
PROJET MEIJE

Parallélisme, Synchronisation et Temps-Réel

Localisation : *Sophia*

Mots-clés : MEIJE, concurrence, coopération, réactivité, synchronisation, langage fonctionnel, sémantique, mobilité, langage synchrone, temps réel, parallélisme asynchrone, vérification de programme.

1 Composition de l'équipe

Responsables scientifiques

Robert de Simone, directeur de recherche, Inria
Gérard Berry, directeur de recherche, École des Mines de Paris

Responsable permanent

Gérard Boudol, directeur de recherche, Inria

Assistantes de Projet

Gisèle Zimol, vacataire, Inria, jusqu'au 12 janvier 1996
Françoise Martin-Trucas, technicien de recherche, Inria depuis le 15 janvier 1996
Florence Morcamp, École des Mines, jusqu'au 23 décembre 1996
Dominique Micollier, École des Mines

Personnel Inria

Amar Bouali, chargé de recherche, depuis le 1^{er} octobre 1996
Ilaria Castellani, chargé de recherche
Eric Madelaine, chargé de recherche
Annie Ressouche, chargé de recherche
Davide Sangiorgi, chargé de recherche

Personnel École des Mines de Paris

Jean-Paul Marmorat, maître de recherches
Marc Bordier, ingénieur de recherche
Michele Boréale, Conseiller extérieur, jusqu'au 31 décembre 1996
Frédéric Boussinot, maître de recherches
Xavier Fornari, attaché de recherches
Valérie Roy, chargé de recherche
Monica Robert, ingénieur de recherche
Jean-Marc Tanzi, maître-assistant

Ingénieurs experts

Amar Bouali, jusqu'au 31 août 96
Jean-Ferdinand Susini, jusqu'au 30 avril 1996

Chercheurs post-doctorants

Ellen Sentovich, bourse américaine, jusqu'au 30 novembre 1996
Delphine Terrasse, GENIE, jusqu'au 30 septembre 1996

Chercheurs doctorants

Michel Bourdellès, allocataire MENESR
Silvano Dal-Zilio, ENS Lyon
Karima Machane, allocataire CIES
Massimo Merro, allocataire gouvernement Italien, depuis le 1^{er} octobre 1996
Olivier Ploton, allocataire MENESR
Horia Toma, allocataire Ecole des Mines de Paris

Collaborateurs extérieurs

Roberto Amadio, chargé de recherche CNRS, jusqu'au 1er Septembre 1996. Actuellement
Professeur à l'Université de Provence
Monica Robert, attaché de recherche non permanent Armines jusqu'au 30 juin 1996

Autres personnels

Edward Lee, professeur invité, U.C. Berkeley, mai et juin 1996
Uwe Nestmann, étudiant allemand, U. Erlangen, jusqu'au 30 juin 1996
Jean-Ferdinand Susini, stagiaire DEA RSD Nice, 1^{er} avril au 30 octobre 1996
Marlène Esteve, stagiaire DEA MDFI Marseille, de mai à juin 1996
Mounir Alaoui, stagiaire DEA MDFI Marseille, de mai et juin 1996
Charlotte Truchet, stagiaire ENS, de juillet et septembre 1996
Mouloud Bourbel, stagiaire de DEA ENS-Lyon, d'avril à juillet 1996

2 Présentation du projet

Le projet «Parallélisme, Synchronisation et Temps Réel» est un projet commun INRIA-Ecole des Mines à Sophia-Antipolis. Ses activités s'articulent autour de trois thèmes :

- l'étude et la modélisation algébrique des concepts théoriques liés aux systèmes et langages parallèles et répartis,
- l'étude plus spécifique du domaine des formalismes réactifs synchrones, et surtout le développement du langage Esterel et de ses modèles sémantiques, ainsi que du langage RC extension réactive du langage C,
- le développement d'outils d'analyse et de vérification automatique de programmes inspirés des deux thèmes précédents.

3 Actions de recherche

3.1 Sémantique des langages concurrents d'ordre supérieur

Participants : Gérard Boudol, Ilaria Castellani, Silvano Dal-Zilio, Uwe Nestmann, Davide Sangiorgi, Roberto Amadio

Mots-clés : bisimulation, concurrence, lambda-calcul, langage fonctionnel, modélisation, parallélisme asynchrone, processus mobiles, synchronisation, sémantique, vérification de programme.

Nous poursuivons notre étude des langages d'ordre supérieur, qui expriment directement l'idée de mobilité. La mobilité concerne, dans les modèles que nous étudions, soit la topologie du réseau de communication, soit les programmes eux-mêmes. Le premier cas est bien représenté par le π -calcul de Milner, où les canaux de communication sont transmis d'un agent à l'autre. D'un autre côté, dans les langages d'ordre supérieur proprement dit, ce sont les agents qui sont transmis au cours de la communication. Nous divisons, de façon quelque peu arbitraire, la présentation de cette activité en deux parties. L'une concerne la *sémantique de la mobilité*, et l'autre traite de *mobilité et vérification*.

3.1.1 Sémantique de la mobilité

Les langages pour le parallélisme d'ordre supérieur, étant beaucoup plus expressifs que les modèles « classiques » comme les calculs de processus MEIJE ou CCS que nous étudions précédemment, posent des problèmes plus difficiles, aussi bien en ce qui concerne leur fondement sémantique que leur utilisation comme modèles permettant de raisonner et de faire des vérifications sur des applications concrètes. Dans cette section nous décrivons notre travail sur le premier point.

Un problème ouvert depuis quelques années, et qui a été résolu indépendamment et presque simultanément cette année par plusieurs chercheurs, est celui de la *sémantique dénotationnelle* du π -calcul. Contrairement au cas des algèbres de processus classique, la « bonne » notion de bisimulation pour le π -calcul est loin d'être clairement établie, et la notion de modèle l'est encore moins. D. Sangiorgi, en collaboration avec Marcelo Fiore de l'université d'Edimbourg et Eugenio Moggi de Gènes, a résolu ce problème en proposant une construction abstraite de modèle qui est complètement adéquate pour une forme particulière de bisimulation, où un statut explicite est donné aux fonctions qui associent des agents aux noms de canaux [532].

La notion de bisimulation elle-même continue de faire l'objet d'actives recherches. Nous avons en particulier proposé, pour la variante dite « asynchrone » du π -calcul, une nouvelle définition qui paraît mieux adaptée dans ce cadre. L'idée est la suivante : dans un monde asynchrone, tout message est émis sans délai, et cette émission ne bloque aucune autre activité. Par conséquent on peut penser qu'un observateur ne peut, dans ce cas, « contrôler » le processus observé en lui envoyant des messages. Autrement dit, seuls les messages émis – et non les réceptions de message – par le processus testé peuvent être observés. Cette idée a été mise en œuvre dans l'article [523], qui développe la théorie équationnelle de cette notion de bisimulation « asynchrone ». Dans le même ordre d'idées, le papier [536] enrichit l'étude des diverses formes de bisimulations proposées pour le π -calcul, et plus particulièrement du point de vue de leur compatibilité avec les opérateurs du langage. Les auteurs montrent en particulier que la version « synchrone » du π -calcul, où l'émission de message est bloquante, requiert une théorie élaborée, où la relation de causalité entre les actions d'un processus est prise en compte.

Deux autres travaux sont à mentionner dans cette section de sémantique. Le premier, dû à D. Sangiorgi [540], établit que le calcul d'objets typés récemment proposé par Abadi et Cardelli peut être codé dans le π -calcul en respectant les types. Ce n'est pas un résultat a priori évident : par exemple le codage « naturel » du calcul d'Abadi et Cardelli dans un λ -calcul étendu avec des « records » récursif ne préserve pas le typage. Le codage de Sangiorgi, qui utilise un π -calcul étendu avec une construction « par cas », reste au premier ordre, et est donc plus simple que le codage dans le λ -calcul du second ordre établi par Viswanathan (POPL'96). Enfin, dans un article à paraître dans les actes de POPL'97 (Janvier 97), G. Boudol a proposé une synthèse du λ -calcul et du π -calcul asynchrone, où l'émission de message est

interprétée comme l'application du nom du destinataire au contenu du message, tandis que le récepteur est un agent associé à ce nom. Ainsi la réception d'un message correspond à l'opération qui remplace un nom par sa valeur trouvée dans l'environnement. On obtient un calcul où l'on peut, comme en λ -calcul, manipuler directement des données d'ordre supérieur, tout en gardant les facilités de communication du π -calcul asynchrone. Ceci apparaît clairement dans le système de types, qui généralise à la fois celui du λ -calcul (types simples de Church et Curry) et le «*sorting*» du π -calcul donné par Milner. Ce π -calcul généralisé permet aussi de se rendre compte que les diverses traductions de stratégies de réduction du λ -calcul dans le π -calcul ne sont rien d'autre qu'une transcription de diverses transformations CPS («*continuation passing style*») connues depuis longtemps.

3.1.2 Mobilité et vérification

Comme nous l'avons dit, les modèles d'ordre supérieur pour le parallélisme posent des problèmes nouveaux et difficiles en ce qui concerne la vérification. On peut, dans certains cas, se ramener à une approche similaire à celle qui utilise les systèmes de transition finis comme modèle de base (voir la section 3.4). C'est ce qui a été fait par D. Sangiorgi, dans un travail commun avec M. Pistore [533], pour les agents «*à contrôle fini*» du π -calcul. On doit rappeler ici que, en raison du fait que ces agents peuvent recevoir des noms arbitraires, ils déterminent des systèmes de transitions qui ont en général une infinité d'états. Néanmoins, en exploitant le fait que les états ne diffèrent que par les noms reçus en paramètre, D. Sangiorgi et M. Pistore ont montré qu'on peut adapter l'algorithme classique de réduction à un modèle minimal pour vérifier la bisimulation pour ces agents particuliers. Une implémentation prototype a été réalisée en CAML-Light.

Les articles [523] et [536] mentionnés ci-dessus relèvent aussi de l'activité de vérification, mais ne peuvent conduire à des procédures de décision que dans des cas très particuliers. Constatant que les méthodes «*énumératives*» classiques en vérification ne sont plus adaptées au cas de l'ordre supérieur, on peut essayer de généraliser les méthodes «*logiques*» qui se prêtent mieux au traitement d'objets infinis. Dans cet esprit, R. Amadio, dans un travail commun avec M. Dam [524], a étendu la logique modale usuelle (Hennessy-Milner) avec point-fixe au π -calcul, montrant qu'elle est suffisamment expressive pour caractériser les agents modulo [une approximation quelconque de la bisimulation. Il s'est ensuite particulièrement intéressé au problème de la vérification du fait qu'un agent satisfait une spécification donnée (par une formule). Deux méthodes sont envisagées : une méthode algorithmique, qui termine si l'agent considéré satisfait une certaine propriété de finitude, et une méthode déductive (système de preuve formel), qui est complète pour les agents sans récursion.

Un autre travail représentatif de la recherche sur les moyens de raisonner sur le parallélisme d'ordre supérieur est celui qu'a fait D. Sangiorgi avec B. Pierce [539]. Une méthode bien établie dans les langages de programmation pour offrir un mode de «*raisonnement*» élémentaire sur les programmes et éviter des erreurs consiste à utiliser un système de types. Un type est une spécification, ou une propriété certes très grossière du comportement d'un programme, mais qui, du fait de sa simplicité, peut être effectivement vérifiée. De plus, l'usage de types permet d'écarter certaines constructions comme illégales, et ceci permet de démontrer des propriétés ou des équivalences qui ne vaudraient pas autrement. Cette approche du raisonnement et de la vérification dans un cadre typé, et plus généralement dans le cadre d'une logique «*simplifiée*», semble très prometteuse (et assez proche de l'activité de programmation), et le travail de D. Sangiorgi et B. Pierce est un premier pas dans ce sens.

3.2 Esterel et la programmation réactive synchrone

Participants : Gérard Berry, Xavier Fornari, Monica Robert, Olivier Ploton, Ellen Sentovich, Jean-Marc Tanzi, Horia Toma

Mots-clés : langage synchrone, réactivité, causalité, sémantique, temps réel, compilation, optimisation de code, circuit numérique.

Esterel est un langage parallèle synchrone dédié aux systèmes réactifs qu'on trouve dans les applications de contrôle de processus, de protocoles de communication, ou de conception de circuits électroniques. Esterel est développé dans notre groupe depuis 1983. Il se caractérise par une sémantique formelle rigoureuse, un outillage de compilation ou de synthèse de circuits de qualité pré-industrielle, et des interfaces vers des systèmes de vérification de programmes. Esterel est utilisé par plusieurs industriels.

3.2.1 Etude de la causalité

Participants : Gérard Berry, Ellen Sentovich, Delphine Terrasse

Comme dans tous les langages synchrones, la propagation de signaux en Esterel est supposée instantanée. Ceci rend possible d'écrire des programmes paradoxaux exprimant par exemple « j'émet un signal si et seulement si je ne le reçois pas ». La caractérisation exacte de ces programmes paradoxaux est connue sous le nom de problème de la causalité ; c'est le problème central de la sémantique d'Esterel. Nous avons établi le rapport entre la causalité en Esterel, le point de vue constructif en logique mathématique, et la propagation des fronts électriques dans les circuits digitaux. Nous avons d'abord défini la notion de logique booléenne constructive et établi qu'un circuit est bien défini dans cette logique si et seulement si ses fils se stabilisent électriquement quels que soient les délais de ses portes. Nous avons ensuite défini une sémantique constructive intrinsèque d'Esterel et donné une traduction d'Esterel en circuits qui préserve la constructivité [517]. Cette traduction a été prouvée correcte pour un sous-ensemble du langage en utilisant le système de vérification COQ dans le cadre de l'action Génie 4.3. Cet ensemble de résultats indique que la notion de constructivité est probablement la notion de causalité la plus naturelle en Esterel. Par ailleurs, nous avons défini des algorithmes permettant de tester efficacement la constructivité des programmes.

3.2.2 Le compilateur Esterel v5

Participants : Gérard Berry, Xavier Fornari, Horia Toma, Monica Robert

Le point de vue constructif a été complètement implanté dans le nouveau compilateur Esterel v5. Ce compilateur améliore de façon considérable le compilateur Esterel v4 précédent qui était très restrictif sur les problèmes de causalité. Il a été testé intensivement, à l'aide d'une procédure de tests systématique mise en place par M. Robert.

Les algorithmes formels de vérification de la constructivité ont été implantés en utilisant des techniques de calcul booléen par diagrammes de décision binaires (BDDs) disponibles dans la bibliothèque TiGeR acquise par l'INRIA (travail de thèse de H. Toma). Ils ont été utilisés dans des applications industrielles chez Dassault Aviation. Nous avons développé un nouveau système de débogage symbolique permettant en particulier de visualiser les messages d'erreur de causalité. Enfin, nous avons poursuivi le développement de systèmes d'optimisation et de vérification de programmes à l'aide de diagrammes de décision binaires [534] (H. Toma, E. Sentovich).

Pour la suite, le principal problème reste l'amélioration du calcul pratique de la causalité. Si l'interprétation des programmes est toujours réalisable en temps linéaire et ne pose donc aucun problème, leur compilation est intrinsèquement beaucoup plus coûteuse lorsque les circuits générés ont des cycles. Les algorithmes dont nous disposons actuellement nous ont permis de traiter des exemples industriels de grande taille. Ils sont cependant très gourmands en temps de calcul et en place mémoire.

Il serait très intéressant de savoir si les circuits constructifs peuvent se caractériser de façon topologique et d'en dériver de nouveaux algorithmes.

3.2.3 Formats communs de représentation

Participant : Annie Ressouche

Nous avons réalisé la traduction du format SC vers le format DC, tous deux formats internes de représentation des systèmes d'équations booléennes. SC est un format interne au compilateur ESTEREL tandis que DC a été défini comme format commun partagé par plusieurs langages synchrones, dans le cadre du projet Eureka SYNCHRON.

3.3 Scripts et objets réactifs

Participants : Frédéric Boussinot, Karima Machane, Jean-Ferdinand Susini

Mots-clés : réactivité, objets, scripts, parallélisme, événement, World Wide Web, Reactive-C.

Nous avons poursuivi les travaux concernant la programmation par objets réactifs [521], menés dans le cadre d'une action industrielle avec Cnet/France Télécom (cf section 4.2). Une description de ces travaux est disponible sur le Web à l'URL : <http://cma.cma.fr/RC/rc-project.html>. Un livre a été publié sur le sujet [518].

3.3.1 Scripts Réactifs

Participants : Frédéric Boussinot, Jean-Ferdinand Susini

Nous avons défini un langage de *scripts réactifs* [537] qui mélange, dans un cadre interprété, l'approche réactive synchrone du langage SL [520] et les objets réactifs. L'objectif est d'obtenir une grande puissance et une grande souplesse de programmation, bien adaptées aux applications multimédias, en particulier le Web. Un interpréteur de script réactifs, construit au dessus de Tcl/Tk et écrit en Reactive-C, tourne actuellement sur plateformes Unix. Un portage expérimental sous Netscape du noyau de cet interpréteur a été réalisé.

3.3.2 La programmation par ICOBJS

Participant : Frédéric Boussinot

On introduit une programmation simplifiée, complètement graphique, avec un mécanisme puissant de combinaison de comportements : la *programmation par icobjs* [538]. Un icobj combine une partie iconique, une partie comportementale et une animation graphique. Le but est de définir un nouveau style de programmation consistant à construire des sortes de «dessins animés» de comportements. L'aspect animation des icobjs est particulièrement naturel dans le cas de la migration, lorsqu'il s'agit de visualiser le transport d'un agent à travers le réseau. Un système expérimental a été réalisé et est disponible sur le Web. Il est implémenté par des scripts réactifs et a été conçu comme un cadre ludique pour expérimenter avec la notion d'icobj.

3.3.3 Production d'automates à partir de la POR/Rlib

Participant : Karima Machane

Nous continuons l'étude de la production d'automates, en liaison avec le langage SL, à partir d'un noyau de la POR/Rlib de la société Soft Mountain, basé sur Reactive-C et utilisé dans un contexte industriel.

3.4 Vérification et analyse de systèmes communicants

Participants :

Amar Bouali, Michel Bourdellès, Eric Madelaine, Annie Ressouche, Valérie Roy, Robert de Simone

Mots-clés : spécification formelle, vérification de programme, sûreté, concurrence, synchronisation, parallélisme asynchrone, parallélisme synchrone, automate, bisimulation, abstraction, observation, diagramme de décision binaire, protocole de communication, système embarqué, système distribué.

Nos outils de vérification sont basés sur des techniques d'analyse des processus réactifs communicants par leurs modèles, dans le cas d'états finis. La prise en compte exhaustive des configurations possibles réclame l'invention de méthodes et d'algorithmes permettant de combattre la complexité combinatoire inhérente.

Les processus modélisés peuvent être *synchrones* (ESTEREL) ou *asynchrones* (CCS, LOTOS) ; les méthodes de représentation d'états peuvent être *explicites* (énumération) ou *implicites* (« Binary Decision Diagrams »). Nos outils couvrent désormais tous ces cas. L'approche est centrée sur des méthodes algorithmiques de réduction (par abstraction, observation, quotient ...), complétant d'autres approches de type « model-checking » sur des logiques temporelles.

Les développements logiciels sont basés sur des formats communs de représentation de systèmes, permettant la conception et l'ajout rapide de nouveaux modules de vérification. Les manipulations symboliques utilisent la bibliothèque BDD TIGER.

3.4.1 Vérification par représentation implicite et BDD

Participants : Amar Bouali, Robert de Simone

Nous avons implanté ou réimplanté dans le logiciel FC2TOOLS, et à l'aide de la bibliothèque TIGER, des fonctionnalités de détections de blocage, de divergence, d'absence de progression sur les réseaux d'automates, ainsi que des méthodes plus élaborées de minimisation par quotient et de comparaison pour des équivalences de type bisimulation. Ces approches ont également été largement adaptées au cas de processus synchrones, dans un nouveau logiciel XEVE.

Un accent particulier a été mis sur la présentation agréable de diagnostic à l'utilisateur, au moyen de visualisations distribuées sur les réseaux en AUTOGRAPH, ou par le biais du simulateur graphique de programmes ESTEREL.

L'introduction de techniques d'automates observateurs a été mise en œuvre sans problèmes particuliers. Une interface avec le logiciel TEMPEST, développée à l'université d'Austin (Texas), permet d'exprimer de tels observateurs sous forme logique, et de les appliquer dans le cadre de la technologie symbolique BDD.

Ces travaux sont conduits dans le contexte d'une collaboration CIT CNET (voir section 4.1), et leurs extensions (prise en compte partielle des valeurs de données pour la vérification, méthodes incrémentales, réseaux asynchrones de processus synchrones) devraient également se poursuivre dans un nouveau projet Esprit LTR SYRF.

3.4.2 Vérification par représentation explicite d'espaces d'états

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

Les systèmes réactifs synchrones décrivent efficacement le comportement de contrôleurs et se représentent par des machines de Mealy à états finis. Nous avons implanté une minimisation d'automates décrits explicitement de tels contrôleurs, présentés sous la forme de fichiers au format fc2. La relation de bisimulation choisie abstrait la partie réceptions et tests d'absence de signaux du contrôle en l'interprétant symboliquement comme un domaine de combinaisons nécessaire pour un produit

d'émissions de signaux donné. Les fonctions d'équivalence d'automates et de réduction compositionnelle de réseaux d'automates par cette bisimulation sont implantés. Les résultats en terme de simplification de l'expression logique des transitions sont très prometteurs. Ce travail a été conduit dans le contexte d'une collaboration CIT CNET (voir section 4.1).

3.4.3 Application de techniques déductives aux preuves de bisimulation

Participants : Mouloud Bourbel, Eric Madelaine

Une première expérience, dans le cadre du stage de DEA de M.Bourbel, a consisté à implanter sous forme d'un système déductif le mécanisme de preuve de bisimulation formelle du système ECRINS. Un prototype a été réalisé en utilisant Centaur et Coq, et appliqué à la construction de preuves de bisimulation pour de petits exemples. Une telle architecture nous promet une grande souplesse par rapport au langages parallèles étudiés, tant au niveau de leur syntaxe que de la combinaison de systèmes de preuves pour les parties contrôle (bisimulation) et données.

Le prototype réalisé a montré la réalisabilité en Coq de l'algorithme de bisimulation, mais ne couvre pas les aspects de coordination avec un système de preuve sur les données, ni même les aspects de contrôle de la preuve de bisimulation lorsque celle-ci peut ne pas terminer automatiquement. Ceci devrait faire l'objet de développements à venir.

4 Actions industrielles

4.1 CTI CNET «Modélisation de processus mobiles et vérification de processus synchrones»

Participants : Gérard Boudol, Ilaria Castellani, Silvano Dal-Zilio, Davide Sangiorgi, Roberto Amadio, Amar Bouali, Michel Bourdellès, Robert de Simone

Nos travaux dans la partie « vérification » de cette collaboration ont principalement porté sur l'extension de nos méthodes d'analyse par réduction (abstraction, quotient, observation) au cas des processus synchrones. Ceci a donné lieu à des développements logiciels importants, voir sections 3.4.1 et 3.4.2. Un cas d'étude nous a été fourni par le CNET, portant sur l'étude et la détection d'interactions entre services ajoutés en téléphonie. La modélisation « synchrone » du problème en Esterel est encore en cours.

Pour la partie « modélisation de processus mobiles », le travail prévu dans cette collaboration est dans le droit fil des travaux décrit en 3.3, particulièrement dans les aspects typage et vérification.

4.2 CTI CNET « Objets Réactifs »

Participants : Frédéric Boussinot, Karima Machane

Lors de cette CTI, nous avons défini et implémenté plusieurs formalismes : objets réactifs, scripts réactifs, programmation par icobjs et langage Reactive-C++. Ces travaux ont été présentés au colloque de fin de la CTI, à Issy-les-Moulineaux, fin octobre. Plusieurs réunions techniques se sont tenues à Sophia, Lyon ou Issy-les-Moulineaux. Un contrat France Télécom, Soft Mountain et Inria vise à industrialiser ces travaux, en particulier ceux autour de Reactive-C++ et de l'application au Web des scripts réactifs.

4.3 Genie-Dassault

Participants : Gérard Berry, Ellen Sentovich, Delphine Terrasse, Horia Toma

Dans le cadre de cette action de développement, des travaux prospectifs sur la preuve de la traduction de programmes Esterel vers leur forme de circuits digitaux, en utilisant l'assistant de preuve COQ, ont donné des résultats forts encourageant : nous avons en particulier détecté à cette occasion quelques erreurs dans la spécification formelle « papier » de cette traduction. Notre projet bénéficie ici du financement de neuf mois d'ingénieur-expert.

Par ailleurs les collaborations avec la société Dassault sur leur utilisation du langage Esterel se sont encore amplifiées (contrats Armines).

4.4 Motorola

Participants : Gérard Berry, Ellen Sentovich, Robert de Simone

La phase initiale du contrat, d'une durée de six mois, consiste en l'étude de la modélisation par les langages synchrones, et dans notre cas Esterel, d'une spécification simplifiée d'un arbitre de bus matériel. Cette modélisation doit être l'occasion de démontrer les possibilités de l'environnement de développement, dans une perspective d'implantation matérielle.

4.5 Soft Mountain/ France Telecom

Participants : Frédéric Boussinot, Jean-Ferdinand Susini

Cette action de développement, d'une durée de six mois, nous associe à la société Soft Mountain du groupe de technologie de l'Inria, et s'insère dans le cadre plus général de collaborations avec France Telecom. Il s'agit principalement de démontrer l'utilité des notions de scripts réactifs (voir section 3.3.1) dans le cadre d'applications Internet. Notre projet bénéficie ici du financement de six mois d'ingénieur-expert.

4.6 Alcatel-AAR Reutel

Participant : Frédéric Boussinot

Nous participons à cette collaboration, initiée par le projet Ep-Atr de l'Irisa, sur le thème de la définition de comportements réactifs pour des objets distribués avec interface de type CORBA. Nous pourrions prolonger cette activité sur des thèmes de vérification.

4.7 Ventes du compilateur Esterel v5

Les sources du compilateur Esterel v5 ont été vendus à Dassault Aviation et à la compagnie américaine Synopsys. Un contrat Armines de coopération industrielle est en cours avec Dassault Aviation pour le développement de ce compilateur.

5 Actions nationales et internationales

5.1 Actions nationales

Nous participons au groupement C2A, sur le thème des langages synchrones, ainsi qu'au PRC PRS (en sommeil faute de finances).

Nous sommes équipe d'accueil des DEA Informatique et RSD de l'université de Nice, et du DEA MDI de l'université de Marseille.

R. de Simone fait partie d'un groupe de réflexion dans le cadre de l'action interministérielle civil/défense « Maitriser les systèmes complexes réactifs et sûrs ».

5.2 Actions internationales

Le projet a reçu la visite de Matthew Hennesy (université de Sussex) pour deux mois et demi, de Edward Lee (U.C. Berkeley) pour 2 mois, de Michele Boreale (université de Rome) pour quatre mois, Josva Kleist (université d'Aalborg) pour quatre mois, Marco Pistore (université de Pise) pour un mois.

Visites d'une semaine ou moins : M. Fiore (université d'Edimbourg), C. Laneve (Bologne), C. Fournet (Inria Rocquencourt), B. Pierce (Cambridge), X. Esparza (Munich), D. Walker (Warwick), K. Honda et N. Yoshida (Edimbourg), C. Palamidessi (Gènes), A. Rauzy (U. Bordeaux), J-C. Madre (Synopsys).

G. Berry a donné la première Milner Lecture à l'université d'Edimbourg, ainsi qu'un cycle de cours sur Esterel.

6 Diffusion des résultats

6.1 Enseignement

R. Amadio et G. Boudol ont donné un cours (8h + 12h) sur « Lambda-Calcul et Parallélisme » au DEA de mathématiques discrètes et fondements de l'informatique de l'université de la méditerranée. Dans le même DEA I. Castellani a donné un cours sur la « Modélisation et vérification de systèmes distribués » (4h).

R. Amadio a donné un cours sur le π -calcul (6h), dans le cadre d'une école pour jeunes chercheurs organisée par le GDR Programmation. Outre les enseignements liés à sa fonction, il donne un cours de logique dans le DEA Informatique de Marseille.

R. de Simone a donné un cours (24h) commun aux DEA Info et RSD de l'université de Nice. F. Bousinot a participé (6h) à l'enseignement du DEA Info de Charles André.

G. Berry a donné un cours (14h) sur la synthèse et la vérification des systèmes d'états finis à la 9eme Asian School of Computer Science, Rayong, Thaïlande.

V. Roy, A. Bouali, F. Boussinot, M. Robert, G. Berry, X. Fornari enseignent à l'Isia (de l'ordre de 150h globalement), et y encadrent des projets d'étudiants.

Carolina Lavatelli a soutenu le 22 janvier sa thèse « Sémantique du Lambda-Calcul avec Ressources », effectuée sous la direction de G. Boudol, à l'université Paris 7.

6.2 Participation à des conférences et colloques

R. Amadio : exposés de séminaires aux LIM, universités de Marseille et de Belfort, participation aux conférences CONCUR'96 (article [523] présenté par I. Castellani), Formal Techniques in Real Time and Fault Tolerant Systems (présentation de [524]).

G. Berry : conférences invitées au congrès AMAST'96, Munich, et au Symposium SMC/AE on Informatics, Amsterdam. Workshop Logic and Models of Computation à Marseille (exposé invité). Workshop on Formal Verification à Rutgers (exposé invité). Conférence invitée à la journée industrie-recherche organisée par la compagnie Synopsys à Bruxelles. Conférences IDTC'96, (Paris) [535] et ACSC'96 (Singapour).

A. Bouali : conférence AMAST'96 (Munich) (présentation d'outils et démos) [528].

G. Boudol : séminaires Méthodes Formelles pour les Systèmes Critiques à Grenoble, Formal Methods for Open Object-based Distributed Systems à Paris, Semantics of Concurrent Systems à Dagstuhl, Logic

and Models of Computation à Marseille (exposé invité), séminaire de logique de l'université Paris 7 (exposé invité), conférences CAAP-ESOP'96 (article [529] présenté par C. Lavatelli), CONCUR'96.

F. Boussinot : conférence RTCSA'96 à Séoul [530].

I. Castellani : exposés de séminaires à l'Inria Rocquencourt, aux universités de Sussex et Aarhus. Séminaires Semantics of Concurrent Systems à Dagstuhl (présentation de [523]), Partial Order Methods in Verification à Princeton (exposé invité), colloque du projet EXPRESS à Dagstuhl (présentation de [523]), conférence CONCUR'96 (présentation de [523]).

A. Ressouche : conférence CONCUR'96 (Pise) (présentation d'outils et démos).

M. Robert : Congrès AFCET « Modélisation des systèmes réactifs » (Brest) [531].

D. Sangiorgi : séminaires Foundations of Object Oriented Languages (exposé invité [540]), conférences LICS'96 ([532]), CAV'96 ([533]), CONCUR'96 ([523]).

R. de Simone : cours à l'École d'Été MOVEP à Nantes, workshop TACAS (Passau) [526], conférence FLoC/CAV'96 (New Brunswick) [527], réunion Dagstuhl langages synchrones.

H. Toma : conférences ICCAD'96 [534] (San Diego) et FTRTFT'96 [525](Uppsala).

6.3 Organisation de colloques et de cours

Le projet a participé à l'organisation de l'École d'Été MOVEP à Nantes. D. Sangiorgi était membre du comité de programme de CONCUR'96. E. Madelaine était membre du comité de programme de TACAS'97. E. Sentovich était membre du comité de programme de ICCAD'96.

6.4 Diffusion de produits

La version v5 d'ESTEREL a été mise dans le domaine public, en beta-release puis en première version officielle. Des informations sont disponibles à l'URL <http://cma.cma.fr/Esterel/> ou par e-mail à esterel-request@cma.cma.fr.

La version 1 des outils de vérification FC2TOOLS et de l'environnement dédiée aux programmes ESTEREL a également été mise dans le domaine public. Des informations sont disponibles à l'URL <http://cma.cma.fr/Verification/> ou par e-mail à fc2team@cma.cma.fr.

Le langage RC est disponible en domain public à l'URL <http://cma.cma.fr/RC/rc-project.html>, ainsi que divers formalismes associés tels que décrits en section 3.3.

6.5 Autre

Présentation de nos résultats au grand public lors de journées *Sciences en fête* organisées par le ministère de la recherche...

7 Publications

Livres et monographies

[517] G. BERRY, *The constructive semantics of Esterel*, 1996, draft at <http://cma.cma.fr/Esterel>.

[518] F. BOUSSINOT, *La programmation réactive, Collection Technique et Scientifique des Télécommunications*, Masson, 1996.

Articles et chapitres de livre

- [519] A. ARNOLD, I. CASTELLANI, «An algebraic characterization of observational equivalence», *Theoretical Computer Science* 156, 1996, p. 289–299.
- [520] F. BOUSSINOT, R. DE SIMONE, «The SL Synchronous Language», *IEEE Transactions on Software Engineering* 22, 4, avril 1996, p. 256–266.
- [521] F. BOUSSINOT, G. DOUMENC, J.-B. STEFANI, «Reactive Objects», *Annales des Télécommunications* 51 (9-10), 1996, p. 459–473.
- [522] I. CASTELLANI, «Observing distribution in processes: static and dynamic localities», *Int. Journal of Foundations of Computer Science* 6(4), 1995, p. 353–393.

Communications à des congrès, colloques, etc.

- [523] R. AMADIO, I. CASTELLANI, D. SANGIORGI, «On bisimulations for the asynchronous π -calculus», in : *Proceedings CONCUR 96, LNCS*, 1119, p. 147–162, 1996.
- [524] R. AMADIO, M. DAM, «Toward a modal theory of types for the π -calculus», in : *Formal Techniques in Real Time and Fault Tolerant Systems, LNCS*, 1996. Extended version available as SICS-RR 95-18.
- [525] A. BOUALI, J.-P. MARMORAT, R. DE SIMONE, H. TOMA, «Verifying Synchronous Reactive Systems programmed in ESTEREL», in : *Proceedings FTRTFT'96, LNCS*, 1135, p. 463–466, 1996.
- [526] A. BOUALI, A. RESSOUCHE, V. ROY, R. DE SIMONE, «Fc2Tools: a toolset for verification of concurrent systems», in : *Proceedings Workshop TACAS'96*, 1996.
- [527] A. BOUALI, A. RESSOUCHE, V. ROY, R. DE SIMONE, «The Fc2Tools set (tool paper)», in : *Proc. CAV'96, LNCS*, 1102, 1996.
- [528] A. BOUALI, A. RESSOUCHE, V. ROY, R. DE SIMONE, «Presenting Fc2Tools (presentation paper)», in : *Proc. AMAST'96, LNCS*, 1101, 1996.
- [529] G. BOUDOL, C. LAVATELLI, «Full abstraction for a lambda-calculus with resources and convergence testing», in : *Proc. CAAP'96, LNCS*, 1059, 1996.
- [530] F. BOUSSINOT, L. HAZARD, «Reactive Scripts», in : *Proc. RTCSA'96, Seoul*, 1996.
- [531] R. DE SIMONE, M. L. DE SOUZA, «Les automates d'ordre partiel et les équivalences comportementales», in : *Actes du Congrès AFCET sur la Modélisation des Systèmes Réactifs*, p. 105–112, 1996.
- [532] M. FIORE, E. MOGGI, D. SANGIORGI, «A fully abstract model for the π -calculus», in : *LICS'96*, p. 43–54, 1996.
- [533] M. PISTORE, D. SANGIORGI, «A Partition Refinement Algorithm for the π -calculus», in : *Proc. CAV'96, LNCS*, 1102, Springer-Verlag, p. 38–49, 1996.
- [534] E. SENTOVICH, G. BERRY, H. TOMA, «Latch Optimization in Circuits Generated from High-Level Descriptions», in : *Proc. ICCAD'96, San Diego*, 1996.
- [535] T. SHIPLE, G. BERRY, H. TOUATI, «Constructive Analysis of Cyclic Circuits», in : *Proc. IDTC'96, Paris*, 1996.

Rapports de recherche et publications internes

- [536] M. BOREALE, D. SANGIORGI, «Some congruence properties for π -calculus bisimilarities», *rapport de recherche n°2870*, INRIA, 1996.
- [537] F. BOUSSINOT, L. HAZARD, «Reactive Scripts», *Research Report n°RR-2868*, Inria, avril 1996.
- [538] F. BOUSSINOT, «Icobj Programming», *Research Report n°RR-3028*, Inria, 1996.

- [539] B. PIERCE, D. SANGIORGI, «Behavioral equivalence in the π -calculus», *rapport de recherche*, INRIA, 1996.
- [540] D. SANGIORGI, «An interpretation of typed objects into typed π -calculus», *rapport de recherche n°3000*, INRIA, 1996.

8 Abstract

The Meije research team (a.k.a. “Concurrency, Synchronisation and Real-Time” group at Inria), is a joint project with the *Ecole des Mines de Paris* in its *Centre de Mathématiques Appliquées* laboratory at Sophia-Antipolis. Its main research domains are :

- Concurrency Theory, and in particular process algebraic semantics of *higher-order* processes, allowing channel names or even processes to be carried across by communications. Work this year was conducted more specifically on comparative expressivity and relation to verification methods. This work was carried partly inside the european consortium Esprit BRA Confer, and supported by a CTI grant from CNET.
- The synchronous reactive language ESTEREL. The language is designed for the programming of embedded systems, software or hardware controllers typically. It enjoys an industrial-quality programming environment. We have focused recently on precise semantic treatment of causality issues (such as “races” in circuits), with a particular concern for efficient compilation and visualisation of results. Version 5 of the system is now available. This research is a topic of intensive industrial collaborations, primarily with the french aircraft company Dassault but also with several other leading hardware companies such a Motorola and Synopsys.
- The model-based verification of process algebraic networks. In this domain we developed the AUTO/GRAPH automatic verifier, then the FC2TOOLS follow-up including symbolic BDD-based verification modules. Recently we focused on verification of synchronous reactive formalisms. This work was carried partly inside the european consortium Esprit BRA Concur2, and is supported by a CTI grant from CNET and the forthcoming Esprit LTR SYRF project.
- Reactive programming. Here, based on a simple extension of the C language to include basic synchronous reactive paradigms such as *atomic instants* and *signal broadcast*, we developed a range of derived formalisms, like REACTIVE SCRIPTS and REACTIVE OBJECTS. Experiments on using reactive behaviours for graphic programming in the Internet are supported by an *Action de Développement*.

