances et une grande robustesse. Il est diffusé sur le Web en version binaire d'évaluation à l'URL `http://www.inria.fr/meije/esterel/`. Les sociétés Dassault Aviation et Synopsys en ont acquis des licences sources. Il est désormais industrialisé et diffusé par la société Simulog, qui développe également autour de cette base un formalisme de représentation graphique, issu à l'origine des travaux de Charles André de l'I3S, et nommé SyncCharts. Un GIE dénommé «Esterel Valley» est en cours de constitution pour fédérer les efforts de notre groupe de recherche et de cet industriel dans cette perspective de développement.

5.2 Outils de vérification Fc2tools

Participants : Amar Bouali [correspondant], Michel Bourdellès, Robert de Simone, Annie Ressouche, Valérie Roy.

Mots clés : vérification automatique, bisimulation, BDD, abstraction, minimisation, réduction, équivalence comportementale, observateur, sûreté, vivacité, logiciel fiable.

Résumé : *Cette boîte à outils forme la nouvelle génération du logiciel Auto/Graph. Elle permet l'analyse par modèles des systèmes parallèles communicants, synchrones ou asynchrones. Elle repose largement sur des techniques d'analyse symbolique à base de BDDs.*

Les modules logiciels rassemblés dans Fc2Tools autorisent l'édition graphique, la réduction compositionnelle (par abstraction et minimisation d'états), ainsi que la comparaison de systèmes parallèles communicants. Ils permettent également d'extraire des chemins d'exécutions comme contre-exemples éventuels de propriétés d'absence de blocage ou de fatalité. Une méthodologie basée sur la mise en parallèle d'observateurs opérationnels avec le système permet de ramener la validation d'autres propriétés à ces cas simples.

Les outils utilisent pour une part des méthodes *symboliques*, à base de diagrammes de décision binaires, pour la représentation implicite d'espaces d'états. D'autres modules opèrent sur des représentations explicites de machines à états finis, dans une approche de minimisation compositionnelle. Les systèmes considérés peuvent être des spécifications de systèmes parallèles communicants exprimés dans une algèbre de processus (comme CCS ou Lotos), ou des programmes réactifs synchrones (provenant principalement d'Esterel). Dans le second cas une interface graphique conviviale Xeve permet une activité d'analyse simple pour les utilisateurs d'Esterel non spécialistes.

Les développements BDD se fondent sur la bibliothèque TiGer, initialement développée à DEC-Prl et commercialisée par la société Xorix.

Nos outils de vérification, ainsi que des descriptifs et leur documentation technique, sont disponibles par ftp ou sur Internet à l'URL `http://www.inria.fr/meije/verification/`.

5.3 Reactive C, SugarCubes, Icobjs

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

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

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 les Scripts réactifs 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.inria.fr/meije/rc/SugarCubes`.
Une comparaison entre les threads Java et les SugarCubes est disponible en `http://www.inria.fr/meije/rc/SugarCubes/ComparisonThreadsSugarCubes`.

5.3.1 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.inria.fr/meije/rc/Softwares` L'interpréteur de Scripts réactifs au dessus de Java est disponible en `http://www.inria.fr/meije/rc/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.

5.3.2 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 applet Java et introduit une démonstration logicielle ludique dans laquelle on programme dynamiquement le comportement d'un petit robot. Cette démonstration, nommée WebIcobj, est disponible en `http://www.inria.fr/meije/rc/WebIcobj`.

6 Résultats nouveaux

6.1 Compilation et optimisation modulaire d'Esterel

Participants : Gérard Berry, Amar Bouali, Loïc Henry-Gréard, Annie Ressouche, Robert de Simone.

Mots clés : Esterel, modularité, optimisation, compositionnalité.

Le compilateur Esterel v5 actuel compile d'un seul coup toute une application. Cette compilation monolithique trouve actuellement ses limites, en particulier pour les gros programmes écrits par Dassault Aviation. Nous avons étudié théoriquement de nouveaux algorithmes de compilation modulaire, qui découlent directement des propriétés mathématiques de la sémantique constructive du langage.

Une part importante de la chaîne de compilation consiste en l'optimisation séquentielle des systèmes d'équations booléennes obtenus comme format intermédiaire. Ces optimisations utilisent des techniques puissantes provenant du domaine de la synthèse de circuits digitaux, et pouvant réclamer le calcul symbolique de l'espace des états atteignables. Ce calcul est coûteux sur des systèmes complexes, et il peut se révèler fort payant (en temps et espace) d'optimiser des sous-composants avant de les réassocier dans un contexte de programme plus large. Ces travaux ont demandé une refonte importante des processeurs concernés de l'environnement de développement Esterel, et ils ont même fait naître des nouvelles problématiques de recherche, comme le montre l'exemple suivant.

Tout sous-composant Esterel peut se décomposer en deux parties : la *surface*, ou code implantant le comportement à l'instant initial, et la *profondeur*, encodant la suite des comportements. Ce découpage, a priori arbitraire, prend son sens dans une approche de compilation modulaire où la surface peut se voir dupliquée en plusieurs instances coexistant au même instant du fait des capacités de préemption et de redémarrage de sous-modules dans un programme contextuel. Par contre la profondeur reste unique et ne sera jamais dupliquée lors de

l'instanciation. Il est donc important de garder ces entités séparées. Mais la surface définit les possibilités d'affectation de registres déterminant la suite des comportements, et une optimisation négligeant ces aspects se révèle généralement décevante en efficacité. Il faut donc savoir optimiser la profondeur *sous la contrainte* de la surface, et définir des extensions des algorithmes dans ce cadre.

Ce travail a été effectué partiellement dans le cadre du projet LTR Esprit Syrf [20].

6.2 Modélisation d'un décodeur de longueurs d'instructions en Esterel

Participants : Gérard Berry, Loïc Henry-Gréard.

Mots clés : Esterel, modularité, optimisation, compositionnalité.

La recherche de spécifications de haut niveau *synthétisables* pour les circuits électroniques est devenue critique pour l'industrie de la conception de hardware, confrontée à l'augmentation exponentielle de la complexité de ses produits, et à une baisse en conséquence de la productivité des concepteurs. Dans le cadre, en particulier, des recherches de la société Intel (Strategic CAD Labs), et pour la synthèse des parties contrôle des circuits intégrés, l'utilisation d'Esterel est envisagée. Une coopération entre Intel SCL, l'INRIA et le CMA a ainsi été engagée, matérialisée par un séjour de quatre mois aux Etats-Unis de Loïc Henry-Gréard.

Au cours de ce séjour, l'utilisation du compilateur Esterel pour la synthèse de hardware a été expérimentée. En particulier, les performances d'optimisation dans le cadre de l'optimisation modulaire ont été évaluées. Le traitement modulaire des spécifications est indispensable dans la description de hardware, en raison de la taille des problèmes. La caractérisation modulaire du *timing* des circuits dans le cadre des outils DESIGN COMPILER de Synopsys a été faite. Nous avons aussi hérité d'un cas d'étude intéressant inspiré par Intel : un décodeur de longueurs d'instructions, qui est un programme présentant des cycles combinatoires constructifs, et sur lequel les outils actuels d'analyse de cycles d'Esterel butent encore.

6.3 Implémentation distribuée d'Esterel

Participants : Gérard Berry, Ellen Sentovich.

Nous avons étudié les perspectives de distribution d'un code Esterel synchrone sur une architecture asynchrone mais très spécialisée, correspondant à des systèmes proches du matériel et dans laquelle les communications entre zones prurement synchrones se font par l'intermédiaire de tampons de taille unitaire, sans test d'écrasement des données précédentes lors d'écritures. Ce travail est décrit dans [8].

6.4 Production de code efficace à base d'équations booléennes

Participants : Gérard Berry, Dumitru Potop.

Mots clés : système d'équations booléennes.

Dans le cas des spécifications de grande dimension il est important de produire du code séquentiel avec un bon rapport vitesse/taille sans avoir à construire l'automate associé. La solu-

tion adoptée actuellement consiste à simuler, par du code logiciel, le fonctionnement du circuit produit par Esterel, en évaluant l'ensemble des variable du système d'équations booléennes à chaque réaction de comportement. Elle a le désavantage d'explorer des parties inactives du programme. L'objectif est ici de structurer le code produit pour n'effectuer dans la mesure du possible que les évaluations nécessaires à la déduction correcte du comportement, suivant la sémantique du langage.

Pour réaliser cela, nous nous sommes concentrés sur le code intermédiaire SC. Le code SC représente les spécifications synchrones Esterel sous la forme de circuits séquentiels, constitués de registres et de portes logiques. En plus, l'information sur la structure du programme Esterel initial est présente sous la forme de décorations du circuit, mais elle n'est pas actuellement exploitée.

Nos investigations ont porté dans deux directions :

1. la structuration du code SC qui nous permettra d'exécuter à chaque instant un nombre minimal d'instructions;
2. les optimisations du code SC en utilisant des informations sur la structure initiale du programme Esterel.

Nous avons défini une procédure de structuration pour le code SC qui donne des bons résultats sur des exemples de tests. Cette représentation permet l'application de bonnes politiques de planification pour l'exécution séquentielle de la spécification SC. Des optimisations du code SC avec des bons résultats ont aussi été définies.

6.5 Compilation mixte automates/circuits

Participants : Gérard Berry, Yannis Bres.

Mots clés : automates, circuit, compilation.

A partir de la version 4 du compilateur Esterel les automates d'états finis, jusqu'alors formalisme pivot du compilateur, ont été délaissés au profit des circuits séquentiels. De fait, les automates explosent rapidement, à la fois en taille nécessaire pour leur représentation et en temps requis pour leur génération. Néanmoins, une communauté assez importante d'utilisateurs d'Esterel, qui manipulent des programmes de taille raisonnable, demeure concernée par la génération d'automates. Dans la version 4, les automates étaient donc générés à partir de circuits séquentiels, vus comme un ensemble d'équations logiques, qu'il fallait ordonner ; une fois triées, les équations etaient évaluées séquentiellement. Appliqué à des circuits cycliques, le tri des équations nécessite une étude approfondie de la causalité du circuit, ce qui peut se révèler très lourd. Dans la version 5, ce compilateur n'a pas evolué, alors qu'a été introduite la sémantique constructive d'Esterel, qui permet un traitement formel des cycles. Nous avons donc développé un nouveau compilateur générant des automates directement à partir de circuits séquentiels, qui traite indifferement les circuits cycliques et non-cycliques, pour peu qu'ils soient « corrects ». Basé sur une approche des circuits en tant que constructions électroniques, et non plus en tant qu'ensembles d'équations, ce compilateur ne nécessite plus ni tri ni décyclisation des équations. Même sans prendre en compte la surcharge auparavant nécessaire pour

les circuits cycliques, le nouveau compilateur s'avère souvent plus performant. Basé sur une utilisation efficace de techniques de hash-code avec caches, il permet l'adoption de différentes stratégies visant à optimiser à la fois le temps de compilation et la qualité des automates produits.

6.6 Génération de tests à partir de méthodes formelles

Participants : Amar Bouali, Robert de Simone.

Mots clés : génération de tests, simulation exhaustive, modèle de référence.

Cette recherche a été initiée dans le cadre d'une collaboration naissante avec Texas Instruments (Villeuneuve-Loubet, près de Sophia [12].

La conception actuelle de microprocesseurs de type DSP pour les téléphones cellulaires chez cet industriel s'effectue à la base à l'aide d'un modèle de référence assez exhaustif, programmé en C, modélisant tous les aspects de l'architecture du système (ou de la gamme de systèmes). Ce modèle est très précis, fidèle au cycle d'horloge, et fait foi contre toute implémentation ultérieure en circuit, à base de code VHDL reencodé manuellement. À cette fin, un jeu de tests de taille impressionnante a été assemblé, pour tester les fonctionnalités dans le détail. Le modèle logiciel (la référence C) fournit les réponses attendues, qui sont comparées aux résultats de test sur implémentation.

La démarche demande à être améliorée sur plusieurs points. Tout d'abord, même si un soin particulier est apporté au codage, la spécification en C reste informelle et le langage Esterel semble bien mieux à même de fournir une syntaxe adaptée de haut-niveau pour certaines parties critiques du modèle, en apportant sémantique formelle et vérification de surcroît; ceci a été validé par la reprogrammation de composants de ce modèle de référence, qui se sont révélés bien plus compacts, lisibles et réutilisables. Ensuite, on aimerait définir pour les jeux de tests une notion de *bonne* couverture des états de configurations possibles du systèmes, et plus seulement des fonctionnalités. Ceci est rendu possible par les techniques d'analyse et de «model-checking» traditionnellement associées aux méthodes formelles disponibles sur Esterel.

Nous avons défini un certain nombre de méthodes pour extraire une représentation symbolique de l'espace d'états couramment atteint par le jeu de test sur le sous-composant décrit formellement en Esterel, pour le comparer avec l'espace d'états symboliques *atteignables*, et enfin pour compléter le jeu de test par des ajouts les plus compacts possibles (en nombre de tests et en longueur de tests) afin d'améliorer la couverture et de rapprocher ces deux ensembles. Enfin, comme il s'est avéré que les états accessibles dans le sous-composants pouvaient ne pas tous l'être dans le système global, du fait du comportement du reste du système vu comme environnement du sous-composant décrit en Esterel, nous avons intégré la notion d'observateur et la projection d'états dans le vérifieur Xeve afin de pouvoir modéliser ce phénomène.

Sur l'exemple de taille industrielle de notre partenaire la couverture d'états est rapidement grimpée de 30% à 90%, puis à presque 100% grâce au travail sur la modélisation d'environnements. Ces travaux ont été très bien reçus au sein du partenaire industriel, et devraient mener à des collaborations étendues.

6.7 Compilation des langages réactifs synchrones en Vhdl/Verilog

Participants : Xavier Fornari, Annie Ressouche.

Mots clés : langages réactifs synchrones, verilog, vhdl.

Les langages de description de circuits ont été introduits pour aider au développement d'applications de plus en plus complexes dans le domaine de la conception de circuits intégrés. Ce ne sont pas de vrais langages de programmation mais des langages de description adaptés à une méthodologie descendante : à partir d'un niveau de spécification comportementale (où les contraintes matérielles liées au circuit sont abstraites), des outils de synthèses et des simulateurs transforment le code à un niveau structurel (où les caractéristiques du circuit sont prises en compte). Enfin, la dernière transformation consiste à générer automatiquement la réalisation.

Par contre, les langages réactifs synchrones, qu'ils soient orientés flots de données comme Lustre et Signal ou de style impératif comme Esterel, sont de vrais langages de programmation relativement proches du niveau comportemental de description. Ils sont bien adaptés à la spécification d'applications hardware et ils apportent le bénéfice d'une sémantique claire et l'utilisation d'outils de simulation au niveau comportemental et de vérification pour éventuellement engendrer automatiquement des séquences de tests. Sur un autre plan, le domaine de la vérification formelle a été largement exploré pour les langages de description de circuits et des outils adaptés comme SMV ou FORMALCHECK ont été développés. Aussi, en compilant les langages synchrones en VERILOG/VHDL, nous pouvons utiliser ces outils pour prouver les programmes réactifs.

Lustre, Signal et Esterel partagent un certain nombre de notions de base et un format commun DC a été défini afin de permettre la combinaison des différents formalismes ainsi que le partage d'outils (vérifieurs, simulateurs, producteurs de code,...). À partir de cette plate forme commune, nous générons du code VERILOG ou VHDL, et ainsi nous nous interfaçons avec les outils de description de circuits. Mais inférer du code synthétisable à partir d'une description comportementale n'est pas toujours évident, dans la mesure où un certain nombre de constructions de haut niveau n'ont pas de corrélation au niveau structurel. La compilation en VERILOG/VHDL doit donc respecter un certain nombre de règles afin de produire un code synthétisable. Toutefois, il n'y a pas de standardisation des constructions acceptées ou rejetées, ceci dépendant fortement de l'outil de synthèse utilisé. On doit donc envisager de générer plusieurs formes de code dépendant des outils visés. Pour valider ces traductions, nous sommes en train d'acquérir les outils de synthèse de SYNOPSYS, «leader mondial» de ce marché, par le biais de l'initiative européenne EUROPRACTICE.

6.8 Prototypage rapide d'extensions au langage Esterel

Participante : Valérie Roy.

Mots clés : Esterel, analyse syntaxique, prototypage rapide.

Le compilateur Esterel est désormais un logiciel important et fortement utilisé, dans lequel toute modification a des conséquences parfois difficilement prévisibles. Certaines parties du code sont en outre désormais anciennes. Il semblait donc souhaitable d'initier la création d'un

environnement léger pour le prototypage rapide de nouvelles extensions et la validation d'idées de recherche et d'algorithmes «alternatifs». Ce travail a commencé par la réalisation d'un analyseur syntaxique du langage écrit en Java, et bientôt disponible librement. Cet outil devrait bientôt servir de plate-forme de prototypage rapide pour introduire en particulier des notions de type de données spécifiques à la description de circuits hardware.

6.9 Sémantique et Expressivité

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

On décrit ici les travaux qui concernent le pouvoir expressif des modèles de la mobilité, et plus généralement les questions de sémantique abordées par les chercheurs du projet (d'autres travaux plus spécifiques sont évoqués ci-dessous).

Le pouvoir expressif du π-calcul est mis en œuvre dans un travail de Davide Sangiorgi avec Christine Röckl (TU Munich) [17] (soumis pour publication), où une traduction d'un langage impératif, parallèle et d'ordre supérieur (« Concurrent Idealised ALGOL ») est donnée. La correction opérationnelle de cette traduction étant établie, on peut utiliser les méthodes de preuves fournies par la théorie de la bisimulation pour le π-calcul afin d'établir quelques égalités de programmes dans le langage source, égalités dont la preuve directe est problématique.

D. Sangiorgi a également écrit des notes [18] sur l'utilisation des types pour raisonner sur les systèmes concurrents, comme support pour le *tutoriel* qu'il a présenté à la conférence ETAPS'99.

Dans [15], Massimo Merro étudie des caractérisations d'équivalences contextuelles par systèmes de transitions étiquetées dans le π-calcul asynchrone, en l'absence de tests d'égalité sur les noms. Cette étude a été motivée par l'observation que les opérateurs de test sur les noms invalident beaucoup d'optimisations et de transformations de programmes. En application de cette théorie, il est montré que la *mobilité externe* (communication de noms globaux) peut s'exprimer en termes de *mobilité interne* (communication de noms locaux).

Dans le rapport de recherche [21], Silvano Dal Zilio présente une équivalence pour le calcul bleu, une extension du pi-calcul et du lambda-calcul proposée par Gérard Boudol. Cette équivalence se base sur la congruence à barbes, et satisfait les lois de réplication. Par exemple, il est montré qu'une « ressource répliquée », partagée par plusieurs processus, peut être copiée et distribuée de façon sûre.

Roberto Amadio, Gérard Boudol et Cédric Lhoussaine ont étudié dans [10] un π-calcul réparti, avec des primitives explicites pour les localités et la migration. Ils montrent qu'on peut, par une analyse statique très simple, assurer la propriété de réceptivité des noms de canaux, c'est à dire le fait qu'un récepteur est à tout moment disponible pour chaque canal privé ou déclaré dans l'interface d'un processus. Ceci garantit, pour les processus typables dans un système de types simple, l'absence de blocage local, au sens où tout message – migrant ou non – est assuré de trouver un récepteur dans sa localité de destination (« livrabilité » de messages). Cette propriéte n'est pas obtenue au détriment de la puissance d'expression du calcul : il est montré, en effet, que le π-calcul réparti réceptif contient, modulo la bisimulation asynchrone, le π_1-calcul (le π-calcul avec récepteurs uniques introduit précedemment par Amadio). Une série

d'exemples de programmation d'agents mobiles illustre la programmation en style réceptif.

Roberto Amadio, en collaboration avec Sanjiva Prasad (IIT Delhi), a continué l'étude de la partie du protocole Internet (version 6) destinée à la gestion de la mobilité. Une version à états finis du protocole a été simulée dans SPIN. Ce travail va paraître dans [6].

R. Amadio a également étudié avec S. Prasad l'utilisation de calculs de processus pour modéliser des protocoles de sécurité et fournir un cadre où l'on puisse vérifier automatiquement certaines propriétés de sécurité telles que la confidentialité et l'authenticité. Leur approche conjugue les techniques de vérification opérationnelles des calculs de processus avec les analyses d'atteignabilité utilisées traditionnellement pour la vérification de protocoles de sécurité. Quelques résultats de ce travail sont exposés dans l'article [11].

Le travail entrepris par Ilaria Castellani avec Madhavan Mukund et P.S. Thiagarajan du SMI de Madras (dans le cadre du projet franco-indien du CEFIPRA) sur la décomposition en produit de systèmes de transitions avec indépendance a été partiellement mené à bien, aboutissant sur la publication [14]. Dans ce travail on donne une caractérisation des systèmes de transitions qui sont *isomorphes* à un produit d'automates communicants. On a fait aussi des avancées significatives sur la question de caractériser les systèmes de transitions qui sont *bisimilaires* à un produit d'automates. Des résultats complets et effectifs ont été obtenus pour le cas d'un produit déterministe. On a aussi réuni plusieurs éléments montrant que ce problème est difficile dans le cas général (non-déterministe); il pourrait même se révéler indécidable. Ces résultats ont potentiellement des applications importantes dans la vérification automatique de propriétés de réseaux d'agents communicants. Dans ces systèmes, connaître la structure de l'espace d'états peut aider à réduire l'espace de recherche des algorithmes de vérification, diminuant ainsi la complexité de la vérification automatique.

6.10 Objets et mobilité

Participants : Silvano Dal-Zilio, Massimo Merro, Davide Sangiorgi.

Mots clés : objet, migration.

Dans sa thèse soutenue en juillet 99 [5], Silvano Dal Zilio étudie une version du calcul bleu de Boudol, étendu avec des enregistrements. Il montre que le calcul bleu fournit une bonne notation pour la conception d'un langage de programmation concurrent et typé, combinant des traits impératifs, d'ordre supérieur, et à objets. Il étudie notamment comment la programmation fonctionnelle et la programmation à objets sont modélisées, et la généralisation à ce calcul étendu des notions de typage polymorphe et d'héritage.

G. Boudol et S. Dal Zilio ont présenté dans [13] une traduction du calcul d'objets de Fisher-Honsell et Mitchell avec sous-typage dans un lambda-calcul avec enregistrements extensibles. Le système de types utilisé pour la traductiion est une extension du système F_ω des types dépendants par un opérateur de récursion, des enregistrements extensibles et une forme de quantification bornée. Ils ont montré que leur traduction est adéquate, que les règles de typage du calcul de Fisher-Honsell et Mitchell peuvent se dériver simplement à partir de leur codage, et que le système de types proposé vérifie la propriété de préservation du typage.

Massimo Merro, en collaboration with H. Hüttel, J. Kleist, and U. Nestmann (BRICS Aalborg) a étudié la correction de la migration d'objets dans les langages de programmation distri-

buée et orientée-objets. Plus spécifiquement, dans le travail [16], les auteurs se concentrent sur le langage distribué et orienté-objets Obliq de Cardelli. Dans un calcul distribué orienté objet, on considère que la migration d'un objet est correctement implémentée si elle est transparente pour les utilisateurs de l'objet. Dans [16], les auteurs montrent que la migration d'objets en Obliq n'est pas transparente en ce sens, relativement à une notion raisonnable d'équivalence.

6.11 Programmation réactive distribuée

Participants : Frédéric Boussinot, Jean-Ferdy Susini, Navid Nikaein.

Mots clés : distribution, réactif, Java, SugarCubes, Applets.

6.11.1 SugarCubes v2

La version 2 des SugarCubes [19] est maintenant disponible sur le Web avec ses sources. Elle introduits 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.

6.11.2 Le noyau réactif Junior

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 réécritures conditionnelles. Junior peut être vu comme le noyau des SugarCubes. Nous avons donné plusieurs implémentation de Junior, l'une d'entre elle étant capable de traiter un grand nombre de composants parallèles et d'événements (de l'ordre de plusieurs milliers).

Ce travail est décrit dans un rapport de recherche [22].

6.11.3 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 fait l'objet d'un rapport disponible sur le Web et soumis à publication.

6.11.4 RAMA

Sur la base des nouvelles notions introduites par les SugarCubes v2, nous avons exploré la notion d'agent réactif migrant et réalisé un petit prototype appelé RAMA [23]. RAMA propose un ensemble de primitives pour programmer des scripts réactifs se déplaçant sur des architectures en anneaux.

7 Contrats industriels (nationaux, européens et internationaux)

7.1 CTI CNET: 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.

7.2 Machine répartie virtuelle et langage pour objets mobiles

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

Ce projet RNRT d'une durée de 3 ans, qui a débuté en décembre 1998, inclut comme partenaires France Télécom CNET (coordinateur), 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 Synopsys: synthèse de circuits

Participants : Gérard Berry, Amar Bouali, Xavier Fornari.

Ce contrat, passé par la composante CMA / Ecole des Mines du projet Meije, porte sur l'utilisation d'Esterel pour la synthèse de circuits. La société Synopsys de Mountain View (Californie) est le leader mondial sur ce sujet. Elle a acquis une licence source du compilateur Esterel, et nous travaillons en commun sur les algorithmes de synthèse et d'optimisation.

7.4 Cadence: conception conjointe et globale

Participants : Gérard Berry, Xavier Fornari, Annie Ressouche, Ellen Sentovich.

Ce contrat est passé par la composante CMA / Ecole des Mines du projet Meije avec la société américaine Cadence, qui est le leader de la CAO de circuits et s'oriente fortement sur la conception conjointe matériel / logiciel et la conception globale de systèmes. Le CMA est maintenant un pôle du laboratoire Européen de Cadence. Le contrat porte sur l'utilisation d'Esterel pour la conception conjointe et globale et sur la conception de nouveaux formalismes synchrones comme ECL (Esterel-C) et de leurs compilateurs.

7.5 Simulog: industrialisation d'Esterel et SyncCharts

Participants : Gérard Berry, Amar Bouali, Xavier Fornari, Annie Ressouche, Robert de Simone, Valérie Roy.

Ce contrat est passé par la composante CMA / Ecole des Mines du projet Meije avec la société Simulog qui commercialise les outils Esterel. Simulog développe également le système SyncCharts originellement construit par C. André de l'Université de Nice, et qui est un Esterel visuel. Le laboratoire intervient dans le soutien à l'industrialisation, les formations, et les nouveaux développements conjoints en relation avec les utilisateurs.

Dans le futur, notre collaboration avec Simulog va prendre la forme d'un GIE, nommé «Esterel Valley», associant nos trois organismes (Simulog, CMA/Ecole des Mines, Inria). L'objectif est de permettre un transfert technologique fluide et de présenter aux utilisateurs présents et futurs une structure à la fois industrielle mais également réactive à leurs besoins d'extensions nécessitant de nouvelles recherches avancées. Un des premiers effets de cette création sera la mise en commun d'un certain nombre de moyens contractuels, comme ceux obtenus dans Syntel, un projet RNRT récemment labellisé qui devrait débuter en janvier prochain.

7.6 Schneider: utilisation des langages synchrones dans les automates programmables

Participants : Gérard Berry, Annie Ressouche, Valérie Roy.

Ce contrat est passé par la composante CMA / Ecole des Mines du projet Meije avec la société Schneider qui développe et commercialise des automates programmables pour la conduite d'usines ou de procédés. Il concerne la possibilité d'utiliser les langages synchrones comme Esterel et Lustre pour programmer les automates et vérifier des propriétés des programmes. Ce contrat est réalisé en partenariat avec l'Université de Nice (C. André) et VERIMAG (N. Halbwachs).

8 Actions régionales, nationales et internationales

8.1 Actions nationales

8.1.1 Action de développement Genie2

Participants : Conrado Daws, Robert de Simone.

Pour la dernière année de cette collaboration avec la société Dassault Aviation, nous avons travaillé sur les thèmes de la compilation compositionnelle de programmes Esterel et de l'expression facile pour utilisateur non-expert de propriétés de correction.

8.1.2 Action de recherches coordonnées VERDON

Participants : Robert de Simone, Michel Bourdellès.

Nous collaborons dans cette action avec le projet PAMPA et l'avant-projet VASY, ainsi que le laboratoire LSR-IMAG. Le but est d'étudier les techniques avancées de représentation de processus parallèles communicants incluant des données, ainsi que l'algorithmique efficace d'analyse de tels systèmes [4]. Nous avons organisé un séminaire de travail à Sophia en septembre.

8.1.3 Action de développement AEE

Participant : Amar Bouali.

Cette action («Architecture Electronique Embarquée») s'inscrit dans le cadre du programme du même nom soutenu par le ministère de l'industrie et auquel contribuent de grands groupes industriels : GIE PSA-Renault, Aérospatiale, Sagem, Siemens, Valeo, et des équipes de recherche de l'INRIA, de l'IRCyN et du LORIA. L'objectif est de concevoir et valider un processus rapide et sûr pour la définition de l'Architecture Système et le développement des logiciels associés, embarqués. Cette action vise les applications de transport, notamment l'automobile. Nous y avons principalement un rôle de conseil portant sur la manière d'introduire des éléments de spécification comportementale en phase avec l'état de l'art dans ce domaine.

8.1.4 Action de recherches coordonnées PRESYSA

Participant : Roberto Amadio.

Nous collaborons dans cette action avec les projets PROTHEO et EP-ATR, ainsi que le laboratoire VERIMAG. Le but est d'étudier la vérification de systèmes synchrones et asynchrones ayant un nombre d'*états infini*, en se concentrant en particulier sur les propriétés de *sûreté* et d'*invariance*.

8.2 Actions européennes

8.2.1 Projet LTR SYRF 22703

Participants : Amar Bouali, Robert de Simone, Annie Ressouche.

Ce projet est le lieu principal de collaborations entre les équipes de recherche internationales actives dans le domaine des formalismes synchrones. Le consortium comprend les projets INRIA Meije et Ep-Atr, le laboratoire VERIMAG de Grenoble, le GMD et l'université de Lingkoping (Suède), ainsi que les compagnies Schneider, SAAB, EDF et Prover Technology (Suède). Nous travaillons principalement à la combinaison de formalismes (*control-flow* et *data-flow*) au moyen de formats d'échange, et à l'extension des techniques d'analyse (introduction de valeurs de données, affaiblissement de l'hypothèse synchrone, connexions avec la synthèse de circuits digitaux, ...).

8.2.2 Working Group Esprit LTR Confer2

Participants : Gérard Boudol, Ilaria Castellani, Silvano Dal-Zilio, Massimo Merro, Davide Sangiorgi, Roberto Amadio.

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 est prolongé pour une année, jusqu'en fin 2000.

8.2.3 PROCOPE

Participants : Gérard Boudol, 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 ».

8.2.4 Coopération Franco-Portugaise

Participants : Gérard Boudol, Ilaria Castellani, Silvano Dal-Zilio.

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, Silvano Dal-Zilio.

Ce projet, intitulé « Spécification et vérification de systèmes distribués : systèmes à états finis et d'ordre supérieur », a été reconduit jusqu'en mai 2000. Un travail en commun a été entrepris entre M. Mukund, P.S. Thiagarajan (du SMI de Madras) et notre équipe sur une modélisation de systèmes à états finis basée sur les ordres partiels. Un autre travail se poursuit entre S. Prasad (IIT Delhi) et notre équipe sur l'utilisation de calculs de processus pour la vérification de propriétés de sécurité.

8.4 Visites, et invitations de chercheurs

Nous avons reçu des visites brèves (jusqu'à trois semaines) de : Edward Lee (UC Berkeley), Stephen Edwards (Synopsys Inc.), Christine Roeckl (TU Munich, collaboration PROCOPE), Barbara König (TU Munich, collaboration PROCOPE), Vasco Vasconcelos (université de Lisbonne, coopération franco-portugaise), Stefania Gnesi (CNR Pisa), Uwe Nestmann (université

de Aalborg), Madhavan Mukund (SMI Madras, coopération franco-indienne), et Matthew Hennessy (université de Sussex). Nous avons eu aussi une visite de deux mois de Antonio Ravara (université de Lisbonne, coopération franco-portugaise).

Dans le cadre de notre projet franco-indien, Sanjiva Prasad a visité l'université de Provence pendant un mois. Michele Boreale (Florence) a effectué un sejour d'un mois dans la même université en tant que Professeur invité. Ellen Sentovich, de la société américaine Cadence Systems Design, a effectué de fréquents séjours de moyenne durée au sein du projet, dans le cadre du laboratoire commun Cadence - CMA. Cette collaboration porte sur l'utilisation d'Esterel et de ses techniques associées dans le cadre de la conception conjointe logiciel / matériel, et particulièrement dans l'outil logiciel Polis.

9 Diffusion de résultats

9.1 Animation de la Communauté scientifique

De nombreux membres du projet se sont rendus au séminaire de réflexion annuel sur les *Langages Synchrones*, du 29 novembre au 3 décembre, à Hyères. Cette manifestation, qui a rassemblé une cinquantaine de personnes, était coorganisée par l'INRIA Sophia-Antipolis et le CMA/Ecole des Mines.

Plusieurs membres du projet se sont également rendus à la conférence FemSys'99, à Munich. Cette conférence se veut être un lieu de rencontre entre chercheurs et industriels sur les thèmes des systèmes réactifs et des méthodes formelles pour logiciels embarqués. Nous avons présenté des démonstrations logicielles de nos outils. Frédéric Boussinot a fait une communication.

Gérard Berry a fait partie du comité de programme de CAV'99, et a donné une conférence invitée au colloque CHARME'99. Il a visité les laboratoires californiens des sociétés Cadence et Synopsys. Il a participé à la conférence ASIAN'99, à Phuket, Thaïlande, et a enseigné dans l'Ecole d'automne qui suit traditionnellement cet évènement.

Robert de Simone a été élu membre de la Commission d'Évaluation de l'Institut. A ce titre, il a participé à plusieurs jurys de recrutement ou de promotion, ainsi qu'aux séminaires d'évaluation des thèmes de recherche. Il a été rapporteur de la thèse de Christophe Broult (U. de Caen). Il a fait partie du jury de thèse de Michel Bourdellès comme directeur de thèse, et de Nicolas Turro, au titre de directeur de thèse «officiel». Il a fait partie du comité de programme du colloque MSR'99.

Robert de Simone et Annie Ressouche ont participé à plusieurs réunions de travail internes au projet Esprit LTR Syrf, dont la réunion finale se déroulera en janvier 2000 à Grenoble.

Davide Sangiorgi a été membre des comités de programme de l'ICALP'99 et LICS'99. Il a donné une conférence invitée à ETAPS'99 (European Joint Conferences on Theory and Practice of Software), Amsterdam, Mars 1999. Il s'est rendu à Munich (TUM) sur un projet PROCOPE. Il a été rapporteur de la thèse de B. Mammas (Paris 6), et membre du jury de la thèse de D. Hirschkoff (Ecole Nationale des Ponts et Chaussées).

Gérard Boudol a été membre du comité de programme de ESOP'99. Il a participé aux conférences ECOOP99 à Lisbonne, en juin 99, et au colloque associé sur les Systèmes d'Objets Mobiles, où il a fait une présentation. Il a participé également aux conférences FCT99 en Roumanie, en septembre 99, et FST-TCS99, à Madras, en décembre 99, où il a fait des pré-

sentations. Il a fait une présentation invitée au colloque sur les Fondements du Calcul Mobile à Madras, en décembre 99. Cédric Lhoussaine a aussi participé aux conférences ECOOP99 et FST-TCS99.

Ilaria Castellani a co-organisé le workshop EXPRESS'99, dont elle a coédité les Actes [3]. Elle a participé à la conférence ECOOP99 à Lisbonne, en juin 99, et à la conférence CONCUR99 et au workshop EXPRESS99, à Eindhoven, en août 99. Elle a effectué une visite d'un mois à l'université de Florence, au printemps 99, durant laquelle elle a donné des cours en maîtrise et en doctorat. Pendant cette période elle a également présenté des exposés aux universités de Florence, Pise et Bologne. Elle a effectué une visite à l'université de Sussex en octobre 99, dans le cadre de CONFER2.

R. Amadio, G. Boudol, S. Dal Zilio, C. Lhoussaine et M. Merro ont pris part au workshop CONFER2 de Pise, en mars 99. G, Boudol, I.Castellani, C. Lhoussaine et M. Merro ont participé au workshop CONFER2 de Paris, en novembre 99.

Roberto Amadio a organisé (avec Denis Lugiez) l'Ecole «Nouvelles directions en modélisation, test et validation», qui a eu lieu en Avril 1999 au CIRM (Marseille). Il a aussi été membre du comité de programme de HOOTS99 à Paris en Octobre 1999, membre du comité scientifique des Journées Modélisation et Vérification à Besancon en Novembre 1999 et membre du comité de programme du Workshop « Foundations of Mobile Computation » à Madras en Décembre 1999. Il a été conférencier invité de EXPRESS99 en Septembre 1999 à Eindhoven et il a donné un exposé au workshop CONFER2 à Pise. Il a été membre du jury d'habilitation de Laurence Pierre (U. Provence), membre du jury de thèse de Felix Nicoli (U. Provence) et président du jury de thèse de Line Jakubiec (U. Provence).

Massimo Merro a participé à la conférence CONCUR99 et au workshop EXPRESS99, à Eindhoven, en août 99. Il a fait une visite de deux semaines à l'Université de Aalborg (Danemark), en juillet 99.

Silvano Dal Zilio a soutenu une thèse en Informatique de l'université de Nice Sophia-Antipolis [5]. Il a donné des exposés à l'INRIA Rocquencourt (dans le seminaire de sémantique) et à Microsoft Cambridge, avant d'y démarrer un contrat post-doc.

Xavier Fornari s'est rendu 3 mois dans les laboratoires de la compagnie Synopsys à Mountain View dans le cadre d'une collaboration, sur des sujets tenus à confidentialité.

Loïc Henry-Gréard a visité les laboratoires de recherche de la compagnie Intel à Portland pendant 4 mois pour explorer les possibilités de programmation synchrone sur des exemples de décodeur d'instructions de processeurs avancés.

Frédéric Boussinot a participé à plusieurs réunions préparatoires d'une proposition du projet IST PING, dont le but est d'étudier et développer une architecture logicielle pour le déploiement de mondes virtuels distribués de grande taille. France Télécom CNET est le leader de ce projet.

Jean-Ferdy Susini a présenté le papier sur les SugarCubes v2 à la conférence MSR'99 sur la modélisation des systèmes réactifs qui s'est déroulée en mars 1999.

Michel Bourdellès a soutenu une thèse en Informatique de l'Ecole des Mines de Paris [4]. Il a présenté une communication à la conférence MSR'999.

Amar Bouali a participé à une réunion organisée par Intel début novembre pour la formation d'un groupe de travail autour des thèmes du design et de la synthèse de haut niveau de microprocesseurs («High-Level Design, High-Level Synthesis»). Il a présenté dans ce cadre le langage

Esterel, et les évolutions envisagées pour que notre formalisme soit mieux adapté encore à la description de haut niveau de circuits.

9.2 Enseignement

Gérard Berry enseigne dans le cadre de l'ISIA, ainsi qu'à l'Ecole des Mines dans le cadre des cours européens du GEI Paris (Grandes Ecoles d'Ingénieurs), pour un total de 45 h.

Valérie Roy enseigne (environ 60 h) dans le cadre de l'ISIA. Elle participe par ailleurs activement à l'organisation des enseignements et des projets de programmation de cette Ecole.

Xavier Fornari enseigne dans le cadre de l'ESSI (12h TP), de l'ESIM/ISMEA (20h).

Amar Bouali enseigne dans le cadre de l'ISIA (9 h cours, 12 h TD, encadrement de 2 mini-projets de 3 semaines), ainsi qu'au DEA Informatique UNSA (6h de cours dans le module de tronc commun de Charles André).

Annie Ressouche enseigne (environ 32 h) dans le cadre de l'ISIA.

Robert de Simone enseigne à l'ISIA (18h). Il propose une option au DEA Informatique de l'UNSA (pas ouverte cette année).

Davide Sangiorgi enseigne dans le DEA « Sémantique, Preuves et Programmation » à Paris (12h). Il a aussi donné un cours à l'École d'été SNDIS99 (Scuola Nazionale dei Dottorati di Informatica delle Facoltà di Scienze) en Italie ; cette école regroupe tous les étudiants de première année des doctorats en informatique en Italie (20h). Il a donné un cours sur les modèles et les langages pour la mobilité à l'université TU de Munich, en Allemagne (8h).

Ilaria Castellani a donné un cours de 16h sur la sémantique du parallélisme et les algèbres de processus à l'Université de Florence, destiné aux étudiants de maîtrise et aux doctorants. Elle est intervenue également dans le DEA ARP d'Orsay, avec un cours sur les algèbres de processus (6h) en novembre 99.

Roberto Amadio est responsable de la Licence d'Informatique de l'université de Provence. Il enseigne en Licence et Maîtrise d'Informatique. En outre, il assure un cours « Modélisation et Vérification » au DEA d'Informatique de Marseille (15h) et un cours « Parallélisme » en troisième année de l'ESM2 (15h).

Frédéric Boussinot a présenté l'approche réactive à l'ESSI 3eme année (UNSA).

Silvano Dal Zilio a donné des cours à l'ESSI et en DESS.

Yannis Bres enseigne à l'ISIA (12h), et ainsi qu'à l'UNSA (TD Algorithmique et Programmation).

10 Bibliographie

Livres et monographies

[1] G. Berry, *The Constructive Semantics of Pure Esterel*, version electronique, 1999, `ftp://ftp-sop.inria.fr/meije/esterel/papers/constructiveness3.ps.gz`.

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

[3] I. Castellani, B. Victor (éditeurs), *EXPRESS '99: Expressiveness in Concurrency (Eindhoven, The Netherlands, August 23, 1999)*, *ENTCS*, *27*, Elsevier Science Publishers, 1999.

Thèses et habilitations à diriger des recherches

[4] M. Bourdellès, *Analyse de systèmes réactifs synchrones à des fins de vérification. Applications au langage Esterel*, thèse de doctorat, Ecole des Mines de Paris, 1999.

[5] S. Dal Zilio, *Calcul bleu: types et objets*, thèse de doctorat, université de Nice-Sophia-Antipolis, 1999.

Articles et chapitres de livre

[6] R. Amadio, S. Prasad, « Modelling IP mobility », *Journal of Formal Methods of System Design*, 1999, à paraître.

[7] G. Berry, A. Bouali, X. Fornari, E. Ledinot, E. Nassor, R. de Simone, « Esterel: a formal method applied to avionic software development », *Science of Computer Programming*, 2000, accepté pour publication.

[8] G. Berry, E. Sentovich, « An implementation of constructive synchronous programs in POLIS », *Design Automation for Embedded Systems*, 2000, à paraître.

[9] F. Boussinot, J.-F. Susini, « The SugarCubes Tool Box: A Reactive Java Framework », *Software Practice and Experience*, décembre 1998.

Communications à des congrès, colloques, etc.

[10] R. Amadio, G. Boudol, C. Lhoussaine, « The receptive distributed π-calculus », *in : Proceedings FST-TCS'99*, *LNCS*, Springer Verlag, 1999.

[11] R. Amadio, S. Prasad, « The game of the name in cryptographic tables », *in : Proc. ASIAN'99*, *LNCS*, Springer Verlag, 1999. aussi Rapport de Recherche INRIA 3733.

[12] L. Arditi, A. Bouali, H. Boufaied, G. Clavé, M. Hadj-Chaib, R. de Simone, « Using Esterel and Formal Methods to Increase the Confidence in the Functional Validation of a Commercial DSP », *in : FMICS'99 Workshop*, FLoC Conference, Trento, 1999.

[13] G. Boudol, S. D. Zilio, « An Interpretation of Extensible Objects », *in : Proc. FCT'99, - 12th International Symposium on Fundamentals of Computation Theory*, *LNCS*, Springer Verlag, 1999.

[14] I. Castellani, M. Mukund, P.S.Thiagarajan, « Synthesizing distributed transition systems from global specifications », *in : Proceedings FST-TCS'99*, *LNCS*, 1999.

[15] M. Merro, « On Equators in Asynchronous Name-passing calculi without Matching », *in : Proc. Express'99*, *Electronic Notes in Theoretical Computer Science*, *27*, Elsevier, 1999.

[16] U. Nestmann, H. Hüttel, J. Kleist, M. Merro, « Aliasing Models for Object Migration », *in : Proc. EUROPAR'99*, *LNCS*, *1685*, Springer-Verlag, 1999.

[17] C. Röckl, D. Sangiorgi, « A pi-calculus Semantics of Concurrent Idealised ALGOL », *in : Proc. Fossacs'99*, *LNCS*, *1578*, Springer Verlag, p. 306–322, 1999.

[18] D. Sangiorgi, « Reasoning about concurrent systems using types », *in : Proc. Fossacs'99*, *LNCS*, *1578*, Springer Verlag, p. 31–40, 1999.

[19] J.-F. Susini, « Implementation de l'approche reactive en Java : les SugarCubes v2 », *in : Actes MSR'99*, HERMES, 1999.

Rapports de recherche et publications internes

[20] A. Bouali, L. Henry-Gréard, R. de Simone, « Compositional Optimization and Analysis of Esterel programs », *rapport de recherche*, Esprit LTR Syrf deliverable, 1999, soumis à publication.

[21] S. Dal-Zilio, « A Bisimulation for the Blue Calculus », *rapport de recherche n° 3664*, avril 1999.

[22] L. Hazard, J.-F. Susini, F. Boussinot, « The Junior reactive Kernel », *rapport de recherche n° RR3732*, 1999.

[23] N. Nikaein, « Reactive Autonomous Mobile Agents », *rapport de recherche n° rapport de DEA*, DEA RSD UNSA, 1999.