

*Projet s4**Synthèse et supervision de systèmes,
scénarios**Rennes*

THÈME 1C

*R* *apport
d'Activité*

2002

Table des matières

1. Composition de l'équipe	1
2. Présentation et objectifs généraux	1
2.1. Présentation	1
2.1.1. Un domaine applicatif privilégié : le développement de logiciels temps-réel déployés sur des architectures réparties	2
2.1.2. Adaptation et contrôle des services	2
2.1.3. Déploiement sur une infrastructure donnée	2
2.1.4. Conception par « composants-modèles » décrits selon des formalismes hétérogènes	3
2.1.5. Techniques développées et fondements scientifiques	3
2.1.6. Synthèse de réseaux de Petri	4
2.1.7. Langages de scénarios	4
2.1.8. Modélisation de systèmes répartis faiblement synchrones	4
2.1.9. Classification et résolution de problèmes de contrôle par le μ -calcul quantifié	4
2.1.10. Recherches transversales	4
3. Fondements scientifiques	4
3.1. Fondements scientifiques	4
4. Domaines d'application	5
4.1. Domaines d'application	5
5. Logiciels	6
5.1. Synet : un logiciel de synthèse de réseaux de Petri généraux	6
5.2. Plume : un logiciel d'analyse des performances de HMSC temporisés	6
6. Résultats nouveaux	7
6.1. Réseaux de Petri : synthèse et contrôle	7
6.2. Analyse et synthèse de systèmes réactifs faiblement synchrones	8
6.3. Classification et résolution de problèmes de contrôle par le μ -calcul quantifié	9
6.4. Techniques algébriques linéaires max-plus pour l'analyse des performances temporelles de systèmes concurrents	9
7. Contrats industriels	10
7.1. Ouate : Outils pour l'analyse des performances temporelles de HMSC	10
7.2. Columbus : conception de contrôleurs embarqués pour des systèmes critiques, IST-2001-38314	10
7.3. Réseau d'excellence Artist, IST-2001-34820, 2002-2004.	11
8. Actions régionales, nationales et internationales	11
8.1. Collaboration CNRS-IPI PAN Catalysis	11
8.2. Action coopérative FISC : formalisation et instrumentation des scénarios	12
9. Diffusion des résultats	12
9.1. Édition d'un livre sur le thème de la synthèse et du contrôle de systèmes à événements discrets	12
9.2. Participation à des comités de lecture	12
9.3. Séminaire 68NQRT de l'Irisa	12
9.4. Enseignement universitaire	12
9.5. Éléments divers d'animation scientifique	12
10. Bibliographie	13

1. Composition de l'équipe

S4 est un projet commun à l'INRIA, au CNRS et à l'université de Rennes 1, par l'intermédiaire de l'IRISA (UMR 6074).

Responsable du projet

Benoît Caillaud [CR]

Assistante de projet

Marie-Noëlle Georgeault [TR, à temps partiel dans le projet]

Personnel Inria

Albert Benveniste [DR, à mi-temps dans le projet]

Philippe Darondeau [DR]

Ingénieur expert

Pierre Le Maigat [du 17/02/2002 au 17/08/2002]

Université de Rennes 1

Gilles Lesventes [Maître de conférences]

Sophie Pinchinat [Maître de conférences]

Chercheurs doctorants

Guillaume Feuillade [normalien, à partir du 01/10/2002]

Pierre Le Maigat [allocataire MESR, jusqu'au 16/02/2002]

Stéphane Riedweg [bourse INRIA co-financée par la région Bretagne]

Chercheur post-doctorant

Markus Lohrey [Post doctorant INRIA, jusqu'au 15/04/2002]

Stagiaire

Igor Rosenberg [juillet 2002]

2. Présentation et objectifs généraux

2.1. Présentation

Le projet a pour objectif la réalisation, par des méthodes algorithmiques, de systèmes réactifs et répartis à partir de spécifications partielles ou hétérogènes. Il s'agit de développer un ensemble de techniques, d'algorithmes et d'outils qui permette la synthèse de logiciels réactifs à partir d'une ou de plusieurs descriptions incomplètes spécifiant le comportement attendu du système du point de vue de la fonctionnalité (synchronisation, conflits, communication), du contrôle (sûreté, atteignabilité, vivacité), de l'architecture d'exécution (placement, cloisonnement), ou bien encore des performances quantitatives (temps de réponse, coût de communication, etc).

Ces techniques sont d'abord étudiées sur des modèles de base, comme les automates, les réseaux de Petri, les structures d'événements et leurs extensions temporisées. Les résultats obtenus sont alors adaptés aux modèles plus réalistes mais plus complexes généralement utilisés en télécommunication ou pour les systèmes embarqués temps-réel. En particulier, les vues comportementales de la notation UML (les scénarios et les statecharts), le langage de scénarios HMSC (*High Level Message Sequence Charts*) et le langage synchrone Signal sont à la fois la base des outils développés et le cœur de la stratégie de valorisation des travaux du projet.

L'objectif scientifique du projet peut être caractérisé par les éléments suivants :

Un domaine applicatif privilégié : le développement de logiciels temps-réel, déployés sur des architectures réparties : systèmes de télécommunication, systèmes de contrôle complexes (automobiles, avionique, ...), ateliers flexibles de production, systèmes de travail coopératif, etc.

Un contexte méthodologique : le développement de méthodes et d'outils pouvant assister les concepteurs lors des premières étapes de la conception de logiciels réactifs et répartis. Le point difficile étant d'offrir une conception par « composants-modèles » décrits selon des formalismes hétérogènes, avec un assemblage fiable de ces composants.

Des fondements scientifiques et techniques : les modèles et méthodes formelles associés, prenant en compte les aspects répartis, de concurrence vraie et temporels.

Un effort sur le développement de prototypes et leurs transferts, effort qui est réparti entre des outils sur les langages de scénarios et la synthèse de réseaux de Petri.

Ces divers aspects sont détaillés ci-après.

2.1.1. *Un domaine applicatif privilégié : le développement de logiciels temps-réel déployés sur des architectures réparties*

Spécification des systèmes. Les comportements des systèmes doivent être aisés à adapter, et composables. Les spécifications sont naturellement exprimées sous la forme d'exigences que le système doit être capable de satisfaire. Ces exigences sont de quatre sortes et servent à décrire le comportement attendu du système du point de vue de la fonctionnalité (synchronisation, conflits, communication), du contrôle (sûreté, atteignabilité, vivacité), de l'architecture d'exécution (placement, cloisonnement), ou bien encore des performances quantitatives (temps de réponse, coût de communication, etc.).

Déploiement sur une infrastructure donnée. Une tendance importante de l'industrie est le développement d'un nombre rapidement croissant d'infrastructures logicielles (*middleware*) destinées à faciliter l'intégration des composants et services, ainsi que l'administrabilité des systèmes résultants et leur adaptabilité. Ceci est particulièrement vrai pour les systèmes de télécommunication [29][36][40][28][27], mais tend aussi à se généraliser à d'autres types de systèmes. Le déploiement sur de telles infrastructures des composants et des services spécifiés doit donc pouvoir se faire de manière sûre et efficace.

Parmi les thématiques sous-jacentes, nous avons choisi de nous focaliser sur quelques problèmes plus particuliers que nous détaillons maintenant.

2.1.2. *Adaptation et contrôle des services*

Les équipementiers et opérateurs de télécommunication sont fréquemment confrontés au besoin d'intégrer rapidement de nouvelles fonctionnalités dans leurs mises en œuvres (piles) de protocoles et de services. De l'avis général des intéressés, ceci présente des difficultés sérieuses, et requiert de longues phases de test après modification et intégration, afin de valider le comportement d'ensemble et l'absence d'interactions indésirables entre services. Actuellement, cette intégration de nouveaux services est réalisée au niveau du code exécutable, et non pas de la spécification ou des exigences [35][26].

Ce besoin n'est pas spécifique à l'industrie des télécommunications. Dans des domaines aussi variés que les systèmes de production, les systèmes de contrôles dans les transports ou bien les systèmes de travail coopératif, les ingénieurs procèdent souvent à des adaptations de ces systèmes consistant à en restreindre (on parle alors de contrôle) ou en étendre l'ensemble des comportements possibles.

Le projet S4 contribue au développement de méthodes et d'outils permettant d'assister les concepteurs de ces systèmes dans ces tâches d'adaptation et de contrôle.

2.1.3. *Déploiement sur une infrastructure donnée*

La démarche générale est la suivante. On produit une structure de contrôle et de communication distribuée, dont la validité repose uniquement sur des hypothèses de nature générique portant sur les services de communication et de nommage offerts par le *middleware* considéré. Ceci permet de couvrir de larges familles de *middlewares* supports, plutôt qu'un *middleware* en particulier. Dans l'état actuel de notre savoir-faire, nous ne savons prendre en compte que des aspects purement fonctionnels. Notamment, nous supposons la disponibilité d'une architecture support fiable. Nous étudions actuellement la prise en compte d'aspects

temporels du *middleware* et du service à déployer. Ceci est indispensable quand on veut déployer des services comportant des contraintes temporelles au dessus de couches de communication non fiables mais assurant l'acheminement des messages en temps borné.

2.1.4. Conception par « composants-modèles » décrits selon des formalismes hétérogènes

Les éléments suivants sont au centre de notre démarche :

Hétérogénéité des notations. Nous pensons qu'UML [38] offre une diversité adéquate dans les styles de notations (notations de type « scénarios », notations de type « machines à états », aspects architecturaux). Pour les scénarios on dispose des cas d'utilisation (*use case*), diagrammes de séquence et de collaboration. Pour les machines à états, ce sont les diagrammes d'états, les *statecharts* [32], l'*action semantics* [24], et les évolutions futures concernant la convergence de SDL [34] vers UML. Les aspects architecturaux sont les plus populaires dans UML : diagrammes de classes, d'objets, de déploiement.

Notre ambition est de contribuer à l'unification des sémantiques des vues comportementales d'UML, et d'en permettre, par là-même, la combinaison.

Des exigences aux multiples facettes. Les exigences sur le système devant être réalisé ou modifié sont de plusieurs natures :

- fonctionnalité : synchronisation, conflits et communication ;
- contrôle : sûreté, atteignabilité et vivacité ;
- Architecture d'exécution : localisation et cloisonnement ;
- performances quantitatives : temps de réponse et coût de communication - nous laissons de côté les aspects, importants en ce qui concerne les logiciels embarqués, liés à la consommation de ressources : mémoire, énergie, etc.

Ces différents aspects sont exprimables en UML ou dans ses extensions envisagées (UML-RT [31]).

Le problème est de pouvoir analyser et transformer ces exigences au niveau système.

C'est l'un de nos objectifs

Ces techniques de modélisation et d'analyse étant disponibles, il convient de procéder à la *synthèse* vers un modèle homogène et exécutable, satisfaisant aux exigences ci-dessus.

2.1.5. Techniques développées et fondements scientifiques

Le projet s'attache à développer un ensemble de techniques, d'algorithmes et d'outils qui permette la synthèse de logiciels réactifs à partir d'une ou de plusieurs descriptions incomplètes spécifiant le comportement attendu du système (fonctionnalité, contrôle, architecture d'exécution, performances quantitatives). Ceci implique, d'une part, de savoir transformer ces descriptions hétérogènes en descriptions homogènes de plus bas niveau : automates communicants, réseaux de Petri, etc. D'autre part il est aussi nécessaire de pouvoir valider les descriptions hétérogènes : vérification de propriétés.

La démarche scientifique du projet s'appuie sur une approche rigoureuse de ces questions et la constitution de fondements théoriques solides. Cela permet de prouver la correction des transformations proposées (fonctionnalités et contrôle) ; mais aussi l'optimalité des performances quantitatives des réalisations produites par nos méthodes. Les techniques de synthèse et de vérification envisagées sont d'abord étudiées sur des modèles de base (automates, réseaux de Petri, structures d'événements, systèmes de transitions synchrones). Les résultats obtenus sont alors adaptés aux modèles plus réalistes mais plus complexes généralement utilisés pour la conception des systèmes de télécommunication, de production, de travail coopératif ou les systèmes embarqués.

Le programme scientifique du projet est composé de quatre axes principaux de recherches, et plusieurs thèmes transversaux.

2.1.6. Synthèse de réseaux de Petri

Il s'agit de poursuivre l'activité principale du projet Paragraphe au cours de ces dernières années. En plus du développement de cette théorie, les applications visées sont la réalisation répartie de spécifications incomplètes (scénarios, etc) et le contrôle réparti de systèmes à événements discrets.

2.1.7. Langages de scénarios

HMSC [33] et nouveaux formalismes à base de structures d'événements infinies régulières. Les aspects développés seront la caractérisation de classes décidables de scénarios, la réalisation de scénarios en familles de processus communicants et l'utilisation de nouveaux formalismes comportementaux pour l'expression des exigences.

2.1.8. Modélisation de systèmes répartis faiblement synchrones

Le troisième axe porte sur le développement des fondements théoriques et des aspects expérimentaux d'un formalisme synchrone pour la modélisation de systèmes réactifs temps-réel répartis.

2.1.9. Classification et résolution de problèmes de contrôle par le μ -calcul quantifié

Le mu-calcul quantifié permet d'exprimer en toute généralité de très nombreux problèmes de contrôle ainsi que l'existence de solutions optimales à ces derniers. Cette logique permet donc d'établir une classification des problèmes de contrôle. Cette logique permet également la résolution de ces problèmes, par l'utilisation d'algorithmes de calcul de stratégies gagnantes dans les jeux à parité. Cet axe de recherche contribue donc à classifier et à résoudre une classe la plus large possible de problèmes de contrôle.

2.1.10. Recherches transversales

De ces quatre axes de recherche, nous dégagons deux sujets de recherches transversaux : la modélisation par structures d'événements régulières et l'analyse quantitative de systèmes à l'aide d'algèbres tropicales.

3. Fondements scientifiques

3.1. Fondements scientifiques

L'assise du projet est l'ensemble des modèles et théories algébriques, combinatoires ou logiques des systèmes de transitions, séquentiels ou concurrents, synchrones ou asynchrones, étudiés et développés dans les trois dernières décennies. Ces modèles visent en général à décrire le fonctionnement de systèmes finis ou infinis dans lesquels il est fait abstraction des valeurs des variables soumises à une évaluation fonctionnelle, pour se concentrer sur les flux de contrôle qui régissent ou commandent leurs changements d'états. Les systèmes décrits peuvent être autonomes ou réactifs, c'est-à-dire être ou non immergés dans un environnement dont ils dépendent par des flux entrants ou sortants. Les descriptions peuvent être explicites (c'est le cas lorsque on considère des automates donnés en extension par un ensemble d'états et un ensemble de transitions entre états) ou implicites (automates déterminés par des règles de transitions symboliques, c'est-à-dire paramétrées par des variables d'états ou de contrôle, produits d'automates partiellement synchronisés, réseaux de Petri, systèmes d'équations orientées contraignant les réactions instantanées d'un système synchrone à ses flux entrants, ...). Les descriptions peuvent être non ambiguës, c'est-à-dire qu'elles déterminent entièrement le système décrit (cas des exemples précédents), ou bien être ambiguës, c'est-à-dire qu'à une description donnée correspond un ensemble éventuellement vide de systèmes (c'est en particulier les cas des descriptions données par des formules de fragments logiques comme le μ -calcul ou la logique du second ordre monadique sur les arbres ou sur les graphes, mais c'est aussi le cas des descriptions dans lesquelles des exigences sur les comportements des systèmes sont formulées à l'aide de scénarios ou de langages de scénarios, par exemple des MSC ou des HMSC). Selon le cas, on s'attachera à décrire la structure d'un système ou bien à l'oublier pour spécifier le comportement du système. La plupart des modèles considérés sont suffisamment généraux pour permettre l'un et l'autre, le passage de l'un à l'autre s'effectuant par une transformation interne appelée dépliage (ceci vaut pour les automates, pour les produits d'automates, et pour les réseaux de Petri). D'autres modèles s'attachent exclusivement à la description des comportements, c'est-à-dire des systèmes dépliés (le meilleur exemple en est donné par les structures d'événements). Un autre critère de classification des

modèles est la forme de concurrence qu'ils permettent de décrire. On parlera de systèmes séquentiels dans le cas d'automates classiques, ou bien de systèmes synchrones lorsque les équations d'évolution servent à synchroniser plusieurs flux entrants ou sortants, ou bien encore de systèmes asynchrones s'agissant de produits d'automates partiellement synchronisés. Ce clivage est assez superficiel, sachant que la compilation d'un système synchrone, comme l'évaluation d'un produit d'automates, aboutissent l'une et l'autre à l'obtention d'un automate classique, à moins qu'on n'assujettisse ces transformations à une contrainte supplémentaire concernant la ventilation des événements ou des transitions sur les sites d'une architecture distribuée. La cible de ces transformations, au lieu d'un automate, est alors une collection finie d'automates communiquant par émission-réception asynchrone de messages (modèle des *message passing automata*). Les contraintes de distribution ne sont pas sans entraîner un surcroît de complexité, sachant qu'on demande souvent à l'implémentation distribuée de respecter l'ordre partiel entre les événements du système à réaliser, considéré par certains comme d'importance primordiale dans la sémantique des automates concurrents (entre autres les divers automates reconnaissant les langages de traces sur des alphabets munis d'une relation de commutation entre lettres) ou des réseaux de Petri.

Pour plus d'information sur les différents modèles, on pourra entre autres consulter les ouvrages suivants :

1. Jan van Leeuwen (ed.), *Handbook of Theoretical Computer Science - Volume B : Formal Models and Semantics*, Elsevier, 1990.
2. Wolfgang Reisig and Grzegorz Rozenberg (eds.), *Lectures on Petri nets : advances in Petri nets*, Lecture Notes in Computer Science, Vol. 1491, 1492, Springer, 1998.
3. Volker Diekert and Grzegorz Rozenberg (eds.), *The Book of Traces*, World Scientific, 1995.
4. André Arnold and Damian Niwinski, *Rudiments of μ -Calculus*, North-Holland, 2001.
5. Gérard Berry, *Synchronous languages for hardware and software reactive systems Hardware Description Languages and their Applications*, Chapman and Hall, 1997.

De façon générale, nous nous appuyons sur les résultats de décision ou d'indécision afférents à ces modèles (décision de l'inclusion des langages réguliers, de la bisimilarité entre les automates, de l'accessibilité dans les réseaux de Petri, de la validité des formules du μ -calcul, etc.), et sur les théorèmes de représentation effectifs qui permettent de traduire tout ou partie d'un modèle dans un autre (théorème de Zielonka pour passer d'un langage de traces reconnaissable à un produit d'automates partiellement synchronisé, synthèse d'un réseau de Petri à partir d'un automate, d'un langage de mots ou d'un langage de scénarios, construction d'un jeu de parité à partir d'une formule du μ -calcul, etc.). Nous contribuons par nos travaux à enrichir l'éventail de ces résultats de représentation effectifs, en étudiant en particulier leur contenu algorithmique dans le cas de la synthèse de réseaux. Nos travaux s'inscrivent de fait pour l'essentiel dans la problématique de la synthèse, qui consiste à fournir partout où cela est possible des outils d'assistance automatisée à la conception des systèmes logiciels, avec un soin particulier pour les problèmes de réalisation distribuée. Ceci s'applique à nos travaux sur les langages de scénarios, l'objectif étant de synthétiser la meilleure réalisation distribuée de ces langages les approximant par le haut. Ceci s'applique à nos travaux sur la synthèse de contrôleurs. Dans ce cadre dû à M. Wonham, le problème consiste à calculer un système (le contrôleur) tel que son produit partiellement synchronisé avec un système donné (à contrôler) se comporte conformément à un objectif donné (confinement dans un langage régulier, ou bisimilarité avec un automate donné, ou bien satisfaction d'une formule du μ -calcul, etc...). Le problème de l'adaptation de services que l'on rencontre dans l'industrie du logiciel est dans une large mesure réductible au problème précédent, le contrôleur définissant indirectement les modifications à apporter au logiciel existant.

4. Domaines d'application

4.1. Domaines d'application

Les applications des travaux du projet se situent principalement dans le domaine de la conception du logiciel réparti pour des systèmes de télécommunication et des systèmes embarqués temps-réel. Nos travaux s'appliquent également à la conception de systèmes de production automatisés et de travail coopératif.

Les travaux de recherche du projet S4 s'appliquent à la conception de systèmes temps-réel comportant du logiciel déployé sur des architectures réparties. Deux domaines d'applications sont privilégiés : les systèmes de télécommunications et les systèmes embarqués (avionique, automobile, ...).

Les travaux sur l'analyse des performances de systèmes concurrents à l'aide de techniques algébriques max-plus (paragraphe 6.4) et l'outil Plume (paragraphe 5.2) permettent de prédire les performances temporelles d'un système ou d'un service de télécommunication décrit à l'aide de scénarios HMSC (*High Level Message Sequence Charts*). Ce travail est réalisé en collaboration avec France Télécom Recherche et Développement (paragraphe 7.1).

Les travaux sur l'analyse et la synthèse de systèmes réactifs faiblement synchrones (paragraphe 6.2) contribuent à l'amélioration du processus de développement (de la spécification des exigences à la mise en œuvre) du logiciel réparti constitutif de systèmes de contrôle complexes utilisés en avionique et en automobile. Il s'agit de proposer des méthodes et des outils de conception de logiciel devant être déployé sur des architectures dites « mollement activées par le temps » (*Loosely Time-Triggered Architectures*, LTTA), utilisées en particulier dans les avions Airbus, et en fait assez largement répandues sous diverses variantes. Dans ces architectures, la communication se fait par des écritures et lectures non bloquantes et non synchronisées. En d'autres termes les messages peuvent être sporadiquement perdus ou dupliqués. Sur ce sujet, le projet IST Columbus (paragraphe 7.2) vise à proposer un cadre théorique et méthodologique permettant de lisser le processus de développement de ces logiciels et de proposer des passerelles sémantiques permettant l'intégration et l'inter-connexion des différents outils mis en œuvre au cours de ce processus.

Par ailleurs, certains résultats ont été appliqués à des domaines aussi variés que les systèmes de production automatisés et les systèmes de travail coopératif. En particulier les techniques de synthèse de réseaux de Petri généraux développées par Philippe Darondeau ont été utilisées dans le projet MACSI de l'INRIA Lorraine pour résoudre des problèmes de contrôle d'ateliers de production flexibles. De plus, l'outil Synet (paragraphe 5.1) est utilisé à l'université technique de Berlin dans le cadre de travaux de recherche sur l'optimisation et le contrôle de systèmes de travail coopératif.

5. Logiciels

5.1. Synet : un logiciel de synthèse de réseaux de Petri généraux

Participants : Benoît Caillaud, Igor Rosenberg.

Mots clés : réseaux de Petri, synthèse, automates, semi-linéarité, HMSC, scénarios, régions, algèbre linéaire, polyèdres.

Le logiciel SYNET est un outil polyvalent de synthèse de réseaux de Petri généraux. Il sert de plate-forme de démonstration des algorithmes de synthèse de réseaux développés dans le projet S4. L'outil commence à être diffusé de façon restreinte en dehors de l'IRISA : il est utilisé à l'université technique de Berlin dans le cadre de travaux de recherche sur l'optimisation et le contrôle de systèmes de travail coopératif et également à l'université de Toronto sur des questions de synthèse de contrôle de systèmes concurrents.

En 2002 l'outil a évolué par l'adjonction d'un algorithme de synthèse de réseaux à partir d'un langage (travail de stage de Igor Rosenberg). Étant donné un langage régulier (défini par un automate ou une expression régulière), l'outil peut maintenant calculer un réseaux (éventuellement non borné) dont le langage est le plus petit langage de réseau contenant le langage donné en entrée. Une deuxième évolution de l'outil est en cours (mini-projet de 3e année du DIIC, le diplôme d'ingénieur en informatique et communication de l'IFSIC) par l'adjonction de l'algorithme de synthèse de réseaux à partir d'un HMSC [5]. Il s'agit à nouveau de calculer la meilleure approximation (dans la classe des langages de réseaux) d'une spécification incomplète d'un système. Le réseau produit pouvant servir de base à la réalisation du système décrit par des HMSC.

5.2. Plume : un logiciel d'analyse des performances de HMSC temporisés

Participants : Benoît Caillaud, Pierre Le Maigat.

Mots clés : *max-plus, SCILAB, langages de scénarios, HMSC, analyse de performances, régime stationnaire, contrôle.*

Le logiciel Plume [22] permet d'analyser et de contrôler des automates d'ordres temporisés, issus de scénarios HMSC (High Level Message Sequence Charts). Il met en œuvre les techniques de calcul max-plus de débits moyens et de périodes pour des comportements ultimentents périodiques en régime stationnaire, développés dans la thèse de Pierre Le Maigat [10]. Ce logiciel a été en partie développé dans le cadre de la collaboration Ouate avec France Télécom Recherche et Développement, Lannion (cf. paragraphe 7.1). Il est composé de deux modules : 1/ un ensemble de fonctions d'analyse d'automates d'ordres bipartis, développé par Pierre Le Maigat ; et 2/ un utilitaire permettant d'importer des HMSC (initialement développé par Loïc Hérouët et mis à jour par Benoît Caillaud). Ce logiciel a été déposé à l'Agence de protection des programmes.

6. Résultats nouveaux

6.1. Réseaux de Petri : synthèse et contrôle

Participants : Benoît Caillaud, Philippe Darondeau.

Mots clés : *réseau de Petri, graphes marqués, synthèse, région, graphe automatique, superviseur.*

Glossaire

graphe marqué Un graphe marqué est un réseau de Petri ordinaire dans lequel chaque place a exactement une transition d'entrée et une transition de sortie.

synthèse Le problème de synthèse des réseaux de Petri consiste, étant donné un système de transitions initialisé, à construire un réseau de Petri dont le graphe des marquages accessibles soit isomorphe à ce système de transitions.

région Les régions d'un système de transitions sont les morphismes qui l'envoient dans le graphe de Cayley des entiers, restreint aux nœuds positifs ; elles sont interprétées comme des places des réseaux synthétisés.

graphe automatique Graphe accessible dont l'ensemble des nœuds est représenté par un sous ensemble rationnel des chemins incidents et par une relation rationnelle définissant leur confluence, et dont l'ensemble des arcs étiquetés est représenté par autant de relations rationnelles entre les nœuds que d'étiquettes

superviseur Un superviseur est un système de contrôle qui permet ou interdit les transitions contrôlables du système contrôlé en fonction de ses transitions observables.

Un article faisant le point de nos acquis sur la synthèse des réseaux de Petri à partir de systèmes de transitions finis et sur ses applications à la distribution a été publié [12]. Un article montrant que les correspondances de Galois entre réseaux de Petri et automates peuvent être étendues à des automates généralisés a été publié [11]. Un article montrant une anomalie des réseaux élémentaires vis-à-vis des autres classes de réseaux a été publié [13]. Un article proposant, pour des graphes marqués vivants ayant des actions incontrôlables ou inobservables, un calcul du superviseur optimal imposant une contrainte linéaire sur les vecteurs de tir, a été accepté pour publication [15]. Les informations sur l'ensemble de ces sujets figurent dans le rapport 2001 du projet Paragraphe. Un nouveau résultat acquis cette année dans un travail mené en collaboration avec E. Badouel est la décision constructive du problème de la synthèse des réseaux de Petri à partir des graphes automatiques. Divers travaux non encore aboutis ont par ailleurs été dédiés à l'étude combinatoire des systèmes de transitions asynchrones et des graphes marqués bornés, en vue de leur transformation en des réseaux élémentaires équivalents à l'étiquetage des transitions près. Ces problèmes, et plus généralement tous les problèmes concernant la décomposition des systèmes en composantes séquentielles, sont critiques dans la perspective de répartition automatique des systèmes et de leurs contrôleurs sur des architectures distribuées. Les résultats seront décrits dans un rapport ultérieur. Dans les dernières années, nous avons successivement

traité la synthèse des réseaux de Petri généraux à partir des automates finis ou des langages rationnels, à partir des langages algébriques déterministes, c'est-à-dire reconnus par des automates à pile déterministes, et à partir des graphes de transitions de ces automates à pile. Ce dernier travail a permis de montrer que les techniques de parallélisation et de distribution inhérentes à la synthèse de réseaux s'appliquent aussi bien à des systèmes infinis, mais il est regrettable que les graphes pris en compte ne permettent pas de représenter le comportement d'un système aussi simple que le produit asynchrone de deux compteurs entiers totalement indépendants. Notre effort a donc porté sur l'adaptation de notre algorithme de synthèse à une classe plus large de graphes infinis, contenant au moins la grille sur le quart de plan positif. La classe considérée est directement inspirée des graphes à structure automatique définis et étudiés par Géraud Senizergues [37], eux mêmes dérivés des groupes automatiques d'Epstein *et al.* [30], à ceci près que nous relâchons toutes les contraintes d'asynchronisme borné posées sur les automates à deux bandes qui définissent la structure automatique (c'est-à-dire l'équivalence sur les chemins qui représentent les nœuds du graphe et les relations de transition). Il ressort de notre étude que notre méthode de synthèse s'applique avec des simplifications notables dans ce cadre beaucoup plus général. Nous avons aussi constaté que cette méthode s'applique toujours lorsque l'on part de spécifications automatiques de graphes au lieu de graphes automatiques, c'est-à-dire lorsque la spécification donnée ne détermine plus un graphe mais un ensemble de graphes ayant en commun des propriétés de structure (techniquement, on n'exige plus que deux chemins confluent si et seulement si ils sont équivalents au sens de la présentation automatique, ni qu'une transition soit possible si et seulement si elle est définie par l'une des relations de la présentation automatique). Cette remarque ouvre la voie à la synthèse de réseaux à partir de descriptions ambiguës, mais il reste du chemin à parcourir afin de relier spécifications automatiques de graphes et autres modèles moins hermétiques (par exemple les HMSC). Un document présentant nos résultats sera prochainement disponible.

6.2. Analyse et synthèse de systèmes réactifs faiblement synchrones

Participants : Albert Benveniste, Benoît Caillaud.

Mots clés : *synchrone, réseaux de Kahn, architectures réparties, endochronie, isochronie, architectures faiblement synchrones.*

Dans la suite des projets européens SACRES et SAFEAIR (ex-projet EP-ATR), A. Benveniste et B. Caillaud ont poursuivi leur réflexion sur le déploiement de spécifications synchrones sur des architectures qui ne satisfont pas au modèle synchrone. Les premiers résultats fondamentaux avaient été obtenus en 2000, dans l'article de référence [25]. Dans cet article, deux théorèmes de réduction étaient formulés, qui permettent de caractériser de manière effective les programmes synchrones qui se déploient sans perte de sémantique sur des architectures de type « réseaux de Kahn ». Ces résultats complétaient de manière importante des résultats antérieurs dûs à P. Caspi, qui portaient sur les programmes synchrones *fonctionnels*. Les résultats de [25] étaient valides y compris pour des spécifications incomplètes ou des abstractions d'architectures. Les deux notions centrales d'*endochronie* (originellement due à P. Le Guernic en 1990, mais non formalisée mathématiquement jusqu'alors) et d'*isochronie* (due à B. Caillaud).

Les architectures de type « réseaux de Kahn » sont des média de communication sans perte, qui se comportent comme un réseau point-à-point de FIFO non bornées. Donc : écriture non bloquante, mais lecture bloquante. Or, les architectures utilisées en contrôle distribué de systèmes critiques n'acceptent pas les protocoles de communication bloquants, pour des raisons évidentes de tolérance aux fautes. Dans le cadre du projet européen CRISYS avec Airbus, P. Caspi (de Verimag), avait attiré notre attention sur des architectures dites « mollement activées par le temps » (*Loosely Time-Triggered Architectures*, LTTA), utilisées en particulier dans les avions Airbus, et en fait assez largement répandues sous diverses variantes. Dans ces architectures, écriture et lecture sont non bloquantes, et la mise à jour des variables est en outre désynchronisée des accès en lecture ou écriture : on peut perdre des messages ou en dupliquer, mais « pas trop ». Dans l'article [18], en collaboration avec le groupe de Verimag, nous montrons comment un petit protocole logiciel simple déployé au-dessus de LTTA permet d'émuler un « réseau de Kahn », et donc

d'effectuer un déploiement correct. Ce protocole a été prouvé à la main, et (presque) automatiquement avec les prouveurs LESAR (pour Lustre) et SIGALI (pour Signal).

Dans une conférence plénière [17] à Euro-Par'2002, A. Benveniste a présenté un tutoriel sur cette thématique.

Enfin, il convient de signaler l'article [14], sur l'ensemble des langages synchrones, à paraître dans les Proceedings of the IEEE, numéro spécial sur les systèmes embarqués. Cet article fait un point intéressant sur la discipline et ses progrès, douze ans après la parution du numéro spécial qui leur fut consacré en 1991 dans cette même revue.

6.3. Classification et résolution de problèmes de contrôle par le μ -calcul quantifié

Participants : Sophie Pinchinat, Stéphane Riedweg.

Mots clés : *contrôle, systèmes à événements discrets, observation partielle, systèmes communicants, logique, mu-calcul, automates d'arbres, stratégies, jeux de parités.*

Nous abordons les problèmes de synthèse de contrôleur par le biais de la logique.

Nous avons développé une théorie dans laquelle les énoncés des problèmes de contrôle se ramènent à l'évaluation de formules logiques sur le système à contrôler, c'est-à-dire au *model-checking* de cette logique.

La théorie proposée s'avère extrêmement bien adaptée aux problèmes de contrôle et extrêmement expressive : elle offre un cadre très général dans lequel les paramètres comme le type de systèmes (fermés vs. ouverts), le type de propriétés, le type d'interaction avec le système, la possibilité d'énoncer des critères d'optimalité du contrôle (équité, permissivité maximum), etc. sont abordés dans le même cadre. A notre connaissance, aucune des approches proposées jusqu'ici ne permet de capturer d'un seul coup un aussi grand nombre de concepts.

La logique considérée est une logique temporelle de points fixes, le μ -calcul, enrichie par des quantifications monadiques. Bien que cette extension augmente en général le pouvoir d'expressivité du μ -calcul, on montre que dans le cadre des modèles d'arbres de comportements, elle admet, tout comme son fragment du μ -calcul, une sémantique basée sur les automates d'arbres. Cette sémantique est alors utilisable pour ramener les problèmes de décision de *model-checking* et de satisfaisabilité à des calculs de stratégies gagnantes dans les jeux de parité. Sur le plan pratique, ces calculs étant constructifs, nous obtenons un modèle de contrôleur.

L'étude des fondements théoriques du μ -calcul quantifié nous montre que cette logique est non élémentaire autant pour le problème du *model-checking* que pour celui de la satisfaisabilité. Toutefois, la complexité de programme (i.e. le *model-checking* lorsque la formule est fixée est polynomiale. En fait, les fragments successifs de profondeur d'alternation de quantifications ont des complexités croissantes dans l'échelle des exponentielles. Toutefois, comme les problèmes de contrôle s'énoncent dans le fragment d'alternation deux, on obtient une complexité élémentaire. Comme de plus, les problèmes de contrôle sont exprimables en pratique dans une famille fixée et finie de formules du μ -calcul, on tombe à nouveau, comme expliqué plus haut, dans la classe polynomiale. Une partie de ces premiers travaux ont été soumis à une conférence d'informatique théorique.

Notre approche offre d'excellentes perspectives de travail. Dans un futur proche, nous établirons la preuve de la compatibilité de notre approche avec celle de l'équipe de Bordeaux. Dans ces techniques, il est par exemple possible de forcer l'appartenance des contrôleurs à celle des contrôleurs agissant sous l'observation partielle du système. À moyen terme, le μ -calcul quantifié et d'éventuelles extensions pourront être étudiés pour les problèmes de la synthèse de composants communicants.

6.4. Techniques algébriques linéaires max-plus pour l'analyse des performances temporelles de systèmes concurrents

Participants : Pierre Le Maigat, Albert Benveniste.

Mots clés : *max-plus, scénario, HMSC, réseau de Petri.*

Dans sa thèse de doctorat [10], Pierre Le Maigat a développé et appliqué des techniques algébriques max-plus linéaires pour l'analyse des performances temporelles de systèmes concurrents décrits par des scénarios HMSC (*High Level Message Sequence Charts* [33]) ou des réseaux de Petri. L'algèbre max-plus permet de représenter naturellement et d'analyser les contraintes de synchronisation et de causalité présentes dans les systèmes concurrents temps-réels.

Les langages de scénarios permettent la spécification partielle d'applications réparties complexes, en particulier dans le domaine des télécommunications. Il est important de pouvoir vérifier les exigences de performances dès les premières étapes de la conception d'un système, alors que le concepteur ne dispose que de scénarios décrivant les comportements du système envisagé. Afin d'obtenir des estimations de ces performances, nous appliquons l'algèbre max-plus à une définition temporisée d'un langage de scénarios : les HMSC. Ceci nous permet de calculer différentes notions de trafics pour des régimes stationnaires des systèmes. Dans le cas d'une accumulation de messages dans un canal de communication, Pierre Le Maigat a montré comment estimer la vitesse d'accumulation et comment cette accumulation peut être éliminée par un contrôle optimal sur les instances. Les algorithmes d'analyse et de synthèse de contrôle issus du travail de doctorat de Pierre Le Maigat ont été mis en œuvre dans l'outil Plume, dans le cadre de la collaboration Ouate avec France Télécom Recherche et Développement, Lannion.

Ces mêmes techniques d'analyse s'appliquent aux réseaux de Petri temporisés bornés vérifiant certaines hypothèses de linéarité. Le cas plus général des réseaux de Petri temporisés déterministes a demandé l'adaptation de la notion de polynômes max-plus. Il a ainsi été obtenu une dualité algébrique entre compteurs et dateurs, ce qui a permis le développement en séries des équations polynomiales. Ce formalisme permet l'analyse et le contrôle des performances de certains réseaux.

7. Contrats industriels

7.1. Ouate : Outils pour l'analyse des performances temporelles de HMSC

Participants : Benoît Caillaud, Pierre Le Maigat.

La collaboration avec France Télécom Recherche et Développement, Lannion (Ouate, contrat 101C04550031334061), porte sur le développement de techniques et d'outils d'analyse de scénarios HMSC (*High Level Message Sequence Charts*) temporisés. Cette collaboration d'une durée de vingt-quatre mois a débuté en septembre 2001. Pierre Le Maigat a été recruté comme ingénieur expert pour une durée de six mois et a finalisé le développement de Plume, un prototype d'outil d'analyse des performances de HMSC temporisés, utilisant certains résultats de son travail de doctorat [10]. Cet outil, décrit en partie 5.2, a été déposé à l'Agence de protection des programmes et a été livré pour évaluation à France Télécom. La suite de la collaboration porte maintenant sur le développement de démonstrateurs mettant en valeur l'outil. En particulier, nous étudions le protocole de diffusion fiable RMTP2 [39]. Ce protocole est intéressant pour plusieurs raisons. Tout d'abord, pour prendre en compte sa nature dynamique (ajout et retrait de nœuds dans l'arborescence d'agrégation des acquittements), il nous faut proposer une composition de HMSC permettant de décrire le comportement d'un ensemble de processus par projections de ses comportements sur une partie de son architecture. Ensuite, il nous faudra interpréter les nombreuses temporisations (*timers*) avant de pouvoir évaluer quelques aspects de performance du protocole RMTP-II avec l'outil Plume. Ce travail se fera en collaboration avec Loïc Hérouët du projet Triskell.

7.2. Columbus : conception de contrôleurs embarqués pour des systèmes critiques, IST-2001-38314

Participants : Albert Benveniste, Benoît Caillaud.

Le projet Columbus est un contrat du 5e PCRD ayant démarré en 2002, pour une durée de deux ans. L'objectif de ce projet est d'étudier un processus de conception pour les systèmes embarqués critiques, qui soit intégré,

semi-automatique et transparent, formellement vérifiable et mathématiquement correct, depuis la spécification jusqu'à la mise en œuvre. Le projet étant de taille modeste et essentiellement universitaire, l'effort porte sur les fondements nécessaires à une intégration de qualité entre les divers outils de la chaîne, depuis les outils d'ingénierie scientifique (Matlab/Simulink/Stateflow), jusqu'aux plates-formes de co-design.

L'originalité de Columbus est d'être un projet à forte coopération UE-USA, la composante américaine étant constituée de partenaires du projet DARPA Mobies, sur les systèmes embarqués. Les partenaires de Columbus sont : l'Université de Cambridge (J. Lygeros, systèmes hybrides), l'Université de l'Aquila à Rome, et plus particulièrement son centre d'excellence DEWS sur les contrôle et les systèmes embarqués (M.-D. Di Benedetto, avec A. Sangiovanni-Vincentelli comme associé), l'INRIA (projet S4, A. Benveniste et B. Caillaud, avec la participation de P. Le Guernic, du projet Espresso), l'Université de Berkeley (S. Sastry, systèmes hybrides et systèmes embarqués, également projet Mobies), et le groupe ISIS de l'Université de Vanderbilt (J. Sztipanovits, plate-forme UML, également projet Mobies).

L'INRIA est responsable du lot « Théorie des Meta-modèles », c'est-à-dire les fondements de l'intégration sémantique des divers outils. Les travaux débutent, ils reposeront sur l'expérience acquise autour des langages synchrones et de Signal, et en particulier sur les travaux autour de la distribution de programmes synchrones réalisés au sein des projets européens successifs Sacres et Safeair.

7.3. Réseau d'excellence Artist, IST-2001-34820, 2002-2004.

Participants : Albert Benveniste, Benoît Caillaud.

Le projet S4 est engagé de manière importante dans le réseau d'excellence Artist, animé par Joseph Sifakis, de Verimag. Artist est un réseau d'excellence du 5e PCRD, dont la thématique centrale porte sur les systèmes embarqués. Les missions principales d'Artist consistent en une activité de type « roadmap », et une réflexion sur les cursus universitaires pour le secteur des systèmes embarqués. Artist mène donc essentiellement une activité de prospective, qui repose à la fois sur des enquêtes auprès du secteur industriel, et une réflexion scientifique propre à ses membres. Enfin, Artist a l'ambition de devenir un réseau d'excellence du 6e PCRD.

Artist comporte actuellement trois actions : l'action 1, dite « Hard Real Time », l'action 2 « Components », et l'action 3 « RTOS and QoS ». Albert Benveniste est coordinateur de l'action 1, et Benoît Caillaud est membre des actions 1 et 2. L'action 2 est animée par Bengt Jonsson, de l'Université d'Uppsala, et l'action 3 est animée par Giorgio Butazzo, de l'Université de Pavie. L'action 1 a choisi de concentrer sa réflexion sur « Semantic Platform », c'est-à-dire les fondements nécessaires à une intégration des divers outils qui interviennent tout au long du processus de conception.

8. Actions régionales, nationales et internationales

8.1. Collaboration CNRS-IPI PAN Catalysis

La collaboration Catalysis se fait dans le cadre d'une collaboration entre le CNRS et l'académie des sciences de Pologne. Elle permet le séjour au sein du projet S4 de chercheurs de l'IPI PAN (Institut de recherche en informatique) à Gdansk. Réciproquement, elle permet des séjours de chercheurs du projet S4 à l'IPI PAN. Les participants à cette collaboration sont : Benoît Caillaud, Philippe Darondeau, Sophie Pinchinat (projet S4) et Marek Bednarczyk, Andrzej Borzyszkowski, Wieslaw Pawlowski et Stefan Sokolowski (IPI PAN, Gdansk).

Benoît Caillaud et Sophie Pinchinat ont séjourné deux semaines à Gdansk en juillet 2002. Ce séjour a permis d'initier avec Marek Bednarczyk un travail sur l'utilisation de produits fibrés dans la catégorie des systèmes de transitions asynchrones par pas (*step transition systems*) pour modéliser la synchronisation et le contrôle de processus concurrents. Ce produit fibré généralise la plupart des produits de synchronisation connus. Ce travail a été poursuivi en novembre 2002, lors du séjour à l'IRISA de Marek Bednarczyk et Andrzej Borzyszkowski. Un article a été soumis à une conférence internationale sur la théorie des réseaux de Petri.

8.2. Action coopérative FISC : formalisation et instrumentation des scénarios

L'ARC FISC (<http://www.irisa.fr/pampa/arc-fisc/>) est une action de recherche coopérative de l'INRIA lancée en janvier 2001 et d'une durée de deux ans. Elle regroupe un ensemble d'équipes autour d'un thème commun : les langages de scénarios pour les télécommunications et l'automatique. Quinze chercheurs répartis dans quatre laboratoires participent à l'action. Les laboratoires impliqués sont : l'IRISA (Rennes), le LIAFA (université Paris 7), l'INRIA Rocquencourt et l'INRIA Rhône-Alpes (Grenoble). Benoît Caillaud en est le coordinateur.

9. Diffusion des résultats

9.1. Édition d'un livre sur le thème de la synthèse et du contrôle de systèmes à événements discrets

Benoît Caillaud, Philippe Darondeau, Luciano Lavagno et Xiaolan Xie sont éditeurs d'un livre sur la synthèse et le contrôle de systèmes à événements discrets [9]. Ce livre a été édité à partir de présentations à deux *workshops* organisés pendant l'été 2001 : le workshop SCODES'2001 (*Symposium on control of discrete event systems*, organisé par Benoît Caillaud et Xiaolan Xie) et le *Workshop on synthesis of concurrent systems* (satellite de la conférence ICATPN'01, Newcastle upon Tyne, Royaume-Uni, organisé par Philippe Darondeau et Luciano Lavagno).

9.2. Participation à des comités de lecture

Albert Benveniste est membre du comité éditorial du journal *Proceedings of the IEEE*. Il est éditeur associé du journal *Discrete Event Dynamic Systems : Theory and Applications*. Il est membre du comité de programme de la conférence internationale TACAS'2003. Benoît Caillaud est membre du comité de programme du workshop national AFADL'2003. Philippe Darondeau a été membre du comité de programme de la conférence internationale ICATPN'2002.

9.3. Séminaire 68NQRT de l'Irisa

Sophie Pinchinat est responsable de la programmation du séminaire hebdomadaire **68NQRT** de l'Irisa (<http://www.irisa.fr/NQRT/index.html>). Ce séminaire propose des exposés scientifiques principalement sur les thèmes : génie logiciel, informatique théorique, mathématiques discrètes, intelligence artificielle. Pour l'année 2002, on a compté 31 exposés dont 24 donnés par des invités extérieurs.

9.4. Enseignement universitaire

Benoît Caillaud et Sophie Pinchinat enseignent le module « Analyse comportementale des systèmes réactifs et répartis » (A2R) de la filière « Génie logiciel et méthodes formelles » du DEA d'informatique de l'université de Rennes 1. Cet enseignement est depuis cette année en deux parties : les années paires, Benoît Caillaud enseigne les techniques d'analyse, de synthèse et de contrôle de systèmes modélisés par des réseaux de Petri. Les années impaires, Sophie Pinchinat enseigne les techniques de vérification sur des systèmes concurrents de propriétés exprimées dans des logiques modales.

9.5. Éléments divers d'animation scientifique

Albert Benveniste a donné une conférence invitée plénière à Euro-Par'2002. Philippe Darondeau est membre du groupe de travail IFIP WG2.2. Gilles Lesventes est vice-président de la commission de spécialistes de l'IFSIC, université de Rennes 1.

10. Bibliographie

Bibliographie de référence

- [1] E. BADOUEL, P. DARONDEAU. *Theory of regions*. in « Lectures on Petri Nets I : Basic Models », série Lecture Notes in Computer Science, volume 1491, Springer, 1999, pages 529-586.
- [2] A. BENVENISTE. *Some Synchronization Issues When Designing Embedded Systems from Components*. in « Embedded Software, First International Workshop, EMSOFT 2001 », série Lecture Notes in Computer Science, volume 2211, Springer, éditeurs T. HENZINGER, C. M. KIRSCH., pages 32-49, Tahoe City, CA, USA, October, 2001.
- [3] A. BENVENISTE, B. CAILLAUD, P. LE GUERNIC. *Compositionality in dataflow synchronous languages : specification and distributed code generation*. in « Information and Computation », volume 163, 2000, pages 125-171.
- [4] A. BENVENISTE, C. JARD, S. GAUBERT. *Algebraic techniques for timed systems*. in « CONCUR'98, Concurrency Theory, 9th International Conference », série Lecture Notes in Computer Science, volume 1466, Springer, éditeurs D. SANGIORGI, R. DE SIMONE., pages 373-388, Nice, France, September, 1998.
- [5] B. CAILLAUD, P. DARONDEAU, L. HÉLOUËT, G. LESVENTES. *HMSCs as specifications... with PN as completions*. éditeurs F. CASSEZ, C. JARD, B. ROZOY, M. DERMOT., in « Modeling and Verification of Parallel Processes », série Lecture Notes in Computer Science, volume 2067, Springer, 2001, pages 125-152, <http://link.springer.de/link/service/series/0558/bibs/2067/20670125.htm>.
- [6] P. DARONDEAU. *On the Petri net realization of context-free graphs*. in « Theoretical Computer Science », numéro 1-2, volume 258, 2001, pages 573-598.
- [7] P. LE MAIGAT, L. HÉLOUËT. *A (max,+) approach for time in message sequence charts*. in « Proceedings of the 5th Workshop on Discrete Event Systems », Kluwer Academic Publishers, éditeurs R. BOEL, G. STREMERSCHE., pages 83-92, Ghent, Belgium, 2000.
- [8] H. MARCHAND, S. PINCHINAT. *Supervisory Control Problem using Symbolic Bisimulation Techniques*. in « 2000 American Control Conference », pages 4067-4071, Chicago, Illinois, USA, June, 2000.

Livres et monographies

- [9] B. CAILLAUD, P. DARONDEAU, L. LAVAGNO, X. XIE (EDS.). *Synthesis and Control of Discrete Event Systems*. Kluwer Academic Press, 2002, <http://www.kap.nl/prod/b/0-7923-7639-0>.

Thèses et habilitations à diriger des recherche

- [10] P. LE MAIGAT. *Techniques algébriques max-plus pour l'analyse des performances temporelles de systèmes concurrents*. thèse de doctorat, université de Rennes 1, école doctorale Matisse, septembre, 2002, <ftp://ftp.irisa.fr/techreports/theses/2002/lemaigat.pdf>.

Articles et chapitres de livre

- [11] E. BADOUEL, M. BEDNARCZYK, P. DARONDEAU. *Generalized Automata and their Net Representations*. éditeurs H. EHRIG, G. JUHÁS, J. PADBERG, G. ROZENBERG., in « Unifying Petri Nets », série LNCS Advances in Petri Nets, volume 2128, Springer Verlag, 2001, pages 304-345, <http://link.springer.de/link/service/series/0558/bibs/2128/21280304.htm>.
- [12] E. BADOUEL, B. CAILLAUD, P. DARONDEAU. *Distributing Finite Automata through Petri Net Synthesis*. in « Journal on Formal Aspects of Computing », volume 13, 2002, pages 447-470.
- [13] M. BEDNARCZYK, P. DARONDEAU. *Looking for diamonds*. éditeurs B. CAILLAUD, P. DARONDEAU, L. LAVAGNO, X. XIE., in « Synthesis and Control of Discrete Event Systems », Kluwer Academic Publishers, 2002, pages 214-219, <http://www.kap.nl/prod/b/0-7923-7639-0>.
- [14] A. BENVENISTE, P. CASPI, S. EDWARDS, N. HALBWACHS, P. LE GUERNIC, R. DE SIMONE. *The Synchronous Languages Twelve Years Later*. in « Proceedings of the IEEE », 2002, à paraître.
- [15] P. DARONDEAU, X. XIE. *Linear Control of Live Marked Graphs*. in « Automatica », 2002, à paraître.
- [16] L. HÉLOUËT, C. JARD, B. CAILLAUD. *An Event Structure Semantics for Message Sequence Chart*. in « Mathematical Structures in Computer Science », volume 12, 2002, pages 377-403.

Communications à des congrès, colloques, etc.

- [17] A. BENVENISTE. *Non-massive, Non-high Performance, Distributed Computing*. in « Euro-Par 2002, Parallel Processing », série Lecture Notes in Computer Science, volume 2400, Springer, éditeurs B. MONIEN, R. FELDMANN., pages 32-49, Paderborn, Germany, August, 2002, Invited talk.
- [18] A. BENVENISTE, P. CASPI, P. LE GUERNIC, H. MARCHAND, J-P. TALPIN, S. TRIPAKIS. *A Protocol for Loosely Time-Triggered Architectures*. in « Embedded Software, Second International Workshop, EMSOFT 2002 », série Lecture Notes in Computer Science, volume 2491, Springer, éditeurs A. SANGIOVANNI-VINCENTELLI, J. SIFAKIS., pages 252-265, Grenoble, October, 2002.
- [19] P. CASPI, A. BENVENISTE. *Toward an Approximation Theory for Computerised Control*. in « Embedded Software, Second International Workshop, EMSOFT 2002 », série Lecture Notes in Computer Science, volume 2491, Springer, éditeurs A. SANGIOVANNI-VINCENTELLI, J. SIFAKIS., pages 294-304, Grenoble, October, 2002.
- [20] M. LOHREY. *Safe realizability of high-level message sequence charts*. in « Concurrency Theory, 13th International Conference CONCUR'02 », série Lecture Notes in Computer Science, volume 2421, Springer, pages 177-192, 2002.
- [21] M. LOHREY, A. MUSCHOLL. *Bounded MSC Communication*. in « Foundations of Software Science and Computation Structures, 5th International Conference, FOSSACS 2002 », série Lecture Notes in Computer Science, volume 2303, Springer, éditeurs M. NIELSEN, U. ENGBERG., pages 295-309, Grenoble, France, April, 2002, <http://link.springer.de/link/service/series/0558/bibs/2303/23030295.htm>.

Divers

- [22] P. LE MAIGAT. *PLUME, un outil pour l'analyse de performance de HMSC temporisés*. juillet, 2002, Livraison dans le cadre du contrat France Télécom N° 011B846.
- [23] I. ROSENBERG. *Synthèse et supervision de systèmes avec Synet*. septembre, 2002, Rapport de stage de première année, Magistère « modélisation mathématique et méthodes informatiques », université de Rennes 1 et ENS Cachan.

Bibliographie générale

- [24] *Response to OMG RFP ad/98-11-01, Action Semantics for the UML*. Object Management Group, February, 2001, <http://www.omg.org/>.
- [25] A. BENVENISTE, B. CAILLAUD, P. LE GUERNIC. *Compositionality in dataflow synchronous languages : specification and distributed code generation*. in « Information and Computation », volume 163, 2000, pages 125-171.
- [26] E. CAMERON. *A Feature Interaction Benchmark for IN and Beyond*. 1994.
- [27] *CORBA components*. Object Management Group, March, 1999, <http://www.omg.org/>.
- [28] *The common object request broker : architecture and specification*. Object Management Group, March, 1999, <http://www.omg.org/>.
- [29] *Proceedings of Middleware'98, IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing*. éditeurs N. DAVIES, K. RAYMOND, J. SEITZ., Springer, Lake District National Park, UK, September, 1998.
- [30] D. EPSTEIN, J. CANNON, D. HOLT, S. LEVY, M. PATERSON, W. THURSTON. *Word Processing in Groups*. Jones and Bartlett Publishers, 1992.
- [31] R. GROSU, M. BROU, B. SELIC, G. STEFANESCU. *Towards a Calculus for UML-RT Specifications*. in « Seventh OOPSLA Workshop on Behavioral Semantics of OO Business and System Specifications », Vancouver, Canada, October, 1998, <http://www4.informatik.tu-muenchen.de/papers/grosuBrouSelicStefanescuOOPSLA98.html>.
- [32] D. HAREL. *Statecharts : A visual formulation for complex systems*. in « Science of Computer Programming », numéro 3, volume 8, 1987, pages 231-274.
- [33] *ITU-TS Recommendation Z.120 : Message Sequence Chart (MSC)*. International Telecommunication Union, Geneva, September, 1993, <http://www.itu.int/home/index.html>.
- [34] *Recommendation Z.100 - Specification and description language (SDL)*. International Telecommunication Union, November, 1999, <http://www.itu.int/home/index.html>.

- [35] *FIW'98 Feature Interactions in Telecommunications and Software Systems*, V. éditeurs K. KIMBLER, W. BOUMA., IOS Press, Lund, Sweden, September, 1998.
- [36] *Middleware 2000 · IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing*. éditeurs J. SVENTEK, G. COULSON., série LNCS, volume 1795, Springer, New York, NY, USA, April, 2000.
- [37] G. SÉNIZERGUES. *Definability in weak monadic second order logic of some infinite graphs*. Dagstuhl seminar on Automata theory : Infinite computations, Wadern **28** (1992) 16, 1992.
- [38] *OMG Unified Modeling Language Specification*. <http://cgi.omg.org/cgi-bin/doc?ad/99-06-09>, June, 1999, Version 1.3.
- [39] B. WHETTEN, M. BASAVIAH, S. PAUL, T. MONTGOMERY, N. RASTOGI, J. CONLAN, T. YEH. *RMTP-II Specification*. IETF Internet Draft, April, 1998, <http://www.globecom.net/ietf/draft/draft-whetten-rmtp-ii-00.html>.
- [40] *ODP Reference model : foundations and architecture*. ITU-T Recommendations X.902 and X.903, ISO/IEC 10746, International Telecommunication Union, November, 1995, <http://www.itu.int/home/index.html>.