



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Équipe tick

*Etude et implémentation des systèmes
réactifs synchrones*

Sophia Antipolis

THÈME 1C

*R*apport
d'Activité

2002

Table des matières

1. Composition de l'équipe	1
2. Présentation et objectifs généraux	1
2.1.1. Contexte et Objectifs du projet	1
2.1.2. Axes de recherche	1
2.1.3. Relations internationales et industrielles	2
3. Fondements scientifiques	2
3.1. Formalismes Réactifs Synchrones pour la conception de Systèmes Embarqués	2
4. Domaines d'application	3
4.1. Télécommunications	3
4.2. Systèmes embarqués en avionique/automobile	3
4.3. Synthèse de circuits	4
4.4. Conception conjointe matériel / logiciel	4
5. Logiciels	4
5.1. Environnement Esterel	4
6. Résultats nouveaux	5
6.1. Exécution « efficace » logicielle de programmes Esterel	5
6.2. Traduction d'Esterel vers AAA/SynDEx	5
6.3. Partitionnement structurel pour le calcul symbolique d'états atteignables	5
6.4. Verification par analyse de causalité	6
6.5. Abstraction de registres dans l'analyse de programmes	6
6.6. Analyse syntaxique de programmes Esterel	6
6.7. Traduction vers des formalismes de description de circuits ou d'architectures de systèmes embarqués	6
6.8. Modélisation comportementale de composants d'un framework	7
8. Actions régionales, nationales et internationales	7
8.1. Actions européennes	7
8.1.1. Projet ITEA Prompt2Implementation	7
8.1.2. Projet ITEA EEA-EAST	7
8.1.3. Réseau d'Excellence IST ARTIST	8
9. Diffusion des résultats	8
9.1. Animation de la Communauté scientifique	8
9.2. Enseignement	8
10. Bibliographie	9

1. Composition de l'équipe

Responsable scientifique

Robert de Simone [directeur de recherches, Inria]

Responsable permanent

Annie Ressouche [chargée de recherche]

Assistante de projet

Catherine Juncker [jusqu'au 1^{er} septembre]

Evelyne Largeteau [depuis le 1^{er} septembre]

Chercheurs doctorants

Yannis Brès [allocataire MENESR et moniteur UNSA, jusqu'au 1^{er} octobre]

Fabrice Peix [allocataire MENESR et moniteur UNSA, seconde année de thèse]

Dumitru Potop-Butucaru [boursier Eiffel, jusqu'au 1^{er} décembre]

Olivier Tardieu [Ingénieur du Corps des Mines en détachement, première année de thèse]

Eric Vecchié [boursier région PACA, seconde année de thèse]

Collaborateur extérieur

Charles André [professeur à l'UNSA]

Autre personnel

Laurence Pierre [Maître de Conférences Université de Provence, en délégation jusqu'au 1^{er} octobre]

Stagiaire

Bertrand Blanc [stagiaire DEA RSD UNSA, du 1^{er} avril au 30 septembre]

2. Présentation et objectifs généraux

2.1.1. Contexte et Objectifs du projet

L'action Tick a comme objectif l'étude et l'analyse des systèmes réactifs synchrones, ainsi que leur implantation effective et efficace. Nous nous basons principalement sur le langage Esterel comme formalisme de représentation. Nous poursuivons l'étude de méthodes algorithmiques, basées sur sa sémantique mathématique formelle, ainsi que de logiciels d'analyse et de vérification associés.

La programmation réactive synchrone est particulièrement adaptée aux systèmes temps réel, manipulant des événements logiques avec préemption. Le même formalisme peut modéliser des circuits digitaux, des contrôleurs logiciels, ou des systèmes mixtes incluant logiciel et matériel. Les domaines d'applications principaux sont les systèmes embarqués à forte composante contrôle.

Ces techniques et leur application ont fait l'objet de collaborations suivies avec des sociétés comme Dassault Aviation, Thales, Texas Instruments, Cadence, Intel et France Telecom R&D pour citer les plus importantes. Esterel est actuellement en phase d'industrialisation et de commercialisation par la société Esterel Technologies. Nous poursuivons en parallèle des études sur de nouvelles techniques avancées de compilation et d'analyse, ainsi que des extensions guidées par les classes d'applications émergentes comme les *SoCs* (« System-on-Chip »).

Une version graphique du langage, dénommée SyncCharts, a été conçue par Charles André, membre du projet SPORTS de l'Université de Nice Sophia-Antipolis, et collaborateur extérieur de notre équipe.

2.1.2. Axes de recherche

La compilation actuelle de programmes Esterel utilise une traduction des programmes réactifs en systèmes d'équations booléennes, un formalisme très proche des modèles de circuits digitaux en portes logiques. Ce format intermédiaire est ensuite manipulé par des algorithmes d'analyse, d'optimisation, puis de compilation (logicielle) ou de synthèse (matérielle). Ces algorithmes ont une complexité parfois importante sur de gros programmes, et nous travaillons essentiellement à la conception de nouvelles techniques permettant leur

passage à l'échelle, ainsi que la production de code (logiciel) efficace. Ces améliorations exploitent les informations structurelles syntaxiques des programmes, et donc partent de modèles de descriptions de plus haut niveau que les circuits. Elles procèdent de méthodes d'analyse statique, qui doivent être imaginées dans le contexte spécifique de la programmation synchrone.

Par ailleurs, nous étudions des extensions au langage pour augmenter son expressivité pratique. Nous étudions également des extensions de nos méthodes d'analyse à base de « model-checking » symbolique, en particulier pour le traitement des données par des abstractions appropriées. Enfin, nous étudions l'extension de nos travaux hors du cadre strictement synchrone, en particulier les réseaux asynchrones de processus synchrones. Ces systèmes dits *GALS* (« Localement Asynchrones/Globalement Synchrones) correspondent à des classes de modèles d'importance grandissante, comme les « Systems-on-Chip » rassemblant de nombreux composants sur un même substrat de silicium, et pour lesquels l'hypothèse synchrone n'a de validité qu'au niveau du composant, et plus de l'assemblage.

Nous nous intéressons à la relation entre nos modèles et d'autres formalismes dans lesquels ils peuvent se traduire, en particulier les langages de description de circuits comme VHDL et Verilog, ainsi que les environnements de cosimulation matérielle/logicielle comme VCC ou SynDex, ce dernier développé dans l'Action INRIA OSTRE.

2.1.3. Relations internationales et industrielles

Le compilateur académique Esterel (version binaire) est disponible par ftp avec son environnement de simulation graphique Xes et son logiciel d'analyse et de vérification Xeve à l'adresse <http://www-sop.inria.fr/meije/esterel/esterel-eng.html>.

Nous participons en collaboration avec d'autres équipes INRIA à plusieurs consortiums européens : le projet **ITEA EAST-EEA** sur le logiciel embarqué dans l'automobile (avec de nombreux constructeurs et équipementiers européens, les projets INRIA OSTRE et TRIO et d'autres partenaires académiques) ; le projet ITEA PROMPT2IMPLEMENTATION, sur les méthodologies de conception conjointe (« codesign ») (avec Nokia, Thales, Esterel Technologies et l'Action INRIA OSTRE) ; enfin le Réseau d'Excellence **IST ARTIST** sur le thème des logiciels embarqués (coordonné pour l'INRIA par Albert Benveniste).

Le projet RNRT Syntel s'est conclu cette année.

3. Fondements scientifiques

3.1. Formalismes Réactifs Synchrones pour la conception de Systèmes

Embarqués

Participants : Robert de Simone, Fabrice Peix, Dumitru Potop, Annie Ressouche, Olivier Tardieu, Eric Vecchié.

Cette activité est un thème de collaboration avec l'équipe SPORTS de l'IS3 (CNRS/UNSA), dirigée par Charles André.

Mots clés : Esterel, programmation réactive synchrone, sémantique, causalité, temps réel, compilation, optimisation, circuit digital, conception conjointe.

Le langage Esterel permet la programmation structurée 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 hiérarchisée. La sémantique formelle permet la définition exacte des comportements de programmes, la 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 syntaxiques 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, les parties contrôle de circuits digitaux et, plus généralement, les systèmes réactifs dominés par le contrôle.

Le comportement d'un programme Esterel est défini par une sémantique mathématique formelle [2]. Un programme peut être 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 d'établir une interface avec de nombreux systèmes de vérification formelle comme ceux développés dans le projet.

Le langage Esterel est désormais commercialisé par la société Esterel Technologies, avec laquelle nous entretenons des relations suivies de coopération.

Les recherches nouvelles autour du langage et de son modèle comportemental concernent :

- la caractérisation de méthodes efficaces d'implantation logicielle, pour se limiter à l'exécution de parties réellement actives dans une réaction ;
- l'étude de l'optimisation et de l'analyse des programmes à ce niveau de représentation ;
- l'étude de la projection de ce type de code vers des formalismes comme SynDex ou VHDL comportemental pour des visées de répartition de code ;
- 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 ;
- la redécouverte de structure dans le cadre de la réingénierie de programmes à partir de systèmes d'équations booléennes « plats ».

Ces recherches sont de plus en plus fondées sur la définition d'une approche d'analyse par sémantique statique dédiée à la programmation synchrone.

Les développements théoriques conduisent à des algorithmes implantés dans des prototypes, pouvant aboutir à des extensions 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.

4. Domaines d'application

4.1. Télécommunications

Mots clés : *télécommunications, protocoles, mobilité.*

Le développement de microprocesseurs dédiés aux systèmes mobiles du futur (téléphones cellulaires notamment) passe par la spécification complète de haut niveau de tels systèmes, dans des buts de tests intensifs et de validation avant la mise en production effective des circuits. Nous avons collaboré avec la société Texas Instruments sur ces thèmes. Nous participons actuellement au projet ITEA P2I pour introduire des modélisations « à la UML », mais basées sur des sémantiques réactives synchrones comme le sont les SyncCharts, dans le flot de conception de tels systèmes.

4.2. Systèmes embarqués en avionique/automobile

Mots clés : *systèmes embarqués, transports, protocoles, programmation réactive, contrôleur, avionique, automobile.*

La fiabilité des systèmes embarqués étant souvent critiques, 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. On retrouve ces caractéristiques dans la nouvelle génération de bus synchrones (ou « Time-Triggered ») qui prennent une place prédominante dans les architectures logicielles de ce type d'applications. Nous participons au Réseau d'Excellence IST ARTIST sur ces thématiques.

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 semble bien adapté à la description et à la synthèse efficace de contrôleurs matériels, comme l'ont démontré des collaborations passées avec des sociétés comme Synopsys, Cadence ou Intel. De plus les circuits complexes du futur seront de véritables systèmes sur une seule puce de silicium (SoC, « Systems-on-Chip »), avec des composants IP (« Intellectual Properties ») réutilisables connectés en réseaux partiellement asynchrones. Il serait souhaitable de nouer des collaborations dans ce domaine sur le site de Sophia-Antipolis, où des activités industrielles existent sur ce thème (cf. Club HiTech Electronique).

4.4. Conception conjointe matériel / logiciel

Mots clés : *ingénierie, circuits digitaux.*

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 dans ce domaine avec la société Cadence Design Systems. Cette société développe 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 »). Nous étudions la possibilité d'intégrer ces formalismes dans le produit VCC (Virtual Components Compiler).

5. Logiciels

5.1. Environnement Esterel

Participants : Dumitru Potop, Olivier Tardieu.

Mots clés : *Esterel, compilateur, synthèse, optimisation, causalité.*

Le compilateur Esterel v5 traduit les programmes Esterel vers C ou des formats de description de circuits digitaux. 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 XES permettant de mettre au point les programmes, des optimiseurs spécialisés fondés sur des techniques de calcul booléen, et des interfaces vers le système de vérification automatique de propriétés XEVE.

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-sop.inria.fr/meije/esterel/esterel-eng.html>. Les sociétés Dassault Aviation et Synopsys en ont acquis des licences sources. Propriété conjointe de l'INRIA et d'Armines, il est désormais industrialisé et diffusé par la société Esterel Technologies, 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. Nos travaux actuels portent sur des modules logiciels opérant des pré- ou post-traitements mettant en oeuvre de nouvelles techniques algorithmiques autour du noyau du compilateur.

6. Résultats nouveaux

6.1. Exécution « efficace » logicielle de programmes Esterel

Participants : Dumitru Potop, Robert de Simone.

Mots clés : *Esterel, code efficace.*

La traduction d'Esterel en circuits mène à un schéma simple d'exécution des programmes, qui consiste à évaluer séquentiellement dans un ordre donné toutes les équations définissant les portes logiques et les registres booléens du circuit, et ce lors de chaque réaction du programme. Néanmoins, dans un programme hiérarchique de larges parties peuvent se révéler structurellement inactives, ce qui est difficile à détecter après traduction en circuit. Le sujet de thèse de Dumitru Potop consiste dans l'utilisation de la structure du programme Esterel pour définir un mode de compilation produisant un code plus efficace car n'exécutant que certaines parties actives du système d'équations. Des travaux similaires ont été conduits par Stephen Edwards à l'université de Columbia (New York), ainsi qu'à France Telecom R&D (Grenoble). Le code résultant peut être vu comme une implantation logicielle (séquentielle) efficace, ou comme un code de simulation de haut niveau pour la validation du circuit modélisé.

L'originalité des travaux, qui ont mené à la soutenance de thèse de Dumitru en novembre 2002 [6], est de tenter de définir un modèle formel de représentation intermédiaire qui concilie efficacité du code et sémantique formelle, et sur lequel on puisse conduire des analyses et des optimisations dont la correction soient mathématiquement établie. En particulier nous avons caractérisé à ce niveau une notion de programme acyclique qui correspond exactement à celle existant au niveau des circuits, la seule existant auparavant. D'autres résultats d'optimisation et d'encodage efficace d'états ont été établis. Le format GRC défini par Dumitru est aussi maintenant à la base d'autres travaux portant sur la répartition de ce code efficace modulaire dans des contextes précis.

6.2. Traduction d'Esterel vers AAA/SynDEx

Participants : Fabrice Peix, Robert de Simone.

Le logiciel SYNDEX, développé par l'Action INRIA Ostre à Rocquencourt, permet à l'origine de répartir un programme de type « synchrone flot de données » sur une architecture matérielle donnée, en optimisant la durée totale d'exécution. Esterel introduit un flot de contrôle bien plus sophistiqué, pour lequel souvent une grande partie du programme est inactive dans une réaction donnée, du fait de l'alternance de modes exclusifs dans ce contrôle. On peut donc envisager une « répartition efficace » de programmes Esterel en SYNDEX en suivant les mêmes principes que ceux du paragraphe précédent (où la cible de compilation restait séquentielle).

Nous étudions actuellement, dans le cadre de la thèse de Fabrice Peix, une traduction basé sur le format GRC introduit précédemment, et visant la nouvelle version de SynDEx qui autorise le conditionnement de tâches par des signaux déclencheurs. Le travail se fait en collaboration avec l'équipe Ostre, pour établir les besoins expressifs pour la bonne combinaison des deux systèmes.

6.3. Partitionnement structurel pour le calcul symbolique d'états atteignables

Participants : Robert de Simone, Eric Vecchié.

L'efficacité des méthodes de model-checking symbolique repose grandement sur le partitionnement des fonctions de transition du programme réactif. Dans les circuits ce partitionnement est réalisé au niveau des registres élémentaires, ce qui fait l'efficacité en particulier de la bibliothèque TIGER que nous utilisons dans nos développements. Mais la structure syntaxique des programmes Esterel doit permettre de définir des méthodes bien plus puissantes de partitionnement, en exploitant des relations de séquentialité, d'exclusion ou de concurrence entre des fragments de programmes. Un exemple simple : dans $P ; Q$ il semble intéressant de calculer d'abord les états atteignables de P , en ne s'appuyant que sur la relation de transition de ce mode, avant de calculer similairement les états de Q . En effet le BDD représentant symboliquement **tous** les états de P (ou

Q) atteignables a toutes les chances d'être plus simple que ceux représentant des configurations partielles, qui de surcroît viendraient se mélanger à des configurations partielles de Q dans une approche non structurelle.

Le cas général est plus complexe. Nous étudions dans le cadre de la thèse d'Eric Vecchié des méthodes efficaces de partitionnement structurel respectant la sémantique globale, et autorisant un calcul fragmenté modulaire de l'espace d'états atteignables.

6.4. Verification par analyse de causalité

Participants : Charles André, Robert de Simone.

La vérification de propriétés temporelles des systèmes s'attache traditionnellement aux propriétés de correction entre instants. Du fait de la richesse des comportements lors d'une réaction unique en Esterel, les propriétés « dans l'instant » sont également un sujet d'intérêt pour établir la correction d'un programme. Nous avons étudié cet aspect en nous appuyant sur un exemple de file FIFO qui autorise le dépôt et le retrait simultané de données. Ceci pose des problèmes causaux aux bornes (FIFO pleine ou vide). Une modélisation fine est nécessaire, et pour sa vérification nous introduisons des observateurs dans l'instant. Ces travaux ont fait l'objet d'une publication [7].

6.5. Abstraction de registres dans l'analyse de programmes

Participant : Yannis Bres.

Mots clés : Esterel, code efficace.

La complexité du calcul symbolique de l'espace d'états atteignables est indirectement liée au nombre de registres composant les états locaux de cette structure. On peut simplifier ce calcul au prix d'une approximation conservative en « oubliant » certains registres. Une technique originale et prometteuse d'interprétation abstraite a été définie dans ce sens. Elle est basée sur une interprétation trivaluée de certaines fonctions de transition. Ces résultats ont fait l'objet de la thèse de Yannis Brès, soutenue en décembre 2002 [5].

6.6. Analyse syntaxique de programmes Esterel

Participants : Robert de Simone, Olivier Tardieu.

Les compilateurs existants du langage, visant des représentations à base d'automates ou de circuits synchrones, n'ont que très partiellement recouru à des techniques d'analyse statique rigoureuses au niveau de la structure des programmes eux-mêmes pour optimiser certains aspects de la compilation. C'est la tâche principale qu'a entreprise Olivier Tardieu dans le cadre de sa thèse. Les travaux ont porté d'abord sur le traitement des boucles combinatoires instantanées, puis sur les phénomènes de duplication de code pour éliminer des ambiguïtés comportementales dans les cas d'activation multiple de fragments de programmes dans un même instant. Dans tous les cas, une analyse fine permet de limiter les expansions d'une part, et d'autre part de trouver des critères statiques mais puissants pour relaxer des contraintes syntaxiques.

6.7. Traduction vers des formalismes de description de circuits ou d'architectures de systèmes embarqués

Participants : Bertrand Blanc, Laurence Pierre, Robert de Simone.

Dans le cadre du stage de DEA de Bertrand Blanc nous avons étudié les relations entre Esterel et VHDL au niveau dit « comportemental » (c'est-à-dire algorithmique) de ce dernière. Cette traduction a été réalisée au travers du format GRC introduit précédemment, dans le but d'analyser la capacité des mécanismes de simulations « event-driven » de VHDL à représenter la progression causale du contrôle dans une réaction Esterel, du moins dans le cas des programmes acycliques.

Une différence essentielle entre les deux langages se situe dans le fait que VHDL n'autorise que l'interconnexion *parallèle* de processus ayant un comportement séquentiel, alors qu'Esterel autorise des compositions *séquentielles* aussi bien que *parallèles* à tous niveaux. Dans un second travail Laurence

Pierre a cherché à déterminer des transformations syntaxiques valides pour « aplatir » les programmes Esterel dans des formes ne faisant apparaître que des interconnexions *parallèles* de processus séquentiels (exactement le paradigme de VHDL). Afin de procéder à des expérimentations, nous avons commencé à réaliser l'*automatisation* de cette expansion au moyen d'un outil permettant facilement le codage de règles de réécriture conditionnelles, le système ELAN (<http://www.loria.fr/equipes/protheo/SOFTWARES/ELAN/>) développé au LORIA.

6.8. Modélisation comportementale de composants d'un framework

Participants : Annie Ressousche, Sabine Moisan [Projet ORION], Jean-Paul Rigault [professeur UNSA-I3S].

Dans le projet Orion S. Moisan et JP Rigault ont défini une approche par composants afin de faciliter la construction de moteurs de recherche pour des systèmes à base de connaissances. Nous nous sommes intéressés à la vérification de leur fonctionnement en utilisant un modèle synchrone de description des comportements directement issus des travaux menés sur le sujet dans le projet Meije puis l'Action Tick de l'Inria.

L'année dernière nous avons défini un langage de spécification comportemental pour les moteurs considérés, avec une interprétation opérationnelle dans un modèle mathématique qui permette d'utiliser les méthodes de "model-checking" pour faire des preuves de sûreté de fonctionnement des moteurs. La structuration comportementale suit naturellement la hiérarchie des composants (le comportement d'un sous-type se déduit du comportement du type plus général par une opération de notre langage).

Cette année nous avons complété la sémantique afin de donner un cadre formel à la notion de sous-typage dans l'approche par composants. Cette caractérisation est basée sur une opération de restriction, associée à la notion usuelle de simulation. Avec cette caractérisation, la sémantique mathématique du langage de description est bien fondée et elle nous permet d'utiliser les outils de model checking usuels.

Sur un plan plus pratique, nous sommes en train d'implémenter ce langage de description comportemental, avec une interface en aval vers le model-checker NuSMV. NuSMV permet de modéliser des systèmes synchrones et asynchrones, et il autorise aussi bien des techniques de représentation sous forme de BDD (Binary Decision Diagram) que des méthodes récentes de preuves basée sur les "SAT solvers".

Par ailleurs, une interface graphique pour notre langage est en cours d'étude et de réalisation.

8. Actions régionales, nationales et internationales

8.1. Actions européennes

8.1.1. *Projet ITEA Prompt2Implementation*

Participants : Robert de Simone, Fabrice Peix.

Nous participons à ce projet en collaboration avec l'équipe Ostre de Rocquencourt (qui le gère administrativement pour l'INRIA), ainsi que les sociétés Thales et Nokia, et des universitaires finlandais. L'objectif est d'étudier la modélisation de systèmes embarqués en alliant les formalismes réactifs synchrones et UML, et en considérant les perspectives de répartition de code de type SynDEx. Les sociétés Thales et Nokia doivent prochainement fournir des études de cas pour soutenir l'approche au niveau expérimental.

8.1.2. *Projet ITEA EEA-EAST*

Participants : Charles André, Robert de Simone.

Le but de ce projet, qui rassemble l'essentiel des constructeurs automobiles européens ainsi que de nombreux laboratoires de recherche (en France l'INRIA, le Loria et l'IRCyN), est de promouvoir une approche de conception de logiciels et systèmes informatiques embarqués dans ce domaine en intégrant des formalismes et notations intégrés et, disons, semi-formels. Nous participons ici encore avec l'équipe Ostre, et nos travaux

visent à proposer l'utilisation des SYNCCHARTS, développés par Ch. André dans l'équipe Sports à l'I3s, en combinaison avec les notations UML. Charles André a participé à plusieurs réunions de travail dans ce cadre.

8.1.3. Réseau d'Excellence IST ARTIST

Participants : Robert de Simone, Olivier Tardieu.

Ce réseau tend à préfigurer la forme des prochains Réseaux d'Excellence du 6^e PCRD. Il rassemble donc l'essentiel des acteurs académiques dans le domaine des langages et formalismes de conception pour les systèmes embarqués temps-réel. Un premier travail a consisté à formuler et promouvoir des directions prospectives (de type « roadmaps ») pour des cursus d'enseignement ou des méthodologies industrielles intégrées sur ce thème.

9. Diffusion des résultats

9.1. Animation de la Communauté scientifique

Robert de Simone a été membre élu de la Commission d'Évaluation de l'Institut jusqu'en juin 2002, et a participé à ce titre à plusieurs jurys de recrutement. Il a présidé pour l'année 2002 la Section d'Audition du recrutement chercheur de Sophia-Antipolis. Il est également membre de la Commission de Spécialistes en 27^e section de l'université de Nice/Sophia-Antipolis, et membre du comité de rédaction de la revue TSI. Il a été rapporteur des thèses de D. Gariou (Université de Nantes) et Mirabelle Nebut (Rennes). Enfin, il a organisé le séminaire Synchron'02 portant sur la modélisation réactive synchrone (48 participants, majoritairement français mais également indiens, américains et allemands).

Laurence Pierre a fait partie des comités de programme des conférences FDL'02 (Marseille) et FM-CAD'2002 (Portland). Elle a présenté une contribution à RCS'02.

Annie Ressousche participe comme collaboratrice extérieure au projet « Hybride » du CMA/Ecole des Mines de Paris. Elle a co-encadré avec Valérie Roy (ENSM/CMA) les stages de Sylvain Mairet (DESS UNSA), qui a réalisé un outil de visualisation des programmes Esterel en Java 3D, et de Ya Gao (DEA Informatique UNSA), qui a étudié l'outil Ptolemy II (environnement de modélisation de systèmes hétérogènes).

Olivier Tardieu et Robert de Simone ont participé aux réunions du Réseau d'Excellence IST ARTIST (Paris et Grenoble). Olivier Tardieu a participé également à l'Ecole de Printemps « Sémantique et Programmation » (Agay), et aux conférences DCC'02 (Designing Correct Circuits, Grenoble) et PLDI'02 (Programming Languages Design and Implementation, Berlin). Annie Ressousche et Olivier Tardieu ont participé à la conférence CAV'02 (Computer Aided Verification, Copenhague). Fabrice Peix, Bertrand Blanc, Olivier Tardieu et Robert de Simone ont participé à la conférence FDL'02 (Forum on specification and Design Languages, Marseille).

La plupart des membres du projet ont présenté des contributions au séminaire Synchron'02.

9.2. Enseignement

Robert de Simone coordonne le cours « Méthodes formelles et fiabilité du logiciel » du DEA Informatique de l'université de Nice/Sophia-Antipolis, et il y enseigne (15h) ; il enseigne également à l'ISIA une semaine de cours sur les méthodes formelles et leurs applications (18h).

Laurence Pierre est responsable de la Licence Professionnelle NTI à l'Univ. de Provence (jusqu'en Sept. 2002). Elle enseigne des cours de Parallélisme, Programmation objet et C avancé (niveau second cycle).

Dans le cadre du monitorat, Fabrice Peix enseigne en DEUG Informatique à l'UNSA : TP Unix et systèmes informatiques (en tout 92h annuelles).

Dans le cadre du monitorat, Eric Vecchié enseigne des TP (Systèmes Informatiques, Programmation Applicative (Scheme), ...) en DEUG Informatique à l'UNSA (en tout 92h annuelles).

Annie Ressousche assure à l'ISIA les TP sur les outils d'analyse syntaxique (15h).

10. Bibliographie

Bibliographie de référence

- [1] G. BERRY. *The Foundations of Esterel*. série Foundations of Computing Series, MIT Press, 2000.
- [2] G. BERRY. *The Constructive Semantics of Pure Esterel*. version électronique, 1999.
- [3] G. BERRY. *The Esterel Language Primer*. version électronique, 1999.
- [4] F. BOUSSINOT, R. DE SIMONE. *The Esterel Language*. in « Another Look at Real Time Programming, Proceedings of the IEEE », volume 79, 1991, pages 1293-1304.

Thèses et habilitations à diriger des recherche

- [5] Y. BRÈS. *Exploration implicite et explicite de l'espace d'états atteignables de circuits logiques Esterel*. thèse de doctorat, 2002.
- [6] D. POTOPI. *Optimisations pour la simulation efficace de programmes Esterel*. thèse de doctorat, 2002.

Articles et chapitres de livre

- [7] C. ANDRÉ, R. DE SIMONE. *Synchronous Programming : property within a reaction*. in « Journal Européen des Systèmes automatisés », 2002.
- [8] A. BENVENISTE, P. CASPI, S. EDWARDS, N. HALBWACHS, P. L. GUERNIC, R. DE SIMONE. *Synchronous Languages Twelve Years Later*. in « Proceedings of the IEEE », à paraître.