École Normale Supérieure de Cachan

Laboratoire Spécification et Vérification

UMR 8643

Bilan scientifique

pour le
contrat quadriennal 2010-2013

Octobre 2008

Sommaire

Rapport scientifique

Présentation générale

Ce rapport fait le bilan scientifique du Laboratoire Spécification et Vérification sur la période allant de janvier 2004 au printemps 2008. Cette période couvre donc un peu plus de quatre ans. Le choix d'inclure l'année 2004 est motivé par le souci de prolonger directement notre précédent rapport quadriennal et d'éviter ainsi de laisser l'année 2004 hors bilan. Certaines informations essentielles mais mouvantes dans le temps, comme la composition des équipes et l'organigramme du laboratoire, ont été arrêtées au premier octobre 2008, soit au plus près possible de la date de remise de ce rapport.

Le LSV a été créé il y a exactement 12 ans. À son origine, 8 chercheurs et enseignants-chercheurs se sont réunis autour d'une même problématique, **la vérification des systèmes informatiques critiques**, qu'ils savaient riche d'avenir et pour laquelle les demandes issues aussi bien du monde industriel que de la société dans son ensemble n'ont fait qu'augmenter. Sous la direction de Michel Bidoit, le laboratoire s'est alors investi dans ce projet collectif. Soutenu par l'ENS Cachan et le CNRS, par l'INRIA depuis 2002, stimulé par le Département d'Informatique de l'ENS Cachan, le Master Parisien de Recherche en Informatique et l'École Doctorale Sciences Pratiques, le LSV est devenu un centre reconnu d'excellence en recherche sur la vérification.

Ces quatre dernières années, nous avons prolongé et amplifié cette dynamique sous les aspects suivants :

Croissance et renouvellement : Le laboratoire est passé de 16 à 22 chercheurs et enseignants-chercheurs permanents et a accueilli une nouvelle équipe, l'axe DAHU dirigé par Luc Segoufin. Sur les 22 permanents actuels, 11 ont été recrutés sur les 4 dernières années. Notre recrutement s'est fortement internationalisé : 3 récents chargés de recherche et 1 maître de conférences sont issus d'universités étrangères. Les départs sont des maîtres de conférences promus professeurs (B. Bérard à Paris 6, F. Laroussinie et R. Treinen à Paris 7) ou des prises de fonctions importantes (A. Petit et M. Bidoit nommés directeurs des centres de recherche INRIA Paris–Rocquencourt et Saclay–Île-de-France, respectivement).

Élargissement thématique : Le laboratoire s'est attaqué à des problèmes nouveaux : la vérification de programmes manipulant des pointeurs et la vérification appliquée aux systèmes à base de données. Il s'agit de **directions nouvelles**, au-delà des évolutions naturelles qu'ont vécues les quatre axes, qui viennent élargir notre spectre de compétences dans le domaine de la vérification.

Rayonnement scientifique : Sur ses thèmes, le rayonnement du laboratoire s'est affirmé avec la présence de nos chercheurs dans de très nombreux comités de programmes (*cf.* page 12) ; l'influence marquante de certains de nos articles, par exemple Comon & Jurski 1998, Gastin & Oddoux 2001, Finkel & Schnoebelen 2001, Comon & Shmatikov 2003, chacun cité plus de

100 fois selon *Google Scholar*[1] ; l'essaimage de nos anciens doctorants dans les principaux laboratoires et universités françaises[2].

Formation des doctorants : Même si, d'un point de vue purement quantitatif, le ratio « nb. doctorants / nb. chercheurs permanents » est relativement faible au LSV, le laboratoire s'est beaucoup investi dans l'encadrement de ses doctorants, lesquels bénéficient souvent d'une cotutelle. Le résultat est un taux exceptionnel d'intégration dans les corps de chargés de recherche ou de maîtres de conférences.

Nouveaux partenariats : Le LSV s'est investi dans des projets de recherche impliquant de nouveaux réseaux de collaborations dans lesquels l'ENS Cachan joue un rôle déterminant : le RTRA Digiteo et le pôle de compétitivité Systematic Paris-Région. De façon plus structurante encore, la création au sein de l'ENS Cachan de l'Institut Farman fin 2006 incite et facilite le lancement de nouveaux projets de recherche interdisciplinaire visant à la maîtrise de systèmes complexes. En parallèle, notre coopération avec l'INRIA Saclay s'est approfondie avec l'accueil d'un second projet INRIA.

Distinctions : Trois de nos chercheurs ont été récemment distingués de façon exceptionnelle. Ainsi Patricia Bouyer a bénéficié d'une bourse Marie Curie et a reçu la médaille de bronze du CNRS en 2007 ; en 2008 François Laroussinie[3] a été nommé à l'Institut Universitaire de France, et Hubert Comon-Lundh va recevoir en novembre prochain la médaille d'argent du CNRS. Ces distinctions sont la partie la plus visible de la dynamique positive que nous évoquons plus haut.

La suite de cette présentation générale va décrire le mode de fonctionnement du laboratoire (organisation) et donner quelques éléments synthétiques de notre bilan (auto-évaluation) avant de dresser un bilan plus exhaustif, axe par axe, de notre activité scientifique des quatre dernières années. (Du fait de sa création récente, l'axe DAHU ne figure pas dans ce bilan.)

Organisation du LSV

Le LSV est organisé en quatre axes, ou équipes, correspondant à des domaines applicatifs ou des spécialités scientifiques particulières. Les effectifs au premier octobre 2008 sont résumés dans le tableau ci-dessous.

	TEMPO	INFINI	SECSI	DAHU	Total
Enseignants-chercheurs	4	3	2	1	**10**
Chercheurs des EPST	5	1	3	3	**12**
Chercheurs accueillis temporairement	1	3	0	0	**4**
Doctorants[4]	6	4	6	2	**18**
	16	11	11	6	**44**

L'axe TEMPO, animé par Patricia Bouyer et Nicolas Markey, est spécialisé sur les modèles et les logiques temporisés et concurrents ainsi que sur les applications de type systèmes embarqués où l'évaluation quantitative joue un rôle important. L'axe INFINI, animé par Alain Finkel, est spécialisé sur les modèles complexes et hétérogènes présents dans un large spectre d'applications

[1] Ces références sont issues de la précédente période quadriennale. Les articles publiés en 2004–2008 sont trop récents pour déjà atteindre la barre de la centaine de citations mais ils sont plusieurs à s'en approcher.

[2] *cf.* `www.lsv.ens-cachan.fr/People/anciens-doctorants.php` pour une liste complète

[3] François Laroussinie a quitté le LSV et l'ENS Cachan en septembre 2007 pour devenir professeur à l'Université Paris Diderot et rejoindre le LIAFA.

[4] Par souci de réalisme, ce décompte inclut les doctorants inscrits au premier octobre 2008, mais omet ceux qui s'apprêtent à soutenir leur thèse en décembre 2008 au plus tard.

allant des protocoles distribués ou embarqués aux systèmes logiciels. L'axe SECSI, animé par Jean Goubault-Larrecq et Hubert Comon-Lundh, se consacre aux applications relevant de la sécurité des systèmes d'information. L'axe DAHU, animé par Luc Segoufin, s'intéresse lui à la vérification appliquée aux systèmes basés sur les données.

Cette structuration du laboratoire en équipes distinctes a été introduite en 2003 et vise à mieux organiser notre vie scientifique, en particulier en identifiant des responsables scientifiques à un niveau intermédiaire entre le directeur d'unité et le porteur de projet ou le directeur de thèse.

Elle ne doit pas être interprétée comme une partition en domaines scientifiques disjoints : les centres d'intérêt communs et les interactions sont très nombreux et contribuent à la dynamique du laboratoire ; le séminaire du laboratoire est commun à toutes les équipes ; les groupes de travail propres à chaque équipe sont annoncés à tout le laboratoire et ouverts à tous.

Elle ne se traduit pas davantage en cloisons administratives : les équipes n'ont pas de budget propre (sur l'utilisation des crédits, *cf.* page 15), la répartition des locaux, des tâches collectives, des allocations doctorales, *etc.*, ne tient pas compte des frontières d'équipes. Enfin, il n'existe pas de « *mailing-list* » permettant de ne s'adresser qu'à une équipe donnée.

La vie du laboratoire est organisée à partir d'un *conseil de laboratoire* qui réunit toutes les deux à trois semaines la totalité des permanents ainsi qu'un représentant des doctorants. Cette organisation, qui reste encore viable avec notre taille actuelle, permet d'associer tous les chercheurs et enseignants-chercheurs aux décisions qui engagent le laboratoire.

Enfin, une fois par an, au printemps, le laboratoire réunit tous ses membres pour un séminaire de deux jours hors de Cachan. C'est l'occasion pour chacun de découvrir les thèmes émergents dans les diverses équipes et, pour le laboratoire, de conduire une réflexion stratégique sur sa politique scientifique.

Auto-évaluation

Avant de lister ce qui nous semble être les points forts et les points faibles du LSV, nous revenons brièvement sur les objectifs qui avaient été affichés il y a quatre ans.

Réalisation des objectifs de l'unité

Les objectifs que nous nous étions fixés il y a quatre ans sont en grande partie réalisés.

Développement des recherches pluridisciplinaires. Ce développement a pu s'appuyer sur les opportunités offertes par l'Institut Farman. Le LSV est aujourd'hui impliqué dans 3 projets où il collabore avec des équipes des laboratoires LURPA (production automatisée) et SATIE (technologies de l'information et de l'énergie) de l'ENS Cachan.

Vérification des systèmes embarqués. Les recherches des axes TEMPO et INFINI ont évolué vers la prise en compte de modèles plus complexes, combinant des aspects distribués, quantitatifs et stochastiques, adaptés à la modélisation et à l'analyse des systèmes embarqués. Aujourd'hui le LSV participe au réseau d'excellence européen ARTIST2 « *Embedded Systems Design* » et au pôle de compétitivité Systematic Paris-Région et notre expertise est reconnue par exemple sur les modèles temporisés à coûts et sur la vérification des systèmes probabilistes.

Analyse statique et vérification des logiciels. Sur ce thème, notre objectif a été en partie atteint avec le développement de nouvelles techniques au sein de l'axe INFINI pour la prise en compte de la gestion de la mémoire dynamique et des objets structurés dans

les programmes. Toutefois nous ne sommes pas encore parvenus à recruter un spécialiste d'analyse statique comme nous cherchons à le faire depuis plusieurs années.

Renforcement des liens avec l'INRIA. Nous avions formé le projet de susciter la création d'un second projet de recherche reconnu par l'INRIA et hébergé à Cachan. Cet objectif a été atteint avec l'arrivée de Luc Segoufin et la création au sein du LSV d'une nouvelle équipe-projet INRIA axée sur les problématiques de vérification dans les grandes bases de données et les systèmes centrés sur les données.

Notons que notre projet initial reposait sur le potentiel scientifique et humain présent au sein de l'axe TEMPO. Ce potentiel reste éminemment exploitable et nous continuerons à explorer les voies d'un développement de la participation de l'INRIA aux recherches du LSV.

Points forts de l'unité

Nous mentionnons les points forts qui nous semblent les plus importants, et qui s'appliquent au laboratoire dans son intégralité, par opposition aux points forts de tel ou tel chercheur ou équipe.

Formation des chercheurs. C'est probablement là le domaine où nos résultats auront *in fine* le plus d'importance. Nous sommes fiers d'avoir réussi à former avec une grande régularité des chercheurs courtisés par nos collègues universitaires. Sur 33 doctorants issus du LSV, 10 sont aujourd'hui enseignants-chercheurs à l'université et 14 sont chercheurs au CNRS (6), au CEA (5), à l'INRIA (2) et à la DCSSI (1), soit une proportion de 73%[5]. La réussite est aussi présente au niveau des maîtres de conférence, qui sont facilement recrutés sur des postes de professeurs dans d'autres universités.

Au delà de la satisfaction du travail accompli, ces succès contribuent à nouer un réseau dense de partenariats avec les principales universités françaises, et ils apportent une dynamisation de nos équipes en en permettant le renouvellement des membres.

Forte cohésion. Les quatre équipes du LSV opèrent dans des domaines d'application assez proches. Cette proximité est encore plus forte quand on se place au niveau des bases théoriques et des outils conceptuels sous-jacents : logique et automates, théorie algorithmique des jeux, théorie des modèles finis, *etc.* Il n'existe alors aucune frontière de spécialisation entre les équipes et chaque chercheur ou enseignant-chercheur peut tirer pleinement profit de toutes les compétences réunies au sein du laboratoire qui peut alors être vu comme une unique équipe. Cette cohérence conduit par exemple à ce que nos collègues étrangers tendent à voir le laboratoire comme un tout, au sein duquel ils ont souvent noué plusieurs relations interpersonnelles sans respecter les frontières d'équipes.

Au-delà de la forte unité thématique, le laboratoire reste habité par l'esprit de solidarité et la vision collective inspirés depuis l'origine du laboratoire par ses membres fondateurs.

Publications. Le nombre des publications des chercheurs du LSV dans les meilleures revues et conférences de notre domaine a connu une augmentation sensible qui va au-delà de la croissance des effectifs[6]. Cette croissance reflète principalement l'arrivée à maturité d'une nouvelle génération de jeunes chercheurs que les seniors du laboratoire ont formés.

Points faibles de l'unité

Notre situation n'est pas sans comporter certains risques ou faiblesses. À notre sens il s'agit principalement des aspects suivants :

[5] *Cf.* liste détaillée à l'url `http://www.lsv.ens-cachan.fr/People/anciens-doctorants.php`.

[6] Nous sommes passés de 34 revues et 66 conférences sur la période 2000—2003, un bilan déjà remarquable, à 58 revues et 135 conférences pour 2004–2007. *Cf.* analyse plus détaillée page 11.

Difficultés structurelles pour l'attractivité. La forte attractivité du LSV doit s'apprécier dans un contexte globalement très défavorable pour la recherche en général, pour la recherche en informatique en particulier, enfin pour la localisation en région parisienne. Ces dernières années la désaffection générale des carrières scientifiques a touché notre discipline d'une façon nettement plus dure qu'ailleurs puisque les informaticiens restent très demandés dans le secteur privé. En parallèle, la hausse des coûts du logement a rendu la situation des doctorants beaucoup moins attractive, en particulier en région parisienne. Il devient ainsi plus difficile d'attirer chez nous les meilleurs étudiants du territoire français. La même difficulté apparaît pour le recrutement des jeunes Chargés de recherche et Maîtres de conférences.

Absence de *killer application*. La vocation de nos résultats scientifiques est bien de s'incarner dans des outils de vérification automatique. Nos efforts en ce sens ont produit plusieurs logiciels expérimentaux innovants (citons FAST, Orchids, LTL2BA, Securify) mais ces prototypes sont jusqu'à présent restés réservés à un usage essentiellement académique.

Autres éléments d'appréciation

Production scientifique des quatre dernières années

Le tableau 1 mesure la production scientifique du laboratoire. Il ne tient pas compte des publications acceptées mais non encore parues ni des publications plus mineures (rapports techniques, conférences sans comité de sélection, invitations à des workshops, ...). La liste exhaustive est donnée pages 79 et suivantes.

	livres	chapitres de livres ou ouvrages collectifs	édition d'actes	revues internationales avec comité de lecture	conférences invitées	conférences internationales avec comité de lecture	thèses et habilitations
2004	1	2	1	9	3	24	2
2005	0	1	0	14	1	30	5
2006	1	2	5	22	1	38	5
2007	0	0	2	13	1	43	6
2008[a]	*0*	*3*	*2*	*11*	*2*	*41*	*0*

[a]Les chiffres de l'année 2008 sont arrêtés au 1er octobre 2008.

Tab. 1 – Répartition des publications du LSV

Ces chiffres méritent quelques commentaires et explications.

1. Ces dernières années, la production de publications a cru nettement plus rapidement que nos effectifs. Cette tendance apparaît clairement dans la colonne « conférences internationales » où se trouve l'essentiel de nos publications. Les irrégularités des autres colonnes proviennent des fluctuations propres aux petits échantillons, ou bien, dans le cas des articles de revues, au temps de latence assez grand —et variant d'un journal à l'autre— entre la soumission et la publication.
2. Les journaux les plus représentés dans cette liste sont *Theoretical Computer Science* (17 fois),

Information and Computation (16), *Information Processing Letters* (13), *Theory of Computing Systems* (5), *Mathematical Structures in Computer Science* (5), *Formal Methods in System Design* (5), *Logical Methods in Computer Science* (4), *Journal of Logic and Computation* (3), *Journal of Logic and Algebraic Programming* (3), *Journal of Automated Reasoning* (3).

3. Les conférences les plus représentées sont *FSTTCS* (14 fois), *FoSSaCS* (13), *CONCUR* (11), *FORMATS* (9), *ATVA* (9), *LICS* (7), *TACAS* (6), *ICALP* (6), *LPAR* (5), *TIME* (4), *LFCS* (4), *CSF* (4), *CAV* (3), *RTA* (3).
4. Enfin, tous les chercheurs et enseignants-chercheurs permanents du laboratoire sont « publiants » au sens des critères de l'AERES.

Enseignement et formation par la recherche, information et culture scientifique et technique

Le LSV est très conscient de l'importance de ses missions de **formation par la recherche** et il s'y implique pleinement. Cet effort porte sur deux fronts principaux : la formation des doctorants et l'enseignement au niveau Master 2 Recherche.

Formation des doctorants. Le LSV investit beaucoup en direction des doctorants. Aucun frein (sinon la pertinence scientifique) n'est mis quant à leur participation à des conférences internationales ou des écoles pour jeunes chercheurs. Tous les chercheurs et enseignants-chercheurs permanents sont incités à participer à l'encadrement de doctorants, de sorte que le ratio encadrants/doctorant est très élevé, et garantit une disponibilité excellente.

Les fruits de cet effort sont très encourageants : 42% des doctorants issus du LSV sont aujourd'hui chercheurs au CNRS, à l'INRIA ou au CEA. *Cf.* page 8.

Enseignement M2. Emmenés par Antoine Petit puis Paul Gastin, le LSV et le Département d'Informatique ont joué un rôle moteur dans la mise en place (en 2004) puis le développement du Master Parisien de Recherche en Informatique, le « MPRI », en habilitation partagée entre l'Université Paris-Diderot (P7), l'École Polytechnique et les Écoles normales supérieures de Cachan et Paris.

L'objectif du LSV a constamment été de permettre aux chercheurs du laboratoire d'intervenir dans les formations de niveau M2, ceci de façon à faciliter le recrutement de stagiaires.

Présence dans des comités de programme

Le rayonnement scientifique du LSV se mesure aussi au nombre et à la qualité des conférences dans les comités de programme desquelles nos chercheurs sont intervenus.

Comités de programme de conférences. Le tableau suivant dresse l'inventaire de nos participations à des comités de programme de ces quatre dernières années (en gras lorsqu'il s'agit de la présidence du comité de programme).

Nom	**Conférences internationales**	**Autres conférences, workshops, écoles, ...**
Michel Bidoit	CALCO'05, ICTAC'06	FROCOS'05

Nom	**Conférences internationales**	**Autres conférences, workshops, écoles, ...**
Patricia Bouyer	TACAS'05, FORMATS'05, CONCUR'06, **FORMATS'06**, FORMATS'07, TACAS'08, CONCUR'08, QEST'08, FORMATS'08	**GDV'06**, AVoCS'06, MSR'07, AVoCS'07, QAPL'08, MOVEP'08
Hubert Comon-Lundh	CSFW'05, LICS'05, IJCAR'06, FOSSACS'08, LPAR'08	
Stéphanie Delaune		WOTE'07, WOTE'08, FMSE'08
Stéphane Demri	RelMiC'06, TIME'06, TIME'07, AiML'08, **TIME'08**	M4M'05, **M4M'07**
Laurent Fribourg	SSS'08	
Paul Gastin	FSTTCS'05, CSR'07, MFCS'08	WATA'06
Jean Goubault-Larrecq	TABLEAUX'05, LPAR'06, CADE'07, RTA'08, LICS'08	FMSE'06
Florent Jacquemard	RTA'05, FOSSACS'07, CRiSIS'08	
Peter Habermehl		COSMICAH'05, GT-VC'06, GT-VC'07, **INFINITY'08**, MEMICS'08
Serge Haddad	ICCP'08, AICCSA'08	
Steve Kremer	SecUbiq'06, ISC'07, ISPEC'08, ISC'08, ICICS'08	IWAP'05, WOTE'06, **FCC'06**, ICS'06, IMIS'07, WOTE'07, WOTE'08, FMSE'08, **SecCo'08**
François Laroussinie	TIME'07	
Etienne Lozes		Geocal'06
Nicolas Markey	FORMATS'06, FORMATS'07	M4M'07
Philippe Schnoebelen	CONCUR'05, CSL'06, MFCS'07, ICALP'08	INFINITY'05, EXPRESS'06
Graham Steel	ASE'08	DISPROVING'04-07, SecReT'07, **ASA'08**
Ralf Treinen	LPAR'05, LPAR'07, RTA'07	**SecReT'07**

Comités de pilotage.

- P. Gastin est membre du comité de pilotage (*steering committee*) de la conférence internationale STACS, *Symposium on Theoretical Aspects of Computer Science* (depuis 2001). Président de la partie française de ce comité depuis 2005.
- R. Treinen est membre du comité de pilotage de la conférence RTA et du workshop SecReT.

Organisation de conférences internationales.

- P. Bouyer a été, avec E. Asarin, présidente de programme de la conférence FORMATS 2006

qui s'est tenue à Paris du 25 au 27 septembre 2006. Ella a aussi pris en charge l'organisation de cette conférence, aidés de nombreux chercheurs du laboratoire.
– R. Treinen a pros en charge l'organisation de la « Federated Conference on Rewriting, Deduction, and Programming (RDP'07) » qui a eu lieu du 25 au 29 juin 2007 (avec Antonio Bucciarelli, V. Padovani (PPS) et X. Urbain (Cedric)).

Comités éditoriaux de journaux internationaux.
– P. Gastin est membre du comité de rédaction du journal international JALC, « *Journal of Automata, Languages and Combinatorics* ».

Visiteurs étrangers de longue durée

Nos nombreuses collaborations internationales s'appuient en particulier sur les relations étroites que nous avons tissées avec des chercheurs reconnus. Elles ont grandement bénéficié du soutien fort de l'ENS Cachan *via* son programme des « professeurs invités ».

Dans le tableau ci-dessous, nous donnons la liste de nos visiteurs étrangers durant la période 2004–2008, en nous limitant à ceux ayant effectué un séjour d'au moins quatre semaines.

Nom	**Affiliation**	**Semaines**
Parosh Aziz Abdulla	Uppsala U.	4
Christel Baier	Technische U. Dresden	4
Adel Bouhoula	École Sup'Com, Tunis	12
Volker Diekert	U. Stuttgart	10
Manfred Droste	U. Leipzig	4
Deepak D'Souza	Indian Inst. Science, Bangalore	8
Vojtech Forejt	Masaryk U., Brno	5
Valentin Goranko	U. Witwatersrand, Johannesburg	14
Rolf Hennicker	Ludwig-Maximilians-U. München	6
Petr Jančar	Technical U. Ostrava	4
Carsten Kern	RWTH Aachen	4
K. Narayan Kumar	Chennai Math. Inst.	8
Kim G. Larsen	Aalborg U.	4
Sławomir Lasota	Warsaw U.	4
Ranko Lazić	U. Warwick	11
Chris Lynch	Clarkson U., Postdam	4
Madhavan Mukund	Chennai Math. Inst.	10
Ulf Nilsson	Linköpings U.	4
Hitoshi Ohsaki	National Inst. AIST, Osaka	9
Joël Ouaknine	Oxford U.	6
Joël Ouaknine	Oxford U.	4
Pavithra Prabhakar	Indian Inst. Science, Bangalore	4
Jean-François Raskin	U. Libre Bruxelles	4

Nom	Affiliation	Semaines
Mark Ryan	U. Birmingham	12
Roberto Segala	U. Verona	8
Helmut Seidl	Technische U. München	4
Jeremy Sproston	U. degli Studi Torino	12
Nadia Tawbi	U. Laval	5
Julie Vachon	U. Montréal	4
Bogdan Warinschi	U. Bristol	7
James Worrell	Oxford U.	6

Éthique et déontologie

Le laboratoire n'a pas mis en place d'instance formelle chargée de veiller au respect des règles d'éthique et de déontologie, la taille réduite de nos effectifs nous permettant de nous reposer sur l'autodiscipline collective ainsi que sur l'expérience et la vigilance des membres seniors du laboratoire.

Mentionnons certains aspects de cette vigilance :

Sur les publications. Le laboratoire est sensible aux aspects déontologiques de l'activité de publication, et il est clairement conscient des risques résultant de la pression croissante qui pousse les chercheurs à publier toujours plus, toujours plus vite. Dans ces conditions, il est remarquable de constater que nos travaux sont tous publiés dans des journaux et des conférences pratiquant une vraie évaluation.

Le recensement des publications du laboratoire s'accompagne d'une classification suivant les catégories usuelles (revue internationale avec comité de sélection, ...). Au LSV, cette classification n'est pas effectuée par l'auteur lui-même mais par le responsable de la bibliographie (N. Markey depuis 2004) en liaison avec le directeur.

Sur les expérimentations. Seul l'axe SECSI voit ses démarches expérimentales confrontées à des enjeux déontologiques. En effet, la recherche en sécurité informatique exige une connaissance concrète des attaques qui ne peut s'acquérir sans une certaine pratique. De telles expérimentations, même menées à des fins de recherche, sont interdites par la loi LCEN. Nous nous sommes donc équipés en 2004 de machines et de serveurs dédiés à nos expérimentations en détection d'intrusion, sur lesquels aucun utilisateur réel n'est jamais connecté.

Sur les thématiques sensibles. La sécurité informatique étant un enjeu de sécurité nationale, nos résultats dans ce domaine, même de nature fondamentale, n'ont pas tous vocation à être communiqués publiquement comme c'est le cas pour les avancées scientifiques en général. Les chercheurs de l'axe SECSI sont conscients de la tension existant entre les enjeux de visibilité scientifique et la sensibilité de certaines questions.

Bilan de la répartition des crédits en 2006 et 2007

Origine des crédits. Elle est détaillée dans les tableaux et les formulaires joints à ce document. On y observe que, dès sa création, l'ANR est devenue une source majeure de nos financements. Cette situation a influencé nos thématiques de recherche et nos stratégies de partenariats. Elle explique

en particulier la prééminence récente au LSV de projets de type exploratoire en collaboration avec des partenaires académiques français.

Ventilation des dépenses. La répartition des crédits se ventile comme indiqué dans le tableau suivant :

Dépenses du LSV sur la période 2006–07		
	2006	**2007**
Fonctionnement	118 431,70	80 936,54
Missions	110 800,54	117 636,08
Équipement	40 871,09	44 283,81
Personnels	46 929,13	86 247,02
Travaux		89 659,28
Total	317 032,46	418 942,73

Quelques explications ou commentaires éclairent la lecture de ce tableau.

1. Les dépenses ne sont pas ventilées équipe par équipe. La politique collective du laboratoire est de mettre en commun toutes nos ressources. Les crédits sont affectés à des projets ou bien au laboratoire dans son entier, pas aux équipes.
2. Les dépenses importantes affectées à des travaux en 2007 correspondent à une opération lourde mais ponctuelle et planifiée à l'avance : l'installation de la climatisation dans 14 bureaux.
3. Le montant des dépenses de type « Personnels » est quasiment impossible à interpréter utilement. Le laboratoire ne gère qu'une très faible partie des salaires de ses membres : quelques bourses de thèse ou de postdoctorat, parfois un contrat à durée déterminée pour un ingénieur ou un personnel administratif.
4. Les dépenses d'équipement sont presque tous des achats de matériel informatique très générique renouvelé régulièrement : les ordinateurs ou stations de travail avec lesquels travaillent tous les membres du laboratoire, des serveurs, des imprimantes partagées, *etc.* Notre recherche repose très peu sur l'utilisation de matériels spécifiques.

Formation permanente

Le laboratoire attache une grande importance à la formation continue des membres du laboratoire, quelle que soit leur grade ou leur fonction.

Doctorants. Les doctorants reçoivent une formation adaptée aux métiers de la recherche et de l'enseignement supérieur *via* leur participation à l'EDSP. Par ailleurs, la quasi totalité de nos doctorants sont bénéficiaires d'un monitorat qui leur permet d'acquérir une réelle expérience d'enseignement. Concernant les aspects plus pointus de la recherche, la participation à des écoles pour jeunes chercheurs est vivement encouragée.

Chercheurs et enseignants-chercheurs permanents. Parmi nos chercheurs, les plus jeunes sont encouragés comme les doctorants à participer à des écoles pour jeunes chercheurs. Par la suite, les formations les plus utiles pour eux sont de type managérial, les préparant à des fonctions d'animation d'équipe, et ceci dès qu'ils ont la responsabilité d'encadrer un doctorant. Ce type de formation est aussi suivie par A. Fliti, l'ingénieur qui au LSV a la responsabilité des systèmes et réseaux. Ici c'est le CNRS, *via* le service de formation continue de la délégation régionale, qui organise l'essentiel des formations auxquelles nos permanents participent.

Ingénieurs et personnels administratifs. L'évolution des outils informatiques d'assistance à la gestion du CNRS et de l'ENS Cachan, ainsi que la mise en place de nouvelles procédures (par exemple dans le cadre de la LOLF), requièrent des personnels administratifs une constante adaptation. Les formations correspondantes sont proposées par les organismes de tutelle. L'ENS Cachan propose aussi pour ses personnels des cours d'anglais qui aident à s'adapter à l'internationalisation croissante de la vie de la recherche.

Hygiène et sécurité

La thématique et les activités du laboratoire ne créent aucun risque spécifique pour le LSV ou ses membres.

Une politique de sécurité est néanmoins définie par le LSV, et c'est A. Fliti, notre ingénieur « systèmes et réseaux », qui est l'agent chargé de sa mise en œuvre (ACMO). Les points particuliers de cette politique sont :

- vigilance face aux attaques informatiques avec une politique stricte des mots de passe et des autorisations d'accès au système ;
- archivage automatisé des fichiers de travail de tous nos ordinateurs, de façon à ce que chaque membre du laboratoire puisse surmonter facilement un crash disque ou une erreur de manipulation ;
- utilisation d'écrans confortables, surveillance des nuisances sonores causées par les ordinateurs, et des sensibilités possibles aux ondes électromagnétiques.

Annexe A

Bilan scientifique détaillé : axe TEMPO

Membres de l'axe TEMPO

Responsables jusqu'en 2007
- François Laroussinie (MCF ENS Cachan, jusqu'en août 2007 ; en délégation CNRS au LSV, 2004-2007) ;

Responsables depuis 2008
- Patricia Bouyer (CR CNRS, en disposition à l'université d'Oxford, 2007)
- Nicolas Markey (CR CNRS, depuis oct. 2004) ;

Membres permanents
- Benedikt Bollig (CR CNRS, depuis oct. 2005) ;
- Thomas Chatain (MCF ENS Cachan, depuis sept. 2007) ;
- Stéphane Demri (CR CNRS)[1] ;
- Laurent Fribourg (DR CNRS) ;
- Paul Gastin (PR ENS Cachan, depuis sept. 2004) ;
- Serge Haddad (PR ENS Cachan, depuis fév. 2008) ;
- Antoine Petit (PR ENS Cachan, en détachement CNRS jusqu'en fév. 2006 ; directeur du CR Paris-Rocquencourt de l'INRIA depuis juil. 2006) ;
- Claudine Picaronny (MCF ENS Cachan) ;

Membres non-permanents
- Laura Bozzelli (Postdoc CNRS, 2005-2006)
- Thomas Brihaye (Postdoc CNRS, 2006-2007)
- Emmanuelle Encrenaz-Tiphène (MCF Université Paris 6, en délégation CNRS au LSV, 2005-2007)
- Stéphane Riedweg (ATER ENS Cachan, 2004-2005)
- Weiwen Xu (Postdoc CNRS, 2005-2006)

Doctorants
- S. Akshay (depuis sept. 2006)
- Étienne André (depuis sept. 2007)

[1] Stéphane Demri a également participé à l'axe SECSI et à l'axe INFINI. Il est maintenant membre de l'axe DAHU.

– Manuel Baclet (2002-2005)
– Houda Bel mokadem (2003-2006)
– Florent Bouchy (depuis sept. 2006)
– Najla Chamseddine (depuis sept. 2006)
– Fabrice Chevalier (2004-2007)
– Arnaud Da Costa (depuis sept. 2007)
– Diego Figueira (depuis sept. 2007)
– Régis Gascon (2004-2007)
– Stéphane Messika (2002-2005)
– Ghassan Oreiby (depuis sept. 2005)
– Simon Pinot (2005-2006, thèse abandonnée)
– Pierre-Alain Reynier (2004-2007)
– Tali Sznajder (depuis sept. 2005)

Évolution de l'axe de recherche

Le *model-checking* consiste à modéliser un système, par exemple sous la forme d'un système de transitions (comme celui associé à un automate fini), à représenter ses propriétés, par exemple par une formule d'une logique adaptée, et à appliquer des algorithmes efficaces permettant de certifier que la propriété est vérifiée par le modèle du système, ou bien au contraire ne l'est pas, en produisant un exemple d'exécution du système violant cette propriété. Comme le rappelle cette année l'attribution du prix Turing à Clarke, Emerson et Sifakis, fondateurs de la méthode, ce sont les algorithmes et les résultats théoriques issus de la communauté académique, qui ont abouti à des outils industriels, devenus aujourd'hui indispensables à la conception et fabrication des systèmes intégrés sur puce, systèmes critiques complexes, protocoles de communication, ...

La prise en compte d'aspects temps-réel a conduit dans les années 1990 à l'extension du modèle classique des automates finis en celui des *automates temporisés*. Ces automates sont munis de variables réelles, appelées *horloges*, qui évoluent de façon synchrone et continue avec le temps. Les logiques temporelles usuelles ont elles aussi été étendues pour prendre en compte des contraintes quantitatives sur les délais. Bien qu'avec l'introduction du temps continu l'espace des états possibles du système devienne infini (un état étant maintenant l'association d'un état de contrôle et d'une valeur réelle pour chaque horloge), l'*abstraction du graphe des régions*, proposée par Alur et Dill en 1990, permet, pour des propriétés simples, de ramener le problème du model-checking des systèmes temporisés à un problème de model-checking sur un système fini. Depuis lors, le model-checking des systèmes temporisés s'est largement développé dans la communauté et est devenu un domaine de recherche très actif. Ces dernières années, nos recherches ne se sont pas exclusivement focalisées sur la vérification des systèmes temporisés, mais ce thème de recherche est néanmoins central dans l'axe TEMPO. Depuis sa création, l'axe entretient notamment de très fécondes relations avec l'équipe de Kim G. Larsen à Aalborg (Danemark) qui développe l'outil de model-checking temporisé Uppaal.

Durant ces quatre dernières années, nous avons apporté de nombreuses contributions à ce domaine de recherche (paragraphe A.1), que ce soit au niveau des extensions du modèle de base, au niveau des logiques temporisées, ou au niveau des problèmes algorithmiques associés. Par exemple, nous avons beaucoup travaillé sur le modèle des automates temporisés avec coûts, permettant de modéliser des systèmes avec des contraintes d'énergie (qui apparaissent de manière naturelle dans les systèmes embarqués), et avons étudié la décidabilité de propriétés telles que l'atteignabilité d'un état en deçà d'un certain coût. Nous avons par ailleurs élargi le cadre de cette recherche dans deux importantes directions.

Nous avons d'abord étudié de nombreux problèmes liés aux systèmes qui sont *ouverts* aux actions extérieures, typiquement les actions provenant de l'environnement (paragraphe A.2). Le problème du model-checking traditionnel se généralise alors en un problème de contrôle ou de synthèse de contrôleur : étant donné un système ouvert S, peut-on synthétiser un système C, appelé contrôleur, de telle façon que la composition $S\|C$ vérifie une propriété donnée, quelle que soit le comportement de l'environnement ? La question se formule naturellement dans un cadre de théorie des jeux : on modélise le système par un jeu temporisé entre le contrôleur et l'environnement (sous forme d'un automate avec des transitions incontrôlables), et le problème se ramène à celui de l'existence d'une stratégie gagnante pour le contrôleur. Dans ce cadre, nous avons travaillé avec des logiques comme ATL (Alternating-time Temporal Logic), qui généralise la logique classique CTL en autorisant des quantifications sur les stratégies dans les formules.

Dans une autre direction, nous avons intégré d'importants aspects des systèmes *distribués*, dans lesquels les composants, organisés en réseau, n'ont qu'une vue locale de leur environnement (paragraphe A.3). Pour cette extension, nous avons également considéré la question du contrôle, et utilisé de nouveaux outils de preuve comme la théorie des jeux distribués. Nous avons également exploité les techniques d'*ordre partiel*. Ces techniques, en exploitant l'indépendance de certains évènements entre eux, permettent de réduire le phénomène d'explosion combinatoire provoquée lors de la composition des entités du système distribué. Nous avons, dans ce contexte, proposé ou revisité des extensions temporisées de modèles distribués comme les réseaux de Petri et les diagrammes de séquences. Ceci nous a par exemple amenés à proposer des algorithmes originaux de dépliage pour les automates temporisés.

Nos contributions dans les deux thèmes principaux mentionnés ci-dessus ont suscité un très grand nombre de chapitres de livres et de conférences invitées. Si les aspects théoriques (définition de nouveaux modèles, de nouvelles logiques, problèmes de décidabilité, complexité, expressivité, ...) sont très présents dans ces travaux, ils s'inscrivent en étroite liaison avec des problèmes pratiques rencontrés dans l'analyse des implémentations d'algorithmes de model-checking ou lors du traitement d'études de cas issus du monde industriel. Citons ainsi : la détection d'un bug dans un algorithme de base du model-checking temporisé et sa correction par de nouveaux algorithmes d'abstraction (paragraphe A.1.1) ; l'extension du modèle avec mises à jour, motivée par l'étude d'un protocole de France-Telecom (paragraphe A.1.2) ; l'extension de la sémantique pour ignorer les états transitoires, motivée par l'étude d'automates programmables industriels (paragraphe A.1.4) ; les extensions du model-checking temporisé aux systèmes ouverts, qui permettent de rendre compte de problèmes d'optimisation comme ceux rencontrés dans l'analyse d'un circuit-mémoire de ST-Microelectronics (paragraphe A.5.1).

Par ailleurs, nos recherches nous ont amenés à développer des algorithmes et des outils, nous les mentionnerons dans les paragraphes appropriés. Mentionnons par exemples les outils MScan et Smyle qui ont été développés pour l'analyse et l'apprentissage des diagrammes de séquences (paragraphe A.3.2), ainsi que l'outil de transformation de formules LTL en automates de Büchi (paragraphe A.4.2).

Les contrats dans lesquels l'axe TEMPO est impliqué sont listés dans la partie A.6.

A.1 Systèmes temporisés

Nos travaux sur le modèle des automates temporisés, ses extensions, et les logiques et algorithmes associés peuvent se décomposer ainsi :

1. de nouveaux algorithmes d'exploration d'état, mis au point pour contourner le *bug* que nous avons découvert en 2004 dans l'algorithme communément implanté dans les outils comme Uppaal (paragraphe A.1.1),

2. des travaux sur d'importantes variantes du modèle de base (paragraphe A.1.2) dont :
   - les automates temporisés avec mise à jour, rencontrés dans l'analyse de protocoles de communication,
   - les automates à coûts — qui permettent de modéliser des contraintes quantitatives plus générales que les durées,
   - les modèles *simplement* temporisés (à une seule horloge), qui offrent un intéressant compromis expressivité—complexité,
   - des extensions probabilistes qui permettent de modéliser des éléments randomisés d'algorithmes, ou rendre compte d'un environnement obéissant à des lois stochastiques.
3. de nombreux travaux sur les logiques temporisées (paragraphe A.1.3) : décidabilité, complexité, expressivité, ...
4. de nouvelles sémantiques ont été proposées pour les automates temporisés, qui doivent permettre de prendre en compte de manière plus fine les caractéristiques des systèmes temps-réel (paragraphe A.1.4) dont :
   - une sémantique robuste, qui permet de tenir compte de l'imperfection des systèmes réels (par exemple l'imprécision des horloges digitales),
   - une sémantique probabiliste qui, au contraire de la précédente, ne prend en compte que les comportements qui ont une certaine chance (au sens probabiliste) d'arriver,
   - une sémantique nouvelle pour les logiques temporisées, permettant d'abstraire les états transitoires des systèmes temporisés.
5. des travaux sur le raffinement et l'abstraction des automates temporisés.

Ces recherches ont fait l'objet de nombreuses synthèses au travers d'exposés invités [In-38, In-33, In-40, In-25, In-24, IN-8] et de chapitres de livres [Ch-7, Ch-4].

A.1.1 Analyse en avant

Les propriétés d'accessibilité jouent un rôle fondamental dans toute démarche de vérification. Pour vérifier ce type de propriétés, l'algorithme le plus utilisé dans les outils de model-checking temporisé est l'algorithme d'analyse en avant. Après avoir mis à jour une erreur fondamentale dans cet algorithme de base, nous avons exploré les possibilités de le réparer. Nous avons aussi travaillé sur des abstractions utilisables dans cet algorithme permettant de le rendre plus efficace.

Détection et correction du *bug* dans l'analyse en avant des automates temporisés. Les outils actuels, comme Uppaal et Kronos, implémentent des algorithmes symboliques parcourant l'espace des états accessibles en avant (en partant de l'état initial) ou en arrière (en partant d'un état final). Pour des raisons d'efficacité, l'analyse en avant est la plus utilisée. Or il peut arriver que l'analyse en avant ne termine jamais. Pour pallier ce problème, les outils intègrent une abstraction, paramétrée par une constante, souvent choisie comme étant la constante maximale apparaissant dans l'automate. L'abstraction est dite *correcte* si elle permet de calculer exactement l'ensemble des états accessibles et non un sur-ensemble strict. Nous avons montré que les esquisses de preuve de correction faites dans le passé étaient toutes erronées, puis montré que l'algorithme implémenté dans les outils existants depuis le milieu des années 1990 était en fait incorrect [RI-72]. Nous avons ensuite dégagé une sous-classe de systèmes (qui exclut les contraintes dites *diagonales*, qui comparent des horloges entre elles) pour laquelle nous avons prouvé que l'abstraction usuelle est correcte. Depuis la découverte de ce *bug*, l'algorithme implanté dans Uppaal retire les gardes diagonales (à la volée), au prix d'une perte potentiellement exponentielle de concision [RI-54]. Nous avons alors conçu un nouvel algorithme basé sur le raffinement par contre-exemple, qui autorise les contraintes diagonales [Ra-42, CI-134]. Cet algorithme a été implanté dans Uppaal avec des résultats expérimentaux convaincants [Ra-21].

Analyse statique, abstraction des automates temporisés. Le précédent travail de recherche nous a conduits naturellement à raffiner l'opérateur d'abstraction utilisé dans Uppaal pour forcer la terminaison du calcul en avant. Nous avons développé en 2004 une abstraction consistant à distinguer les constantes bornant supérieurement les variables d'horloge de celles les bornant inférieurement, de manière à simplifier les tests d'encadrement de ces variables. Ceci nous a amenés à proposer un nouvel opérateur d'abstraction plus performant, qui a été substitué à l'ancien dans l'outil Uppaal [CI-181, RI-41].

A.1.2 Extensions ou restrictions du modèle des automates temporisés

Suivant l'application que l'on considère, une variante du modèle des automates temporisés s'impose souvent plutôt qu'une autre. L'étude comparative de ces variantes est donc importante afin de déterminer le meilleur compromis entre expressivité et complexité de calcul. Cette observation a motivé notre étude de différentes extensions ou restrictions du modèle de base.

Automates temporisés avec mises à jour. Dans le cadre d'un projet RNRT avec France-Telecom, nous avions été amenés à définir une extension des automates temporisés, les *automates temporisés avec mises à jour*. Dans ce modèle, des opérations plus riches que des remises à zéro peuvent être effectuées sur les horloges lorsqu'une transition de l'automate est franchie. Par exemple, il est possible de réinitialiser une horloge à une valeur quelconque plus grande qu'une certaine constante, ou bien elle peut prendre la valeur d'une autre horloge, *etc.* Ce modèle offre des possibilités supplémentaires de modélisation, et inclut notamment le modèle proposé pour analyser le protocole ABR de France-Telecom. Étendant la construction du graphe des régions, nous avons dégagé de larges sous-classes décidables, et avons montrer l'indécidabilité de la plupart des extensions [RI-70, Ra-43], obtenant ainsi une description assez fine des modèles qui pourront être utilisés à des fins de vérification. Nous avons aussi étudié le pouvoir d'expression de ce modèle, en nous attachant plus particulièrement aux classes décidables, et avons montré que celles-ci ne sont pas plus expressives que les automates temporisés, mais qu'elles permettent de représenter de manière beaucoup plus compacte de larges classes de systèmes temporisés [RI-54]. Ces travaux ont fait l'objet de l'exposés invité [In-32].

Automates temporisés à coûts. De multiples contraintes quantitatives (coûts, consommation d'énergie, *etc.*) doivent être satisfaites par les systèmes embarqués. Nous en avons déjà vu une illustration en sécurité, à la section C.6.2. Il est alors naturel de s'intéresser à des problèmes comme l'accessibilité d'un état, non pas dans l'absolu, mais en deçà d'un certain coût. Or, le coût associé à une exécution dépend souvent de la nature des états traversés ainsi que de la nature des transitions franchies le long de l'exécution. La formulation d'une telle fonction de coût n'est pas possible dans le cadre classique des automates temporisés, où seule la somme des durées passées dans chaque état peut être prise en compte. En 2001, le modèle des automates temporisés à coûts a été proposé indépendamment par les groupes de Rajeev Alur (U. Pennsylvania, USA) et Kim G. Larsen (U. Aalborg, Danemark), pour prendre en compte de telles contraintes quantitatives. Dans ce modèle, une exécution a un coût qui dépend des temps d'attente dans chaque état et des transitions franchies. Le but des travaux réalisés en 2001 était de calculer des coûts optimaux (pour atteindre un état fixé), ainsi que de calculer des stratégies pour réaliser (ou approcher) ces coûts optimaux. Nous avons approfondi ces travaux et montré que ce problème, bien que d'apparence plus compliqué, pouvait se résoudre aussi efficacement qu'une propriété d'accessibilité classique dans les automates temporisés [RI-23]. Nous avons également étudié le problème de l'optimisation du coût en moyenne dans les automates temporisés, et avons proposé une abstraction, plus fine que l'abstraction du graphe des régions, qui permet de ramener le

problème à celui de trouver des cycles optimaux dans un graphe fini, problème que nous avons résolu en appliquant l'algorithme développé par Karp dans les années 1970 [CI-182, RI-16].

Nous avons également étudié les problèmes de model-checking de différentes extensions quantitatives des logiques temporelles classiques. Contrairement aux propriétés « simples » dont nous venons de parler, le model-checking pour la logique WCTL, une extension pondérée naturelle de CTL permettant d'exprimer des propriétés comme « *si la machine fonctionne correctement pendant* 8 *heures, alors il y aura moyen de la réparer en un coût inférieur à 560* € », a été montré indécidable par Thomas Brihaye, Véronique Bruyère et Jean-François Raskin en 2004. Nous avons simplifié et amélioré la portée de leur preuve dans [RI-42]. D'autre part, nous avons montré qu'en nous restreignant aux automates temporisés n'ayant qu'une seule horloge, le model-checking de WCTL était décidable [CI-87] alors que celui de de WMTL, une extension naturelle de LTL avec des coûts, était indécidable [CI-59]. Ce résultat positif pour WCTL, bien que reposant sur une restriction assez forte, ouvre des perspectives de vérification de propriétés quantitatives compliquées dans des systèmes où la temporisation est restreinte.

Ces travaux ont fait l'objet d'articles de synthèse [IN-6] et d'exposés invités [In-28, In-26, In-27].

Modèles simplement temporisés. L'analyse des automates temporisés requiert des algorithmes particuliers et bien plus coûteux que ceux utilisés pour les systèmes non temporisés. Ainsi l'accessibilité d'un état de contrôle est déjà un problème PSPACE-complet, et les problèmes de model-checking ont une complexité encore plus élevée, quand ils ne sont pas indécidables. Ces résultats ont motivé la recherche de modèles intégrant une notion plus simple de temps-réel mais permettant une algorithmique efficace. Nous avons d'abord considéré des modèles où l'on associe à chaque transition discrète un intervalle d'entiers représentant les durées possibles de la transition. Nous avons proposé un algorithme plus efficace pour vérifier les propriétés écrites dans le fragment *sans égalité* de la logique TCTL [RI-48]. Nous avons aussi étendu l'outil NuSMV avec notre algorithme [CI-171, CO-28, CO-36]. Ces travaux se sont prolongés aux automates temporisés avec une unique horloge. Une telle restriction limite bien sûr l'expressivité du modèle mais reste plus riche que celle des modèles à temps discret et peut se révéler suffisante en pratique pour certains systèmes. De plus, nos algorithmes polynomiaux ont pu être adaptés aux automates temporisés n'ayant qu'une seule horloge [CI-173]. Notons que le passage à deux horloges entraîne un net saut de complexité [Ra-24].

Systèmes temporisés probabilistes. Enfin, nous avons considéré des extensions probabilistes d'automates temporisés. Dans ces modèles probabilistes, les actions discrètes sont remplacées par des distributions d'actions. Le premier cadre concerne les automates simplement temporisés mentionnés précédemment, pour lesquels nous avons proposé des extensions d'algorithmes efficaces pour leur model-checking temporisé [CI-158, CI-90].

Le deuxième cadre concerne les automates temporisées, appelés « Determinate Probabilistic Timed Automata » (DPTA), pour lesquels il n'y a aucun non-déterminisme : d'une part, on ne peut rester qu'un temps déterminé dans un état de contrôle donné ; d'autre part, on ne peut sortir d'un tel état de contrôle qu'au moyen d'une seule (distribution d') action(s). Ce genre d'automate rend souvent compte du *pire* comportement d'un automate temporisé probabiliste général, où « pire » caractérise ici le comportement engendrant le temps moyen *maximal* pour atteindre un (ensemble d') état(s) absorbant(s). L'intérêt d'un DPTA est que l'on peut calculer le temps moyen d'asorption associé de façon *paramétrique*, dans le sens où les valeurs maximales d'horloges de l'automate ne sont pas des entiers mais des paramètres [Ra-16].

A.1.3 Logiques temporelles temporisées

Les logiques temporelles permettent d'exprimer des propriétés plus intéressantes que de simples propriétés d'accessibilité. Dans le cadre des automates finis (non temporisés), deux logiques se sont imposées : la logique du temps arborescent CTL et la logique du temps linéaire LTL. Nous avons continué le travail entrepris dans le précédent contrat quadriennal, de comparer les différentes extensions temporisées de ces logiques, notamment du point de vue de leur expressivité. Nous avons aussi travaillé sur l'extraction de sous-classes ou modèles pour lesquels le model-checking peut être fait efficacement. Rappelons enfin que nous avons travaillé sur l'intégration des nouvelles notions de coûts (voir paragraphe A.1.2) dans les logiques temporisées.

Expressivité des logiques temporelles temporisées. En parallèle à l'introduction et au développement des automates temporisés dans le cadre de la vérification, les logiques temporelles ont été étendues pour permettre d'exprimer des propriétés quantitatives sur ces modèles. Au début des années 1990, deux extensions de LTL ont été proposées : l'une, MTL, consistait à étiqueter les modalités par des contraintes temps-réel (exprimant par exemple qu'une propriété sera vraie dans au plus 5 unités de temps) ; l'autre, TPTL, utilise des horloges qui peuvent être mises à zéro au cours de l'exécution, puis comparées à un entier lorsqu'un évènement survient. Répondant à une conjecture vieille d'une quinzaine d'années, nous avons prouvé, dans [CI-131], que la logique TPTL est strictement plus expressive que la logique MTL. Un autres aspect de l'expressivité des logiques temporelles a été étudié dans [CI-95] et [CI-60] : il s'agit de relier l'expressivité des logiques temporelles et celles d'extensions d'automates temporisés, en comparant les *langages temporisés* pouvant être définis par chacun de ces formalismes. Les résultats ainsi obtenus étendent les résultats classiques reliant la logique LTL, la logique du premier ordre, et les automates de Büchi sans compteur.

Décidabilité et complexité. Depuis le début des années 1990, il est connu que le model-checking des extensions temporisées MTL et TPTL de LTL est indécidable. Les preuves d'indécidabilité reposent principalement sur l'utilisation de contraintes d'égalités (exprimant par exemple qu'un évènement a lieu exactement 1 unité de temps après un autre). Lorsqu'on interdit les contraintes exactes dans ces logiques, on obtient la logique appelée MITL, dont le model-checking est décidable. Il était donc largement admis que l'utilisation de contraintes d'égalités dans ces logiques rendaient le model-checking indécidable.

Dans [CI-72] puis [CI-25], nous montrons qu'il n'en est rien : nous définissons la logique CoFlatMTL, qui autorise les contraintes ponctuelles mais restreint l'emboîtement de contraintes non-bornées, et montrons que son problème de model-checking est décidable, avec la même complexité que pour MITL.

Model-checking de chemins. Dans de nombreuses applications, on souhaite vérifier des propriétés temporelles sur des modèles « déterministes » : c'est le cas pour l'analyse de *logs*, dont l'objet est de détecter des comportement anormaux sur les enregistrements de l'historique d'un système informatique par exemple (section C.6.1).

D'autres applications concernent l'analyse des contre-exemples produits par des model-checkers classiques. Le *model-checking de chemins* a ainsi été défini au LSV en 2003 dans le but d'obtenir des algorithmes plus efficaces pour tirer profit du caractère « linéaire » du modèle étudié. Nous avons étendu cette approche au cadre temporisé. Dans ce cadre, un *chemin* peut avoir plusieurs significations : soit une seule exécution d'un système temporisé, soit une exécution symbolique, comme par exemple une suite de régions ou de zones, qui représente alors en général une infinité d'exécutions. Nos résultats sont contrastés mais instructifs : la restriction du model-checking de

MTL et TPTL aux chemins symboliques ultimement périodiques reste indécidable. Par contre, pour les chemins symboliques finis, nous montrons que le problème est décidable, de même que pour les chemins simples [CI-176, RI-39].

A.1.4 Autres sémantiques pour les automates temporisés

Les automates temporisés sont un modèle mathématique très pratique pour la vérification formelle, mais leur sémantique s'éloigne quelque peu des comportements de machines réelles, digitales et imprécises. Nous avons été amenés à relâcher la sémantique habituelle des automates temporisées, c'est-à-dire à augmenter ou restreindre l'ensemble des comportements acceptables, de diverses façons : pour autoriser des actions qui se produisent lorsque les horloges satisfont les gardes, non pas exactement, mais « à ε près » (sémantique robuste) ; pour supprimer des comportements problématiques s'ils n'arrivent presque jamais dans un sens probabiliste ou temporisé (c'est-à-dire, s'ils se produisent avec probabilité nulle ou durant une période de durée nulle).

Sémantique robuste. Afin de vérifier des propriétés temps-réel de systèmes informatiques, nous les modélisons par des automates temporisés : nous représentons ainsi un système physique, généralement digital et imprécis, par un modèle mathématique idéalisé. Les transitions du modèle mathématique sont instantanées, les synchronisations sont parfaites, ... Bien que la vérification formelle apporte des informations intéressantes sur le système, rien ne permet de dire que les propriétés vérifiées sur le modèle mathématique sont vraiment satisfaites par le système physique. Afin de prendre en compte cet aspect, une variante de la sémantique des automates temporisés a été proposée : elle consiste à « élargir » les contraintes apparaissant dans les automates temporisés, en remplaçant par exemple « $x \leq 5$ » par « $x \leq 5 + \varepsilon$ », où ε est un paramètre. Cette nouvelle sémantique ajoute des comportements par rapport à la sémantique habituelle, et il a été montré qu'elle modélise bien, en un certain sens, l'imprécision des systèmes informatiques par rapport au modèle des automates temporisés. Dans [CI-169], nous avons étudié le problème de la sûreté pour cette sémantique étendue. Nous proposons pour cela un algorithme basé sur l'automate des régions, et dont la complexité théorique est identique à celle du problème d'accessibilité dans les automates temporisés avec la sémantique classique. Forts de ce premier résultat, nous avons étendu cette étude à la logique LTL dans un premier temps [CI-129], en passant par des automates de Büchi, puis à la logique CoFlatMTL [CI-42] (*cf.* section A.1.3), en utilisant des machines à canaux.

Sémantique probabiliste. Nous avons également défini et étudié une autre sémantique pour les automates temporisés, partant de la constatation que tous les systèmes embarqués ne sont pas forcément critiques, au sens où une défaillance serait fatale. Par exemple, une erreur qui arrive rarement dans le logiciel d'un téléphone portable peut être tolérable, alors qu'elle ne le serait pas dans le système de freinage d'une voiture. L'idée est alors de ne considérer que les ensembles d'exécutions qui ont une certaine chance d'arriver, et d'oublier ceux qui vont presque sûrement ne jamais arriver. La sémantique probabiliste que nous utilisons généralise le modèle des chaînes de Markov en temps dense. On donne une mesure à l'ensemble des exécutions d'un automate temporisé qui vérifient une propriété, par exemple exprimée dans la logique LTL. Cela permet de définir un problème de model-checking presque-sûr, où l'ensemble des exécutions de l'automate vérifiant une propriété donnée doit être de probabilité 1. Nous avons alors montré, en utilisant des techniques basées sur les jeux topologiques de Banach-Mazur, que le model-checking presque-sûr de LTL sur les exécutions finies était décidable [CI-51], et que si l'on restreint le modèle à une

horloge, alors il est aussi décidable sur les exécutions infinies, en construisant une chaîne de Markov finie abstrayant correctement le problème [CI-29].

Nouvelles sémantiques pour l'opérateur « Until » de la logique TCTL. Ces travaux ont été motivés lors d'une étude de cas soumise par le LURPA (Laboratoire Universitaire de Recherche en Production Automatisée de l'ENS Cachan), par le souci d'abstraire les états intermédiaires (où l'on ne passe que durant un temps de durée nulle) de façon à prouver des propriétés qui sont vraies "presque partout" le long d'une exécution. Nous avons ainsi introduit des opérateurs U^k où k est un entier spécifiant la durée maximale des séquences d'états transitoires : U^0 permet d'ignorer les séquences d'états de durée nulle, U^2 les séquences de durée 2, *etc.* Nous avons montré que ces nouveaux opérateurs n'étaient pas exprimables avec la version standard de TCTL et que le model-checking était possible en utilisant les techniques classiques (sans surcoût de complexité). Nous avons aussi considéré une autre sémantique pour les U^k avec $k > 0$ et montré qu'elle conduisait à l'indécidabilité [CI-140, CI-101, TH-8].

A.1.5 Raffinement, abstraction

Nous avons étudié les opérations de raffinement et d'abstraction pour les automates temporisés. Ces opérations sont essentielles car elles permettent de développer un système temps-réel par raffinements successifs ou d'en étudier les propriétés au moyen d'abstractions. Pour les systèmes temps-réel les plus généraux qui allient événements et signaux, nous avons prouvé au moyen de constructions effectives que les automates temporisés sont stables par substitutions (raffinements) et substitutions inverses (abstractions) [CI-107, CI-106, RI-22]. Ces travaux ont aussi donné lieu à deux exposés invités dans des workshops internationaux [In-10, In-17].

A.2 Systèmes ouverts, contrôle, jeux

Lorsque les lois régissant le système que l'on veut modéliser sont complètement spécifiées, on dit qu'on a affaire à la modélisation d'un système « fermé ». En revanche, lorsque le système modélisé interagit avec un environnement (par exemple *via* des capteurs, des interrupteurs, ou même de manière plus continue, comme une cuve d'eau à température ambiante) dont on ne connaît pas les comportements, on dit que le système est « ouvert ». Il faut alors tenir compte du fait que l'on ne peut pas agir sur toutes les évolutions du système, car certaines actions pourront être dues à l'environnement. La synthèse de contrôleur consiste à construire un programme, le *contrôleur*, qui pilote le système de manière à ce qu'il satisfasse la propriété souhaitée, quoi que fasse l'environnement. Dans le cadre classique non-temporisé, les problèmes de synthèse de contrôleurs ont été beaucoup étudiés, en lien avec les problèmes de jeux à deux joueurs (système contre environnement). Dans le jeu qui modélise un tel problème, une position correspond à un couple formé de l'état du système et de l'état de l'environnement ; une *stratégie* est alors un ensemble de règles qui dit au contrôleur ce qu'il doit faire en fonction de la position du jeu ; une stratégie est *gagnante* si le contrôleur, quand il suit ces règles, gagne quelle que soit la réaction de l'environnement. Le problème de la synthèse d'un contrôleur se ramène ainsi à l'existence d'une stratégie gagnante pour un jeu approprié. Dans le cadre temporisé, les nouvelles contraintes (sur le temps) augmentent la difficulté du problème, et d'autres questions se posent naturellement, comme celle de la synthèse du contrôleur qui assure un *temps* optimal pour assurer la propriété, quel que soit le comportement de l'environnement. Plus généralement, dans le cas des automates temporisés à coûts, la fonction optimisée par le contrôleur est la fonction de coût elle-même (et non plus la fonction de temps). L'analyse des systèmes ouverts pose donc des problèmes nouveaux, et impose notamment la définition de logiques et de méthodes adaptées.

De nombreux résultats ont été obtenus par l'axe TEMPO dans l'étude des systèmes ouverts. Ils concernent notamment la synthèse de contrôleurs dans une classe de systèmes hybrides et pour les systèmes temporisés à coûts (paragraphe A.2.1). Nous avons d'autre part obtenu des résultats d'indécidabilité pour le problème de contrôle dans le cadre voisin d'*observabilité partielle* où certaines opérations sont complètement masquées (paragraphe A.2.2). Le problème du contrôle nous a amenés naturellement à réfléchir aux extensions des logiques temporelles temporisées (paragraphe A.2.3), et notamment à des extensions au cadre temporisé de la logique temporelle alternante ATL. Enfin, nous avons travaillé sur différentes applications de problèmes d'optimisation des temps de réponse, notamment dans le cadre des circuits mémoire asynchrones (paragraphe A.5.1) concernant notamment l'identification d'architectures pour lesquelles le problème de synthèse de contrôleurs devient décidable, l'expressivité, la complexité et l'extension de la logique de contrôle ATL, ainsi que la complexité du problème d'observabilité partielle dans le cadre temporisé. Ces recherches ont fait l'objet d'exposés invités [In-31] et d'articles de synthèse [In-21, RE-2, In-7, Ed-7].

Notons aussi que les systèmes ouverts ont aussi des applications naturelles en sécurité, menant ainsi à des modèles et des algorithmes pour la prédiction de vulnérabilités de réseaux. Ceci a été mentionné à la section C.6.2.

A.2.1 Synthèse de contrôleur

Nous nous sommes d'abord intéressés au problème du contrôle dans le cadre des systèmes hybrides o-minimaux, une sous-classe des automates hybrides dans lesquels les variables continues de temps n'évoluent pas toutes uniformément à la différence des systèmes temporisés usuels. Nous avons également étudié le problème de la synthèse d'un contrôleur optimal dans le cas de systèmes temporisés à coûts. Ces travaux ont fait l'objet d'un exposé invité [In-31] et d'articles de synthèse [In-21, RE-2, In-7, Ed-7].

Accessibilité pour les systèmes hybrides o-minimaux. Les systèmes hybrides ont été développés dans les années 1990, à peu près en même temps que les systèmes temporisés. Ces systèmes sont plus généraux que les systèmes temporisés, dans le sens où ils manipulent des variables qui peuvent suivre des lois d'évolution assez complexes, contrairement aux horloges dans les systèmes temporisés, qui évoluent toutes à la même vitesse que le temps universel. Ils permettent de modéliser de nombreux aspects quantitatifs des systèmes embarqués, mais en contrepartie, ils sont indécidables. Plusieurs sous-classes décidables ont été dégagées, notamment la classe des systèmes hybrides o-minimaux, dans laquelle les variables sont réinitialisées à chaque changement d'état discret. Ces systèmes permettent d'exprimer des dynamiques très riches, mais des changements discrets assez restreints, ce qui leur permet de représenter plus naturellement des phénomènes de phase. Nous avons prouvé que le problème de contrôle d'accessibilité pour ces modèles était décidable grâce à une nouvelle abstraction, la *partition suffixe*, qui raffine la partition habituelle de la bisimulation [CI-115].

Coût optimal. Dans le cadre du contrôle des systèmes temporisés à coûts, l'un des problèmes de base est le problème du contrôle optimal (paragraphe A.1.2), c'est-à-dire, la synthèse d'un contrôleur permettant d'assurer un coût minimal, quoi que fasse l'environnement. En 2004, nous avons partiellement résolu ce problème, sous une hypothèse de divergence des coûts le long des exécutions [CI-162], et avons proposé un algorithme basé sur la synthèse de contrôleurs pour les systèmes hybrides, que nous avons codé grâce à l'outil HyTech [CI-159]. Nous avons aussi étudié les propriétés (en terme de mémoire et de représentation) des stratégies optimales ou presqu'optimales [CI-162]. L'équipe de Jean-François Raskin a ensuite prouvé en 2005 que, lorsque

l'on relâche l'hypothèse de divergence des coûts, le problème du calcul du coût optimal devient indécidable. Nous avons nous-même simplifié et amélioré la portée de cette preuve dans [RI-42]. Enfin, nous avons retrouvé des résultats de décidabilité dans le cas où le coût se réduit au temps qui s'écoule [CI-74], ainsi que dans le cas d'automates temporisés avec une seule horloge [CI-93]. Nous avons également étudié l'impact de l'ajout de fonctions de coûts sur les systèmes hybrides o-minimaux et montré la décidabilité du problème d'optimisation du coût d'accessibilité [CI-81].

A.2.2 Observation partielle

Dans le cas des systèmes ouverts, on ne peut plus supposer que tout ce que fait l'environnement peut être observé par le contrôleur. Tout se passe comme si le contrôleur voyait le système à travers une interface d'observabilité qui lui masquerait certaines actions. Il s'agit de l'hypothèse d'*observation partielle*. À la différence du cadre non temporisé où une telle hypothèse ne rajoute que peu de complexité au problème, nous avons montré que les problèmes de contrôle les plus simples (accessibilité, sûreté) deviennent indécidables dans le cadre temporisé [RE-2]. Nous avons poursuivi ce travail en nous intéressant à la détection d'erreurs dans les systèmes temporisés. Ce problème, que l'on rencontre dans le *runtime model-checking* ou la génération de tests, est *a priori* plus simple que le problème de contrôle sous observation partielle, car il consiste à observer un système temporisé et à détecter *en ligne* s'il y a eu une défaillance ou pas. Nous avons cependant montré des résultats d'équivalence des deux problèmes, dans le cadre les systèmes temporisés avec ressources fixées [Ra-39, CI-153]. Ces travaux ont fait l'objet d'un exposé invité [In-30] et d'un article de synthèse [In-23].

A.2.3 Logiques pour le contrôle

Les problèmes de contrôle simples imposent des objectifs tels que l'accessibilité ou l'évitabilité d'un ensemble d'états (voir paragraphe A.2.1). Il est possible d'imposer des objectifs de contrôle plus compliqués, en utilisant des raffinements des logiques temporelles, comme dans le cadre du model-checking. C'est dans cette optique que la logique ATL, une extension de CTL permettant de quantifier sur les stratégies des différents joueurs, a été définie en 1997 sur différents modèles de jeux concurrents non temporisés. Nous nous sommes intéressés à différentes extensions d'une telle logique de façon à augmenter son expressivité à la fois dans le cadre standard (non temporisé) et dans le cadre temporisé. Ces travaux ont fait l'objet d'un exposé invité [In-13].

Modèles non temporisés. Dans [CI-91], nous avons étudié l'expressivité et la complexité de la vérification de la logique ATL sur différents modèles de jeux concurrents. Notre étude a apporté plusieurs résultats surprenants, corrigeant en particulier un résultat de complexité erroné de l'article original définissant ATL. Dans [Ra-12], nous avons également défini et étudié une sémantique différente pour ATL, permettant notamment d'exprimer des propriétés d'équilibres de Nash (ce qu'ATL ne peut pas faire).

Modèles temporisés. Cette approche du contrôle utilisant des logiques temporelles a également été étendu au cadre temporisé. Nous avons défini et étudié des extensions de ATL dans [CI-109] dans le cas des jeux concurrents « simplement temporisés » (*cf.* section A.1.2), et dans [CI-67] pour les jeux concurrents avec horloges.

Dans [CI-114], nous avons utilisé la logique MTL pour exprimer les objectifs de contrôle comme des propriétés quantitatives sur les exécutions finies. Nous avons montré que le contrôle pour les objectifs MTL est indécidable, contrairement au problème du model-checking. Dans [CI-141], une transformation syntaxique d'un fragment du μ-calcul temporisé est proposée de façon à ramener

un problème de contrôle à un problème de model-checking, et fournir ainsi un algorithme pour le problème de contrôle.

A.3 Systèmes distribués

Avec le développement des réseaux et du calcul distribué (Internet, systèmes d'échanges de fichiers, réseaux de capteurs, ...), il est devenu fondamental de proposer des modèles pour les systèmes distribués et des logiques permettant de spécifier leurs bonnes propriétés, de façon à définir correctement leurs sémantiques et poser les fondements des bons outils d'analyse. Parmi les modèles reconnus figurent celui bien établi des réseaux de Petri ainsi que celui, plus récent, des diagrammes de séquences (*Message Sequence Charts*). Ce dernier modèle représente le comportement dynamique d'un nombre d'entités coopératives en définissant un ordre partiel sur les évènements de communication. Cette notion d'ordre partiel est très présente dans le calcul distribué du fait de l'absence fréquente de relation de causalité entre évènements affectant des entités distantes (indépendance). Elle permet de réduire considérablement l'ensemble des exécutions utiles à considérer. Par exemple, si une action c ne peut s'exécuter qu'après l'exécution d'une action a et d'une action b, mais indifféremment de leur ordre d'exécution (ce qui correspond à l'ordre partiel $\{a, b\} \prec c$), on ne s'intéresse qu'à l'exécution de la séquence d'actions a, b, c et ignore la séquence b, a, c. La théorie algébrique des exécutions exploitant cette notion d'ordre partiel a été développée sous le nom de *théorie des traces* par Mazurkiewicz dans les années 1970. L'ordre partiel des transitions a aussi été pris en compte en model-checking, par exemple dans le cadre des réseaux de Petri classiques, notamment pour obtenir des algorithmes de *dépliage* efficaces. De tels algorithmes consistent à déplier la structure d'un réseau de Petri, vu comme un graphe cyclique, en calculant le sous-ensemble représentatif des exécutions. Le dépliage naïf conduit à une structure infinie, mais, sous certaines conditions, McMillan a montré en 1993 qu'il était possible de construire une partie initiale finie contenant toute l'information utile du réseau de Petri (préfixe fini complet), en évitant d'explorer tous les entrelacements possibles des exécutions. Cette technique n'avait été jusqu'ici que peu exploitée dans le cadre des systèmes temporisés. Nous avons alors étudié les différentes extensions temporisées des réseaux de Petri, en les comparant notamment aux réseaux d'automates temporisés, ce qui nous a permis d'obtenir des méthodes de dépliage pour les automates temporisés eux-mêmes (paragraphe A.3.1). Nous avons également travaillé sur le modèle des diagrammes de séquences, et sur leurs extensions temporisées (paragraphe A.3.2). De nouvelles logiques temporelles sur les ordres partiels, adaptées à l'étude des systèmes ouverts et distribués, ont par ailleurs été développées (paragraphe A.3.3), ainsi que des méthodes pour résoudre des problèmes de contrôle dans le cadre distribué (paragraphe A.3.4). Enfin de nouvelles méthodes d'analyse quantitative de systèmes distribués probabilistes (qui sont les seuls à pouvoir résoudre certains problèmes algorithmiques dans un cadre distribué) ont été proposées (paragraphe A.3.5).

A.3.1 Réseaux de Petri temporisés

Dans le but de mieux comprendre comment les aspects distribués peuvent être pris en compte dans les systèmes temporisés, nous avons étudié diverses extensions de réseaux de Petri intégrant des contraintes temporisées, avec l'idée de développer des méthodes de dépliage et de réduction par ordre partiel pour ces extensions aux (réseaux d') automates temporisés.

Les réseaux de Petri temporisés, un modèle temporisé *et* distribué. Les premiers travaux que nous avons développés comparent des modèles de réseaux de Petri temporisés avec

celui des réseaux d'automates temporisés (qui est implanté dans Uppaal). De multiples extensions temporisées des réseaux de Petri classiques ont été proposées : elles diffèrent par la nature de la temporisation (un âge peut être attaché à chaque jeton du réseau, ou bien une horloge peut être associée à chaque transition sensibilisée du réseau, *etc.*) et par l'interprétation faible ou forte des contraintes temporisées (par exemple, sous une sémantique forte, une transition doit être prise avant de dépasser sa date d'expiration, alors que sous une sémantique faible, un jeton peut mourir et ne plus jamais servir à franchir des transitions). Dans [CI-124], nous avons proposé une procédure polynomiale de traduction des réseaux d'automates temporisés dans les réseaux de Petri temporels bornés à sémantique forte. Nous avons également considéré un modèle à sémantique faible, dans lequel le choix de laisser le temps s'écouler est réellement local, et qui semble mieux adapté à la traduction vers le cadre des automates temporisés. Dans [CI-120, RI-18], nous avons alors étudié précisément les liens entre les réseaux de Petri temporisés avec arcs de lecture à sémantique faible : contrairement à l'intuition et à ce qui était affirmé dans la littérature, les réseaux de Petri temporisés ne sont pas strictement plus expressifs que les automates temporisés, mais sont en fait incomparables.

Dépliages de réseaux d'automates temporisés. Dans le cadre temporisé, l'ordre partiel sur les évènements ne correspond pas uniquement à une relation de causalité, mais aussi à une précédence dans le temps. Ceci complique l'application des méthodes de réduction associées. Cependant, en nous appuyant sur nos travaux sur les réseaux de Petri temporisés, nous avons proposé un algorithme de dépliage et de préfixe complet fini pour les réseaux d'automates temporisés avec invariants, basé sur les réseaux de Petri temporisés avec arcs de lecture, ce qui débouche sur des méthodes de réduction pour les (réseaux d')automates temporisés [CI-103].

A.3.2 Diagrammes de séquences

Une partie de nos travaux s'intéresse au développement de systèmes distribués, depuis leur spécification jusqu'à leur implantation, dans le formalisme des « diagrammes de séquences », aussi connus sous le nom de *Message Sequence Charts*. Il s'agit d'un modèle visuel largement utilisé pour spécifier des protocoles.

Analyse et synthèse. Nous nous sommes intéressés à la synthèse de systèmes distribués à partir d'une spécification donnée sous la forme d'un diagramme de séquences. Les systèmes distribués sont synthétisés sous forme d'automates communicants. entre les automates et le formalisme de la logique monadique du second ordre (MSO). Nous étudions plus particulièrement des classes d'automates qui décrivent des systèmes distribués. Nous donnons également une théorie des machines communicantes et de leurs propriétés logiques, en représentant ces machines sous forme d'automates finis reliés par des canaux non-fiables. Les exécutions de tels systèmes sont décrites en utilisant des notions de graphes et d'ordres partiels, liés aux traces de Mazurkiewicz, de diagrammes de séquences et de *live sequence charts*. Nous avons construit un outil, MSCan, qui analyse les caractéristiques de tels diagrammes de séquence [CI-128]. Cet outil analyse une spécification et recherche des sources d'erreurs potentielles. Parmi ses fonctions variées, il aide l'utilisateur à éditer, visualiser et modifier des diagrammes de séquences[2]. Nous avons également considéré des automates distribués modélisant des systèmes réactifs ne s'arrêtant jamais, comme par exemple des appareils médicaux. Nous avons défini une logique qui sert de langage de spécification et avons montré comment traduire une formule de notre logique en un automate distribué [Ra-28, CO-13, RI-9].

[2]Plus de détails sont disponibles sur : `http://www-i2.informatik.rwth-aachen.de/MSCan/`

Extensions temporisées. Ce projet combine le modèle de diagrammes de séquences et celui d'automates temporisé. Nous considérons des techniques de spécification et vérification correspondantes. Dans [Ra-22, CI-50], nous définissons la notion d'automates distribués et temporisés permettant d'introduire le temps dans la représentation d'un système distribué. Les processus sont enrichis avec des contraintes de temps pour mesurer le temps écoulé depuis certains événements. De ce fait, le comportement d'un tel automate peut être décrit par un ensemble de diagrammes de séquences où chaque événement est associé à la date à laquelle l'action correspondante a été effectuée. Nous prouvons l'équivalence de notre modèle et la logique MSO, un langage de spécification qui permet également d'exprimer des contraintes de temps. La vérification de systèmes distribués et temporisés est le sujet de l'article [CI-65]. Nous y proposons une procédure pour décider si chaque comportement d'un automate existe aussi dans la spécification, décrite par un diagramme de séquence.

Apprentissage. L'approche que nous suivons dans ce projet est basée sur des algorithmes d'apprentissage qui permettent un développement de systèmes interactifs. L'objectif principal est d'obtenir une méthodologie flexible, efficace et compréhensible pour le développement de systèmes informatiques, qui est basée sur l'idée d'apprentissage par des diagrammes des séquences (Smyle Modeling Approach). Cet objectif ambitieux a besoin d'une base solide de recherche fondamentale, demandant une extension de la théorie existante, focalisée sur les systèmes séquentiels, aux systèmes distribués. Nous accompagnons le développement de la théorie [CI-86] du développement de l'outil Smyle.[3] Les premiers résultats sont prometteurs : Smyle a déjà permis l'apprentissage de petits systèmes.

Test distribué. Quels que soient les efforts consentis à la vérification des systèmes informatiques, la phase de test du système réel reste utile. Lorsque le système est concurrent ou que la communication avec le système est asynchrone, il n'est pas possible d'observer complètement ou immédiatement le système. Le test d'un tel système est donc beaucoup plus complexe. Nous avons étudié le test local de systèmes modélisés par des diagrammes de séquences [CI-70]. L'idée est de substituer un testeur à l'un des processus du système et d'observer localement les interactions avec le reste du système. Nous établissons des résultats de décidabilité et d'indécidabilité du test local par des sous-ensembles de processus. Nous avons aussi étudié différentes techniques de test pour les systèmes communiquant de façon asynchrone par files d'attentes [CI-102]. Nous établissons leur pouvoir d'expression en terme d'équivalence de systèmes soumis à ces tests. Nous prouvons des résultats sur la décidabilité de nos techniques de test ainsi que d'autres techniques introduites auparavant.

A.3.3 Logiques temporelles

Le développement de systèmes critiques, et en particulier leur vérification, nécessite des spécifications formelles. Parmi les différents formalismes, les logiques temporelles sont très utilisées car elles permettent d'exprimer facilement les propriétés classiques des systèmes et possèdent de bonnes propriétés algorithmiques. L'étude des systèmes distribués nécessite des logiques adaptées qui reposent sur la relation de causalité entre les événements du système et ne dépendent pas des possibles observations séquentielles. Dans ce contexte, il est naturel de considérer des logiques temporelles sur les ordres partiels.

[3] Une version provisoire est disponible sur : `http://www.smyle-tool.org/`

Expressivité des logiques temporelles. Nous nous sommes d'abord intéressés à l'expressivité des logiques temporelles pour les traces de Mazurkiewicz. Un de nos résultats majeurs a été d'établir que les logiques temporelles *locales* ont le même pouvoir d'expression que la logique du premier ordre [RI-68, RI-36]. Ce problème difficile est resté ouvert pendant plusieurs années. Quatre ans plus tôt, nous avions obtenu un résultat similaire pour les logiques temporelles *globales*. Nous avons ensuite prouvé que l'expressivité des logiques globales pouvait se déduire de celle des logiques locales [RI-44]. Nous avions aussi obtenu auparavant plusieurs autres résultats comme par exemple des caractérisations des propriétés de sûreté et de vivacité pour les systèmes concurrents. Un cours de synthèse sur la spécification des systèmes distribués a été présenté à l'école de printemps d'informatique théorique en 2004 [In-39].

Décidabilité et complexité. Il est naturel d'étudier la décidabilité et la complexité des langages de spécification. Nous avons développé des algorithmes généraux pour résoudre les problèmes de satisfaisabilité et de model-checking pour les logiques temporelles sur les ordres partiels. Jusqu'alors, l'introduction de chaque nouvelle logique (et il y en a eu beaucoup) était accompagnée de procédures de décision pour les problèmes ci-dessus. Nous avons introduit un cadre général (logiques temporelles dont les opérateurs sont définissables en logique monadique du second ordre) et prouvé que toutes ces logiques sont décidables en espace polynomial par rapport à la taille de la formule, ce qui est le mieux qu'on puisse espérer car les logiques les plus simples sont déjà PSPACE-complètes [CI-143]. Si l'architecture du système fait partie de la donnée, nous avons prouvé que la complexité est en général beaucoup plus élevée et dépend du nombre d'alternances des quantificateurs. Finalement, nous avons montré que pour une large classe de logiques temporelles, qui comprend toutes celles définies jusqu'à présent, la satisfaisabilité reste PSPACE même lorsque l'architecture fait partie de la donnée [RI-21]. Nous avons aussi considéré la logique dynamique propositionnelle pour les diagrammes de séquences. Nous avons montré comment traduire effectivement une formule de cette logique en automate communicant, même lorsque les canaux ne sont pas bornés. Nous avons ensuite considéré le problème du model-checking pour les automates communicants et cette logique dynamique, montrant qu'on pouvait le résoudre en espace polynomial lorsqu'on se restreint aux exécutions *existentiellement bornées*, c'est-à-dire qui admettent au moins une linéarisation utilisant des canaux bornés [CI-52].

A.3.4 Contrôle distribué

Le développement des systèmes ouverts fait souvent appel à des composants (comme les automates programmables en automatique) qu'il faut adapter aux applications particulières. Or aujourd'hui, les systèmes embarqués sont souvent composés de plusieurs systèmes qui communiquent entre eux et coopèrent pour réaliser certaines fonctionnalités. Dès lors, chaque composant doit être supervisé localement afin d'accroître la réactivité et la robustesse du système global. Les problèmes de contrôle et de synthèse deviennent donc *distribués*. Une difficulté supplémentaire vient du fait que chaque superviseur n'a qu'une vision partielle du système global. Comme dans le cas non distribué, le problème de la synthèse se ramène à l'étude de stratégie gagnante dans un jeu. Ici cependant, chaque processeur correspond à un joueur de l'équipe du contrôleur, et l'environnement à l'autre équipe. Pnueli et Rosner ont exploré en 1990 le cadre synchrone : ils utilisent la notion d'*architecture* qui consiste en un ensemble de sites reliés par des canaux qui permettent de communiquer de façon synchrone. Dans ce cadre, le jeu est alterné : l'équipe du contrôleur et l'environnement jouent à tour de rôle. Dans le cadre asynchrone, la communication entre sites est asynchrone et utilise de la mémoire partagée. Le jeu est alors asynchrone : plusieurs joueurs des deux équipes peuvent jouer simultanément.

Systèmes synchrones. Dans le cadre synchrone, l'observation partielle (voir paragraphe A.2.2) conduit très vite à des problèmes indécidables ou à des complexités très élevées pour les architectures décidables. Nous avons obtenu des avancées importantes dans ce cadre en établissant un critère de décidabilité pour le problème de synthèse sur les architectures uniformément bien connectées [Ra-32, CI-97]. Nous avons aussi montré que ces architectures sont décidables lorsque les spécifications sont *robustes*, c'esst-à-dire lorsque la spécification ne lie une sortie qu'aux entrées auxquelles elle est connectée.

Systèmes asynchrones. Nous avons également étudié le problème du contrôle dans le cadre asynchrone [CI-163]. Notre contribution majeure est l'introduction de stratégies causales pour les superviseurs. Auparavant, les stratégies étaient toujours locales. En permettant aux contrôleurs de se transmettre un peu d'information, nous avons enfin obtenu des résultats de décidabilité du contrôle, au moins pour les systèmes séries-parallèles, le cas général étant encore ouvert. Nos travaux ont aussi donné lieu à deux exposés invités dans des workshops internationaux [In-20, In-16].

A.3.5 Systèmes distribués probabilistes

Auto-stabilisation probabiliste. Dans le domaine des systèmes distribués *uniformes*, c'est-à-dire de réseaux de processeurs identiques communiquant de façon locale avec leurs voisins, il est bien connu que, pour un certain nombre de problèmes généraux, il ne peut exister de solution déterministe : problème de l'allocation de ressource (« comment s'assurer qu'une certaine ressource est bien parvenue à tous les processeurs du réseau ? »), problème du consensus (« comment garantir que tous les ordinateurs prennent une même décision simultanément ? »), problème de l'élection de leader (« comment assurer à un processeur qu'il est seul à être dans un état distingué ? »), ... Les algorithmes résolvant de tels problèmes sont donc nécessairement probabilistes et font intervenir le hasard lorsqu'un processeur passe d'un état à l'autre. Nous avons été amenés à étudier principalement la question de la *convergence avec probabilité* 1 ou *auto-stabilisation*. Cette propriété assure que, quel que soit l'état de départ et quel que soit l'enchaînement des actions partant de cet état, le système atteindra toujours (avec probabilité 1) un ensemble donné d'arrivée en un nombre fini d'actions. Un exemple bien connu est le problème du dîner des philosophes, dans lequel le but est d'assurer qu'un philosophe finira toujours par manger.

Nous avons proposé une nouvelle méthode de preuve de convergence en dégageant des critères garantissant la convergence, et appliqué différentes techniques d'abstraction (par exemple le *lumping*, aussi connu sous le nom de bisimilarité forte) et de calcul pour analyser le temps moyen de convergence, fondés sur la théorie des chaînes de Markov et les processus de décision markoviens (chaînes de Markov intégrant des choix non-déterministes). Cette méthode nous a notamment permis de lever l'hypothèse d'*équité* (requérant la participation de chaque processeur à tout calcul suffisamment long), sur laquelle reposaient jusqu'ici toutes les preuves de correction de l'algorithme des philosophes [RI-75].

Vitesse d'auto-stabilisation. Comme indiqué précédemment, beaucoup de problèmes d'algorithmique distribuée uniforme n'ont pas de solution déterministe. L'introduction de tirage aléatoires dans les algorithmes concurrents permet de réduire la complexité de certains problèmes difficiles, et d'en résoudre d'autres insolubles dans le cadre déterministe.

Afin de démontrer la convergence de tels algorithmes vers des états stables, nous avons exploité la technique de *coupling* utilisée dans la théorie des chaînes de Markov pour démontrer la *stabilité* (au sens de l'unicité de la distribution stationnaire). Cette technique consiste à trouver une distance entre deux copies « fidèles » de la chaîne de Markov étudiée partant de deux

états arbitrairement éloignés, et à montrer que cette distance diminue en moyenne au bout de l'exécution d'un certain nombre de pas de l'algorithme [TH-17]. Un avantage de cette technique est qu'elle fournit en outre un majorant du temps moyen de la vitesse de stabilisation de la chaîne originale [CI-166, CO-24]. La méthode a permis notamment d'améliorer la meilleure borne connue sur la vitesse de convergence vers un état d'équilibre pour les topologies circulaires dans le jeu itéré du dilemme du prisonnier [RI-50].

A.4 Sujets de recherche connexes

A.4.1 Aspects quantitatifs

Comme mentionné précédemment, les systèmes réels doivent vérifier de multiples contraintes, que ce soient des contraintes temporisées, ou bien des contraintes de coûts, et ils sont aussi parfois sujets à des évolutions probabilistes. Dans les paragraphes A.1.2 et A.2.1, nous avons présenté nos travaux sur le modèle des automates temporisés à coûts, qui permet d'exprimer des systèmes ayant à la fois des contraintes temporisées et des contraintes de coûts. Nous allons ici abstraire ces différents aspects quantitatifs et les modéliser en associant à chaque comportement du système une valeur (un poids) dans un semi-anneau. Le système peut alors être décrit par un automate à poids (ou multiplicités) ou par une série formelle rationnelle. L'équivalence entre ces deux modèles est due à Schützenberger et remonte au début des années 1960. Elle a donné lieu au développement d'une très riche théorie.

Nous nous sommes intéressés aux propriétés quantitatives des systèmes concurrents. Nous avons généralisé aux traces de Mazurkiewicz des résultats majeurs établis il y a plus de 40 ans pour les systèmes séquentiels, en particulier l'équivalence entre automates à poids et séries formelles rationnelles ou l'équivalence entre apériodique et sans étoile pour les séries formelles [RI-13].

Nous nous sommes aussi intéressés à la définition de logiques pour spécifier les propriétés quantitatives des systèmes. Même pour les systèmes séquentiels, ce domaine n'avait jamais été abordé auparavant. Nous avons introduit la logique monadique du second ordre pondérée et nous avons prouvé qu'elle a le même pouvoir d'expression que les automates pondérés [CI-146, In-18, RI-25]. Nous avons aussi considéré la restriction au premier ordre et montré qu'elle coïncide avec l'apériodicité. Ces résultats ont été à l'origine de nombreux développements et ont été généralisés par plusieurs chercheurs à des systèmes infinis, ou concurrents, ou encore à des images pondérées.

Dans [CI-80], nous fournissons une caractérisation logique d'une classe d'automates qui permettent de modéliser des systèmes quantitatifs. Des systèmes quantitatifs sont représentés par des automates cellulaires asynchrones pondérés, avec des poids sur les transitions. En particulier, ces automates sont capables de modéliser un comportement probabiliste. Nous prouvons qu'un tel système peut être décrit par une formule existentielle de MSO (avec des poids) et, inversement, que n'importe quelle formule correspond à un automate pondéré.

A.4.2 Logiques temporelles

Les logiques temporelles ont toujours été au cœur de la plupart de nos travaux de recherche. Nous les avons déjà mentionnées à plusieurs reprises dans les parties précédentes, mais nous les avons aussi étudiées en-dehors du cadre de la vérification quantitative.

Expressivité et complexité des logiques temporelles. Les problèmes d'expressivité et de complexité sont au cœur de la problématique du model-checking. Nous avons évidemment continué ce type de travaux et avons par exemple calculé la complexité exacte du model-checking de différents fragments de la logique LTL avec passé [RI-73], montrant que, de manière générale,

l'ajout de modalités du passé dans un fragment n'augmente pas la complexité, alors qu'il augmente l'expressivité. Nous avons aussi montré que la logique ECTL^+ (qui étend CTL avec des propriétés d'équité) a la même expressivité que la logique BTL_2, une logique du temps arborescent dont les formules de chemins sont des formules du premier ordre de hauteur d'alternation au plus 2 [RI-40].

L'étude de la complexité des problèmes de model-checking n'est parfois pas suffisante pour comprendre d'où viennent les problèmes d'explosion combinatoire. Dans [RI-43], nous étudions la complexité paramétrée du model-checking de systèmes non-plats, c'est-à-dire définis comme des produits de systèmes. La vérification de tels systèmes est généralement très difficile, du fait que les produits de systèmes sont en fait une façon concise de coder des systèmes exponentiellement plus grands. Nous analysons la complexité paramétrée de ce problème, prenant comme paramètre le nombre de systèmes impliqués dans le produit.

Model-checking de chemins. Outre le cadre temporisé mentionné à la section A.1.3, nous avons également étudié le model-checking de chemins dans le cadre classique. D'une part, nous avons considéré le problème pour le μ-calcul, montrant que, contrairement au cas des logiques temporelles classiques, la restriction aux systèmes « déterministes » ne permet pas d'améliorer la complexité du model-checking [RI-49].

Nous avons également étudié ce problème dans une optique particulière : est-il encore possible de vérifier des propriétés efficacement sur un chemin si celui-ci est donné de façon compressée (comme ce peut être le cas, par exemple, pour l'analyse de logs). Nous avons montré que le problème de l'acceptation d'un mot compressé par un automate est PTIME-complet [RI-76].

Contre-exemples en SPIN. Nous avons travaillé sur la génération de contre-exemples minimaux lorsque qu'un système ne satisfait pas sa spécification. Il faut dire que les outils de vérification sont très souvent utilisés pour trouver des erreurs et que le « retour » fourni à l'utilisateur se doit d'être facilement exploitable. SPIN ne permet pas de trouver un contre-exemple minimal en taille, alors que c'est essentiel pour faciliter l'analyse du contre-exemple. Nous avons donc développé un algorithme pour générer des contre-exemples minimaux [CI-76]. Il fonctionne en espace linéaire (il n'utilise pas plus d'espace que SPIN) et en temps quadratique. La contrainte en espace est vraiment essentielle (un entier et un octet par état) car c'est actuellement la principale limitation pour vérifier des systèmes de grande taille.

Extension de $\mathsf{CTL}^\star$ au temps continu. Dans [CI-183], nous étendons la logique $\mathsf{CTL}^\star$ pour exprimer des propriétés arborescentes sur des systèmes observés en temps continu, comme les systèmes hybrides ou des systèmes gouvernés par des inclusions différentielles. Nous définissons $\mathsf{CTL}^\star$ dans ce contexte, puis proposons une axiomatisation correcte pour cette logique, à partir de l'axiomatisation de $\mathsf{CTL}^\star$ en temps discret.

Logiques sur les ordinaux. Le théorème de Kamp (1968) énonce que la logique temporelle du temps linéaire (avec passé) LTL a la même expressivité que la logique du premier ordre pour toute classe d'ordres linéaires Dedekind-complets. Lorsque ces ordres sont des ordinaux infinis autres que ω, LTL peut être vue comme un langage de spécification pour exprimer des comportements « de Zénon ». Dans [CI-138], nous avons introduit une classe de logiques LTL (α) indicées par un ordinal α (fermé par addition) dont les modèles sont de longueur α. Nous avons montré que LTL (ω^k) avec $k \geq 1$ a des problèmes de satisfaisabilité et de model-checking PSPACE-complets lorsque les entiers dans les formules sont codés en unaire et $\mathsf{EXPSPACE}$-complets lorsqu'ils sont codés en binaire [CI-138] en introduisant une classe d'automates avec transitions limites. Ce travail a été poursuivi en montrant que la logique du temps linéaire avec les opérateurs stricts « Since » et

« Until » interprétée sur les ordinaux a un problème de satisfaisabilité PSPACE-complet [CI-63], ce qui nous a aussi permis de retrouver facilement de nombreux résultats de complexité établis dans [CI-138].

Les langages définissables au premier ordre. La logique du premier ordre constitue un langage de spécification extrêmement puissant. Elle a été extensivement étudiée depuis la fin des années 1960. Nous avons rédigé un *survey* des principaux résultats sur les langages de mots finis ou infinis définissables en logique du premier ordre [Ch-1]. L'objectif était de donner des preuves complètes et les plus simples possibles des principales caractérisations de cette famille de langages. Nous prouvons ainsi l'équivalence entre langages sans étoile, langages apériodiques, langages définissables en logique de premier ordre, langages définissables en logique temporelle du temps linéaire, langages reconnaissables par des automates de Büchi sans compteur ou apériodiques, et langages définissables par des automates alternants très faibles.

Développement d'un outil pour LTL. Nous nous sommes attaqués à la génération d'automates correspondant à des formules de LTL. Il s'agit d'une étape essentielle de la vérification automatique. L'algorithme de traduction utilisé dans SPIN échouait sur des formules de taille moyenne pourtant très naturelles pour la spécification des systèmes. Nous avons développé un nouvel algorithme beaucoup plus performant (le temps de calcul pouvant être réduit de plusieurs heures à quelques centièmes de secondes) [Lo-3]. Notre algorithme fait maintenant partie de la distribution de SPIN et il est directement utilisable en ligne à partir de l'adresse `http://www.lsv.ens-cachan.fr/~gastin/ltl2ba/index.php`.

A.4.3 Logiques modales

Changements de politiques d'action. Les logiques dites déontiques sont communément définies comme les logiques de l'obligation, de l'interdiction et de la permission. En effet, pouvoir comparer des comportements idéaux à des comportements observés est souvent utile, par exemple pour spécifier les comportements souhaités d'utilisateurs ou pour définir des politiques de sécurité. Parmi ces logiques déontiques, la logique dynamique de la permission (DLP) introduite en 1996 est définie comme une extension de la logique dynamique propositionnelle PDL (sans opérateur de test) mais munie d'opérateurs modaux supplémentaires qui prennent en compte la permission de certaines transitions. Dans un modèle de DLP, l'ensemble des transitions permises forme une politique d'actions. Il a été montré que le problème de la satisfaisabilité pour DLP est dans NEXPTIME et cela en étendant les techniques standard pour PDL. En 2004, DLP a été étendue de façon à pouvoir mettre à jour l'ensemble de transitions permises ce qui revient à modifier dynamiquement la politique d'actions selon l'état courant du système. La possibilité d'ajouter ou de détruire des transitions dans le modèle est similaire à ce qui se passe dans la logique modale du sabotage dont une variante a été montrée indécidable. Cette logique a été introduite pour spécifier des propriétés de réseaux de machines dont certaines machines ou connections peuvent disparaître de façon non contrôlée.

Dans [RI-59] une nouvelle extension de DLP_{dyn} a été définie, appelée ici DLP^{+}_{dyn}, pour laquelle une traduction vers PDL a été proposée. Dans DLP^{+}_{dyn}, nous autorisons l'opérateur de test et des opérateurs logiques qui mettent à jour la politique en fonction de la politique courante, une nouveauté majeure par rapport aux versions précédentes. En dépit de ces extensions substantielles, une traduction exponentielle de DLP^{+}_{dyn} vers PDL permet d'établir que DLP^{+}_{dyn} reste décidable en 2EXPTIME et que DLP est EXPTIME-complète, résolvant un problème ouvert depuis plus de 10 ans. Il a depuis été montré que DLP^{+}_{dyn} est aussi dans EXPTIME en raffinant notre traduction initiale.

De la complexité des logiques grammaticales régulières. Dans les années 1980, la mécanisation du raisonnement conduit à privilégier d'autres critères pour le choix de logiques : les questions de complexité algorithmique et de concision s'ajoutent aux questions plus classiques de complétude ou d'axiomatisation. La PSPACE-complétude des logiques K et S4, établie en 1977, initie la recherche sur le thème de la complexité algorithmique des logiques modales. Nous avons montré [RI-65] comment traduire simplement toutes les logiques modales régulières avec passé vers le fragment gardé de la logique classique avec deux variables, GF2. Nous avons ainsi obtenu de nouveaux résultats de complexité pour ces logiques (une borne supérieure est EXPTIME) et surtout on peut éviter l'usage de l'extension de la logique classique à base d'opérateur de point fixe : le fragment gardé de la logique avec deux variables suffit. C'est un résultat important car aujourd'hui il permet d'obtenir des procédures de décision pour les logiques modales régulières avec passé avec un simple démonstrateur pour le fragment gardé de la logique classique restreint à deux variables.

A.5 Applications

A.5.1 Systèmes intégrés sur puces

Vérification temporisée. Les puces électroniques incluent aujourd'hui beaucoup plus de fonctionnalités que les composants d'hier connectés entre eux : les constituants d'un système complet (processeur, périphériques et mémoires) sont souvent mis sur un seul composant, appelé *System On Chip* (SOC). L'amélioration de ces capacités induit une explosion de la taille mémoire qui prend de plus en plus de place sur le composant de silicium. Le temps mis pour lire et écrire des données sur les mémoires a un impact important sur les performances des SOC. Pour atteindre les performances requises, ces composants critiques sont directement conçus au niveau transistor (et non pas à partir de bibliothèques de composants standard). À ce niveau de conception, peu d'outils d'analyse fonctionnelle et temporelle sont disponibles. Les techniques de simulation actuellement utilisées pour vérifier ces délais ne sont pas exhaustives et risquent de laisser des cas problématiques, dont la réparation est très coûteuse, lors de leur détection après fabrication. En pratique, les concepteurs sont amenés à sélectionner empiriquement les parties du circuit à analyser. Dans les projets MEDEA+ Blueberries puis ANR VALMEM, nous avons exploré l'utilisation de méthodes formelles pour vérifier les temps de réponse de telles mémoires. L'idée est de représenter les éléments constitutifs de la mémoire (portes, registres, ...) ainsi que les signaux d'entrée sous forme d'automate temporisés. Les délais intervenant dans ces automates concernent les temps de propagation des fronts au travers des portes et registres, ainsi que les durées intervenant dans les signaux d'entrée comme leurs temps de stabilisation. En composant les automates temporisés entre eux, on obtient un système qui modélise la mémoire, et on peut calculer formellement les délais d'écriture ou de lecture en mémoire en fonction des délais élémentaires. Dans le projet BLUEBERRIES, l'analyse exploite des valeurs de délais élémentaires par portes logiques, qui restent calculés par simulation électrique manuelle [RE-1]. Les outils de model-checking utilisés ont été Uppaal et HyTech. L'avantage d'utiliser HyTech est de permettre la synthèse de contraintes génériques qui donnent des conditions générales assurant la correction temporisé de la mémoire, et montrer par exemple, que le temps de réponse globale est inférieur au temps maximal de la spécification lorsque les temps de stabilisation satisfont certaines conditions. Cette formulation permet de résoudre certains problèmes d'*optimisation* analogues à ceux mentionnés au paragraphe A.2, comme trouver la valeur minimale des temps de stabilisation garantissant un temps de lecture inférieure au temps maximal spécifié. Pour inférer les contraintes paramétrées, nous avons proposé une méthode de raffinement de contraintes lors de la détection de contre-exemples [CI-108]. La méthode a permis la vérification des caractéristiques

temporelles de plusieurs implémentations d'une même architecture mémoire (SPSMALL) fournie par ST-Microelectronics.

Cette démarche comprenait plusieurs étapes manuelles : la décomposition de la liste de transistors en unités fonctionnelles, la sélection d'une partie du circuit à analyser, supposée critique le choix des contraintes temporelles à raffiner, ... L'idée du projet ANR VALMEM est d'automatiser ce traitement en utilisant des outils permettant de générer automatiquement les descriptions de blocs fonctionnels de portes et de registres avec les délais de traversée associés, d'une part, et de transformer automatiquement cette description sous forme d'automates temporisés. De plus, nous avons défini une méthode efficace et automatisable pour synthétiser les contraintes garantissant un mode de bon fonctionnement à partir d'une instanciation des paramètres du modèle [CI-41]. Notre objectif est d'automatiser la vérification de la mémoire SPSMALL ainsi que d'autres composants mémoire plus complexes.

Vérification d'abstraction. À un niveau plus abstrait, nous avons également étudié un modèle à base d'automates à entrées/sorties, rendant compte du traitement global réalisé à chaque cycle. Ce modèle décrit les flux d'entrés/sorties des composants du SOC, en faisant abstraction des étiquettes et des données. Ce formalisme est utile pour s'assurer de la correction d'une composition d'automates concrets vis-à-vis d'un automate abstrait. L'idée est d'utiliser cette abstraction afin de faciliter les tests fonctionnels de validation du circuit. la correction de l'abstraction repose sur l'égalité des langages reconnus par les deux modèles. Ce travail a nécessité la définition d'un nouveau produit d'automates [TH-12]. La méthode a été implémentée efficacement en utilisant un calcul d'automate minimal, développé dans [CI-112] des caractéristiques proches des outils de manipulation des diagrammes de décision binaire. Cette démarche a mis en évidence une erreur dans l'abstraction d'un composant utilisé en traitement du signal (Inverse Discrete Cosine Transform) [Ra-46].

De façon analogue, nous nous intéressons à la conception incrémentale des composants et à la vérification de propriétés globales, portant sur la combinaison de plusieurs composants [CI-125]. Nous proposons, ici aussi, d'utiliser la spécification de chaque composant pour lui associer un composant abstrait, et de remplacer, dans le processus de vérification de la propriété globale, chaque composant concret par le composant abstrait qui le représente. Dans [CI-73], nous définissons un algorithme permettant de construire une abstraction conservative d'un composant à partir de sa spécification décrite en CTL. Nous montrons également les gains en performances pour la vérification d'une plate-forme réalisant la conversion entre les protocoles VCI et PI, composée de plusieurs initiateurs de transactions.

Vérification d'interblocage. L'intégration croissante de composants sur un circuit a amené à repenser l'architecture globale du circuit : les architectures à base de bus sont remplacés par des réseaux multi-étages. La conception de ces réseaux nécessite la définition de la topologie du réseau ainsi que de sa fonction de routage, décrivant les voies d'acheminement des messages dans la topologie. Un problème important consiste à définir, pour une topologie donnée, des fonctions de routage qui soient *exemptes d'interblocage*. Nous avons étendus les travaux développés par Duato dès 1995 pour prendre en compte les dépendances entre messages apparaisant dans nos réseaux. Nous avons en particulier proposé une condition suffisante garantissant que tous les messages d'un même type peuvent être évacués, ainsi qu'un algorithme de résolution original, basé sur la réduction des composants fortement connexes du graphe de dépendances. Ces modèles et algorithmes ont été implantés dans l'outil ODI, et expérimentés sur différents réseaux d'interconnexions intégrés sur puce [CI-100]. La comparaison des performances des différents algorithmes est décrite dans [RI-6]. L'outil ODI a été utilisé avec succès pour caractériser des configurations de réseau avec routeurs défaillants exemptes d'interblocages.

Ces travaux sont synthétisés dans [TH-4].

A.5.2 Autres études de cas

Dans le cadre de plusieurs de nos projets, nous avons été amenés à appliquer les techniques de model-checking à des systèmes réels, comme des protocoles de communication et des automates programmables industriels. Dans chacun des cas, nous avons utilisé les outils de model-checking existants qui nous semblaient les plus adaptés.

Protocole PGM. Dans le cadre du projet Calife, nous avons étudié le protocole de transport multi-point PGM [RI-71], qui permet de transmettre des données sur un réseau en évitant trop de redondances. Nous avons modélisé ce protocole (pour une petite architecture) par un réseau d'automates temporisés et avons utilisé l'outil Uppaal pour vérifier des propriétés telles que « tout récepteur se rend compte si des paquets envoyés ont été perdus ». Nous avons montré que cette propriété était bien vérifiée par le protocole, à partir du moment où le réseau vérifiait certaines contraintes sur les délais de transmission.

Automates programmables industriels. Dans le cadre du plan pluri-formations VSMT (avec le laboratoire d'automatique de l'ENS Cachan, le LURPA), nous avons été amenés à valider nos techniques de model-checking dans le cadre des automates programmables industriels (et des diagrammes Ladder), très utilisés en automatique. Nous avons commencé par modéliser un évaporateur en utilisant les automates temporisés et l'outil Uppaal [CO-35] et avons ensuite modélisé une partie d'une plateforme MSS (*Mecatronic Standard System*) du groupe Bosch, toujours avec des réseaux d'automates temporisés et l'outil Uppaal [CI-139].

Protocoles CSMA/CD et CSMA/CA. Dans le cadre du projet Averroes, nous avons été amenés à modéliser et analyser des protocoles de télécommunication, comme CSMA/CD [Rc-37, Rc-34, Rc-35], utilisé sur les réseaux Ethernet, et CSMA/CA [Rc-21], utilisés pour les communications Wi-Fi. Ces protocoles font intervenir des algorithmes ayant une composante aléatoire (« randomisée ») dans le but de casser des situations d'interblocage apparaissant, par exemple, lorsque deux émetteurs souhaitent transmettre à nouveau, le plus rapidement possible, des messages entrés une première fois en collision. La modélisation et l'étude de tels protocoles à l'aide de différents outils de model-checking probabilistes (APMC, PRISM) nous a amenés à proposer une méthodologie d'analyse combinée exploitant leur complémentarité, qui nous a permis d'obtenir des résultats quantitatifs de plus grande échelle, en cohérence avec ceux observés par simulation [CI-151]. Nous avons par ailleurs implémenté des techniques d'agglomération des états entre eux, en respectant un principe de bisimulation approchée [Lo-8], visant à contourner le problème de l'explosion combinatoire du nombre d'états apparaissant rapidement dans les systèmes probabilistes intégrant du non-déterminisme (processus de décision markoviens).

A.6 Contrats

L'axe TEMPO est impliqué dans de nombreux contrats, dont nous donnons un bref descriptif ci-dessous.

ARTIST2. Il s'agit d'un réseau d'excellence du programme FP6 de l'union européenne (sept. 2004 – sept.2008 : `http://www.artist-embedded.org/artist/`) sur la modélisation et la vérification des systèmes embarqués [OC-1], dirigé par Joseph Sifakis (VERIMAG). Nous participons au

cluster « Quantitative Testing and Verification », qui implique des équipes européennes comme l'université d'Aalborg (Danemark), l'université de Twente (Pays-Bas), l'IRISA (Rennes). Ce réseau a aussi une vocation pédagogique [RI-64] et nous avons participé dans ce cadre à l'école d'été associée en septembre 2005 [In-24].

AVERROES. Le projet RNTL AVERROES « Vérification de propriétés quantitatives et fonctionnelles » (oct. 2002 – mars 2006 : `http://www-verimag.imag.fr/AVERROES/`) a eu pour objectif le développement de méthodes formelles capables de vérifier, de manière fiable, des propriétés multiformes, à la fois fonctionnelles et non fonctionnelles (quantitatives) apparaissant dans des problématiques industrielles. Dans ce cadre, nous avons étudié des protocoles probabilistes comme le protocole CSMA/CD (IEEE 802.3) [Rc-37, Rc-34, Rc-35], utilisé dans Ethernet, et le protocole CSMA/CA (IEEE 802.11) [Rc-21], utilisée dans les communications Wi-Fi. Ces protocoles présentent la caractéristique commune de recourir à un tirage aléatoire sur le temps d'attente avant (re)transmission d'un message afin de minimiser le nombre de collisions. Nous avons expérimenté une combinaison d'outils de model-checking (PRISM, HYTECH, APMC) en interaction afin d'obtenir des analyses quantitatives de stabilité de ces protocoles et de robustesse vis-à-vis de tentatives d'utilisateurs de raccourcir abusivement leurs temps d'attente.

BlueBerries. Les mémoires embarquées sont des composants devant répondre à des impératifs de taille et de rapidité toujours plus importants. En pratique, ces mémoires très spécialisées sont directement conçues au niveau transistor. Leur vérification fonctionnelle et temporelle est généralement réalisée par le biais de simulations électriques et logiques. Dans le cadre du projet Européen MEDEA+ Blueberries, nous avons utilisé des techniques de model-checking temporisé pour vérifier formellement des caractéristiques fonctionnelles et temporelles de parties de ces composants. Dans ce projet, la description au niveau transistor de la mémoire est abstraite en une représentation fonctionnelle et temporelle, qui est elle-même abstraite en une représentation sous la forme d'automates temporisés, support de la vérification par model-checking. Avec cette méthodologie, il a été possible de déterminer les temps de réponses minimaux et maximaux de certains composants d'une mémoire fabriquée par STMicroelectronics (la SPSMALL), d'optimiser certains délais, et déterminer des contraintes liant les délais et garantissant des plages de bon fonctionnement du circuit [Rc-36][CO-25].

CORTOS. CORTOS (*Control and Observation of Real-Time Open Systems*) est un projet de l'ACI Sécurité Informatique (oct. 2003 – oct. 2006 : `http://www.lsv.ens-cachan.fr/aci-cortos/`) dont nous avons été coordinateurs [Rc-27, Rc-17]. Il s'intéressait à la synthèse de contrôleurs pour les systèmes temporisés et était en collaboration avec l'IRCCyN (Nantes) et VERIMAG (Grenoble). En 2006, nous avons organisé un workshop sur le thème du contrôle temporisé à Bonn (événement satellite de CONCUR'06). C'est dans ce cadre que s'est déroulée la thèse de Fabrice Chevalier, soutenue en juin 2007 [TH-2].

DOTS. DOTS (*Distributed, Open and Timed Systems*) est un projet de l'ARA SSIA (jan. 2007 – déc. 2009 : `http://www.lsv.ens-cachan.fr/anr-dots/`) dont nous sommes coordinateurs. Il s'intéresse aux systèmes complexes qui intègrent des aspects temporisés, des aspects distribués et des aspects interactifs avec leur environnement, et est en collaboration avec le LaBRI (Bordeaux), l'IRCCyN (Nantes), l'IRISA (Rennes) et le LAMSADE (Paris Dauphine).

MODISTE-COVER. MODISTE-COVER est un projet P2R franco-indien réunissant d'un côté les deux universités de Chennai et l'*Indian Institute of Science* de Bangalore et de l'autre

côté le LaBRI (Bordeaux), le LIAFA (Paris 7) et le LSV (2005–2009 : `http://www.labri.fr/perso/weil/frindien/modiste.html`). La thèse de Akshay S., en co-tutelle entre le LSV et l'IMSc à Chennai, a débuté dans ce cadre en septembre 2006.

PROCOPE. Il existe une collaboration avec des équipes de Aachen (RWTH Aachen) et Munich (TU Munich) qui sert au développement de l'outil de synthèse Smyle (`http://www.smyle-tool.org/`) et de la théorie correspondante. Cette collaboration est subventionnée par Procope en 2008. Nous attendons de bénéficier de l'expertise de nos partenaires allemands, notamment dans les domaines des algorithmes d'apprentissage (TU Munich) et de la méthodologie de l'ingénierie logicielle (TU Munich et RWTH Aachen).

SIMOP. SIMOP (Synergie Simulation et Model-Checking Paramétré) est un projet de deux ans initié en janvier 2007 dans le cadre de l'Institut Farman de l'ENS Cachan (en collaboration avec le LURPA). L'objectif du projet est d'évaluer les performances d'une architecture de commande distribuée sujette à panne et d'estimer la plage de valeur des paramètres de fonctionnement de l'architecture qui garantit les performances attendues.

TOAST. TOAST (Théorie des jeux, Outils de l'automatique, de l'Algorithmique et du Signal pour les Télécommunications) est un projet de deux ans, financé dans le cadre de l'Institut Farman de l'ENS Cachan et dans le cadre des Projets Exploratoires Pluridisciplinaires (PEPS) du CNRS. Il regroupe le LSV et la laboratoire SATIE (Systèmes et Applications des Technologies de l'Information et de l'Énergie) de l'ENS Cachan, et porte sur l'étude du contrôle de puissance d'émission dans les systèmes de communications modernes : ces systèmes sont soumis à des interférences liées en grande partie aux autres communications environnantes.

VALMEM. Le projet ANR VALMEM (jan. 2007 – déc. 2009 : `http://www.lsv.ens-cachan.fr/~encrenaz/valmem`), dont nous sommes les coordinateurs, s'inscrit dans le prolongement du projet BlueBerries et regroupe des partenaires universitaires (LSV, LIP6) et industriel (STMicroelectronics). L'objectif est de passer à l'échelle dans le processus de vérification de circuits mémoires fabriquées par STMicroelectonics. Ce passage se fera en automatisant, d'une part, le processus de transformation du modèle en transistors de la mémoire en un code VHDL augmenté d'informations temporelles ; en automatisant, d'autre part, le processus d'abstraction de ce code en un produit d'automates temporisés. L'ambition visée à terme est de fournir une plate-forme logicielle prototype, permettant de vérifier un jeu d'exemples de circuits mémoires commercialisés.

VerSyDis. VerSyDis (Vérification des systèmes distribués) est une ACI Sécurité Informatique (oct. 2003 – oct. 2006 : `http://www.liafa.jussieu.fr/~versydis/`) dont nous avons été les coordinateurs [Rc-20]. Il s'agissait d'aborder les problèmes de vérification et de synthèse à l'aide de formalismes distribués comme les diagrammes de séquences, les automates distribués, ou la théorie des traces. C'est dans le cadre de ce projet qu'a été initiée la thèse de Nathalie Sznajder.

VSMT. VSMT (Vérification de Systèmes Multi-tâches Temps réel) est un plan pluri-formations de l'ENS Cachan entre le LSV et le LURPA (2002 – 2005 : `http://www.lsv.ens-cachan.fr/VSMT/`). La problématique centrale de ce projet concerne la modélisation du comportement des programmes d'API (les automates programmables industriels). La thèse de Houda bel Mokadem [TH-8] s'est effectuée dans le cadre de ce projet.

Annexe B

Bilan scientifique détaillé : axe INFINI

Membres de l'axe INFINI

Responsables (période 2004–2008)
- Alain Finkel (PR ENS Cachan) ;
- Philippe Schnoebelen (DR CNRS) ;

Membres permanents
- Etienne Lozes (MCF ENS Cachan, depuis oct. 2005)
- David Nowak (CR CNRS ; détaché à l'université de Tokyo depuis 2005)

Doctorants
- Sébastien Bardin (2002-2005) ;
- Nathalie Bertrand (2003-2006) ;
- Arnaud Sangnier (depuis oct. 2005) ;
- Florent Bouchy (depuis oct. 2006) ;
- Jules Vilard (depuis sept. 2007) ;
- Remi Brochenin (depuis oct 2006) ;

Membres non-permanents
- Peter Habermehl (MCF Université Paris 7, en délégation INRIA au LSV depuis fév. 2007) ;
- Emmanuelle Encrenaz-Tiphène (MCF Université Paris 6, en délégation CNRS au LSV, 2005-2007).

Évolution de l'axe de recherche

Les techniques classiques de model-checking permettent de vérifier automatiquement la correction d'une modélisation à états finis d'un système. Néanmoins, pour certains systèmes à vérifier, la contrainte de modélisation à états finis ne peut pas être respectée sans dénaturer le problème. Par exemple, certains systèmes manipulent des données de façon essentiellement non-bornée, ou bien le contrôle n'est pas répétitif à cause d'appels récursifs emboîtés (éventuellement dans un cadre concurrent). Enfin, même avec un contrôle et des données à états finis, le système peut dépendre de paramètres et on souhaite le vérifier indépendamment de la valeur de ces paramètres, ce qui conduit à une infinité de modèles finis. Dans tous ces cas, il est très facile d'exhiber des systèmes *Turing-powerful*.

Face à ces situations s'est développée depuis une quinzaine d'années une approche riche et ambitieuse visant à développer des techniques ad-hoc permettant de vérifier algorithmiquement certaines propriétés sur certains modèles à infinité d'états. L'axe INFINI du LSV occupe une place importante dans cette communauté où il a joué un rôle leader sur les systèmes bien structurés, sur les systèmes à canaux non-fiables, et sur la théorie des accélérations de points fixes dans le model-checking symbolique.

Le bilan de nos travaux des quatre dernières années est structuré en quatre parties :

- la représentation des modèles, en particulier d'ensembles infinis de configurations, et leur manipulation effective, cf. section B.1. La question centrale est celle du calcul d'une représentation symbolique d'ensembles de configurations qui puisse être manipulée automatiquement. Le plus souvent, on cherche à calculer l'ensemble des configurations accessibles en avant, noté $\mathrm{Post}^*(\mathit{Init})$, ou celui des configurations accessibles en arrière, noté $\mathrm{Pre}^*(\mathit{Final})$.
- la caractérisation de classes de propriétés décidables sur certains systèmes infinis (section B.2). Cette étude s'est beaucoup appuyée sur les systèmes à compteurs et la théorie des systèmes bien structurés.
- les travaux plus particulièrement axés vers la mise en oeuvre effective de techniques algorithmiques, qu'il s'agisse d'implémentation au sein d'un outil tel que FAST, ou bien de résultats algorithmiques sur les structures de données symboliques particulières (section B.3).
- les systèmes manipulant des pointeurs et/ou allouant dynamiquement de la mémoire, que nous présentons à part (section B.4) car les modèles et les logiques concernées sont très spécifiques.

B.1 Modèles

Dans le domaine des systèmes infinis, les modèles génériques sont automatiquement *Turing-powerful*. Il est donc nécessaire de développer des modèles spécifiques tout en essayant d'identifier les hypothèses aussi générales que possible qui permettent d'atteindre les objectifs.

B.1.1 Une classe générale : les systèmes bien structurés

On appelle « systèmes *bien structurés* » (WSTS) les systèmes de transitions pour lesquels existe un *wqo* (*well-quasiorder*) entre les configurations qui soit compatible avec les transitions (Finkel, *Inf. & Comp.*, 1990). Des résultats de décidabilité génériques existent pour ces systèmes.

Nous avons pu montrer qu'un grand nombre de modèles utilisés pour la vérification symbolique des systèmes infinis peuvent être munis d'une structure de WSTS (Finkel & Schnoebelen, *Theor. Comp. Sci.*, 2001). Depuis la parution de cet article de synthèse, les systèmes bien structurés ont été utilisés par un grand nombre de chercheurs extérieurs au LSV ; chaque année, de nouvelles classes de systèmes bien structurés sont découvertes.

WSN *Travail en collaboration avec Jean-François Raskin et son équipe (Bruxelles, Belgique).*

Toujours avec un objectif de généralisation, nous avons étudié les systèmes bien structurés dont les configurations sont des vecteurs d'entiers (appelés *Well Structured Nets*, ou WSN). L'objectif était en particulier de recadrer les nombreux résultats de décidabilité des propriétés de sûreté sur les réseaux de Petri et leurs extensions. Formellement, un WSN est un système bien structuré où la relation de transition est donnée par un ensemble fini de fonctions récursives croissantes sur les vecteurs d'entiers [RI-69]. Pour de tels systèmes, une question fondamentale est de savoir s'il est possible de couvrir une configuration donnée. Nous avons montré que la décidabilité de cette question dépendait d'un critère d'effectivité assez général : il faut que les fonctions de transition

du système soient oméga-récursives, c'est-à-dire que leur prolongement à l'infini, sur $(\mathbb{N} \cup \{\omega\})^k$, soit encore récursif. Nous avons aussi montré comment on pouvait généraliser l'algorithme de Karp et Miller pour décider certaines questions de finitude lorsque les fonctions de transition sont oméga-récursives et fortement croissantes (c'est-à-dire que certaines de leurs projections sont strictement croissantes). Finalement, dans le cas particulier où la relation de transition est donnée par des fonctions affines (qui sont toutes oméga-récursives) croissantes, nous retrouvons et généralisons des résultats récents sur les réseaux de Petri et les systèmes à compteurs bien structurés (réseaux avec *resets* et/ou transferts, réseaux auto-modifiants, ...) [RI-69].

En collaboration avec Jean-François Raskin et son équipe, nous avons étudié la puissance d'expression (en termes de langages) d'extensions de réseaux de Petri, qui sont bien structurés, typiquement des réseaux de Petri étendus avec différents arcs de transferts (qui ne sont pas bloquants). Nous avons pu caractériser de façon fine les hiérarchies de langages et ainsi séparer de façon formelle (pour la première fois), les réseaux de Petri, les réseaux avec arcs non bloquants, les réseaux avec transferts et les réseaux avec resets [RI-45, CI-155].

Calculs de points fixes *Travail en collaboration avec Christel Baier (Dresde, Allemagne).*

Pour les systèmes bien structurés, l'évaluation d'un point fixe simple comme $Pre^*(Unsafe)$ peut se faire par un calcul itératif naïf tel que « $X_0 = Unsafe\,;\, X_{k+1} = X_k \cup Pre(X_k)$ » dont la convergence est garantie par la théorie des *well-quasiorders*. Mais les points-fixes rencontrés en vérification ne sont pas tous aussi simples.

Dans [CI-98], motivés par des problèmes issus de la vérification des systèmes communicants probabilistes (cf. section A.3.5), nous avons généralisé le résultat classique de convergence à une large famille d'expressions de points fixes emboîtés, obtenant ainsi la décidabilité de nombreuses propriétés telles que l'équivalence avec une spécification finie (cf. [RI-38]), le model-checking de formules de $\exists\mathsf{CTL}$, ou encore les jeux d'accessibilité, éventuellement dans un environnement probabiliste.

Un cadre topologique à la théorie des bien structurés L'étude des capacités et leur positionnement dans la théorie des domaines (section C.5) a apporté un éclairage nouveau et prometteur sur la théorie des systèmes bien structurés, tout en asseyant la théorie dans un cadre plus général. Nous avons en effet montré qu'une classe naturelle d'espaces topologiques d'états sur lesquels le model-checking d'un μ-calcul modal adéquat est décidable est formée des espaces noethériens, c'est-à-dire des espaces dans lesquels tout ouvert est quasi-compact [CI-78]. La notion avait été introduite en géométrie algébrique, avec Zariski. Tout préordre induit une topologie dite d'Alexandroff, dont les ouverts sont les clos par le haut. De façon peut-être surprenante, nous avons observé qu'un préordre était beau si et seulement si sa topologie d'Alexandroff était noethérienne. Ceci nous a permis de généraliser la notion de système de transition bien structuré, et ouvre la voie à des techniques de complétion de wqo qui permettent de généraliser les techniques symboliques en avant à la Karp et Miller.

B.1.2 Systèmes à compteurs

Les systèmes à compteurs sont des automates finis agissant sur un nombre fini de variables entières, les *compteurs*. Les transitions sont gardées par des contraintes généralement exprimées en logique de Presburger, et contiennent des opérations de mise à jour des compteurs généralement données sous forme de transformations affines. Ce modèle généralise les machines à compteurs

de Minsky[1] et permet d'exprimer simplement de nombreuses familles de systèmes : algorithmes distribués, protocoles, abstractions de programmes impératifs, systèmes temporisés paramétrés, etc. Par ailleurs les systèmes à compteurs ont la puissance des machines de Turing. Notre ambition a donc été de mettre au point des techniques symboliques suffisamment puissantes pour parvenir à calculer les ensembles d'accessibilité dans « un grand nombre de cas » (une notion subjective, mais que nous préciserons au paragraphe B.3.3).

Depuis quatre ans, nous avons mené une étude en profondeur de ces modèles. Cette étude a débouché sur des développements algorithmiques importants que nous détaillerons dans la section B.3.1. En ce qui concerne les aspects plus fondamentaux des systèmes à compteurs, nous avons pu obtenir des résultats de décidabilité nouveaux en considérant les systèmes *plats* (ou parfois trace-applatissables), c.-à-d. où le graphe de contrôle ne comporte pas de boucles imbriquées. Il est connu que, dans le cas des machines de Minsky, la platitude suffit à assurer la décidabilité de l'accessibilité. Ce résultat ne se généralise pas aux systèmes à compteurs plats, où l'accessibilité est indécidable, même en l'absence de garde car une seule boucle étiquetée par une fonction affine par morceaux [CI-104] peut simuler une machine de Minsky. Nous avions montré que l'accessibilité est décidable pour les systèmes à compteurs affines plats *où le monoïde des matrices des mises à jour est fini* [CI-180] et nous avons étendu ce résultat à la décidabilité de CTL * [CI-104]. Ceci nécessite quelques explications. On représente chaque mise à jour apparaissant dans le système sous une forme vectorielle « $X := M.X + V$ » où X est l'ensemble des d compteurs, M est une matrice de $\mathbb{N}^{d\times d}$, et V est un vecteur de $\mathbb{Z}^d$. Notre restriction suppose donc que le monoïde engendré multiplicativement par les différentes matrices M du système soit fini. Notons que cette hypothèse ne pose aucune restriction sur les gardes ou sur les vecteurs V des mises à jour. La restriction sur le monoïde des matrices est très libérale du point de vue des applications. Par exemple, elle est automatiquement vérifiée si les matrices ne contiennent que des 0 et des 1 avec au plus un 1 par ligne (ou par colonne), ce qui généralise les réseaux de Petri, les compteurs avec transferts et affectations de constantes.

B.1.3 Systèmes communicants

Les systèmes d'automates communiquant par canaux fifo sont un des modèles les plus naturels pour les protocoles asynchrones. Ici, un contrôle fini agit sur des canaux non bornés. Dans une configuration donnée, le contenu d'un canal fifo (un mot représentant une suite de messages) modélise des messages émis, non reçus, ainsi que leur ordre d'émission. Il s'agit là aussi d'un modèle qui a la puissance des machines de Turing. Une variante de ce modèle consiste à considérer que les messages peuvent parfois être perdus au cours de leur acheminement : on parle de canaux « non-fiables ». Paradoxalement, dans ce cas, l'accessibilité devient décidable.

Systèmes à canaux non-fiables. Les systèmes à canaux non-fiables sont aussi appelés « LCS» , pour *lossy channel systems*. Ces dernières années le problème de l'accessibilité pour les LCS a été relié à de nombreuses questions issues de domaines *a priori* divers : logiques multi-modales, automates temporisés alternants, logiques temporelles *data-sensitive*, logique temps-réel métrique, etc. Ces connexions sont surprenantes et on peut s'attendre à ce qu'en apparaissent de nombreuses autres car elles révèlent que les LCS sont un modèle de calcul occupant une place cruciale jusqu'alors non reconnue.

Nous avons récemment mis en lumière cette place cruciale en exhibant une variante nouvelle du problème de correspondance de Post qui soit algorithmiquement équivalente à l'accessibilité des LCS [CI-54, CI-43]. Cette variante se décline de différentes façons, mais le motif général

[1] Où les gardes sont restreintes aux tests à zéro, et où les mises à jour sont uniquement des incrémentations et des décrémentations.

consiste à remplacer l'égalité (la « correspondance ») par la relation sous-mot et à ajouter une contrainte régulière sur les solutions admissibles. L'équivalence est non-triviale et le « problème de plongement de Post » devrait jouer un rôle pivot dans l'établissement des connexions entre problèmes équivalents aux LCS.

Par ailleurs, nous avons résolu le problème de caractériser exactement la complexité des LCS [CI-33]. En montrant que les LCS sont assez puissants pour calculer les fonctions jusqu'au niveau ω^2 de la hiérarchie de Grzegorczyk, ce travail majeur clôt un problème ouvert depuis 1990. Ses répercussions vont au-delà du seul modèle des LCS en vertu des équivalences dont nous parlions plus avant.

Systèmes probabilistes à canaux non-fiables. *Travail en collaboration avec Christel Baier (Dresde, Allemagne).*

Les LCS ont été introduits à l'origine pour modéliser les protocoles de communication censés fonctionner dans un environnement non fiable où les messages envoyés pouvaient être perdus.

Ce modèle se prête bien à la vérification de propriétés de sûreté par des techniques de model-checking symbolique, mais beaucoup moins à la vérification des propriétés de vivacité. D'une part, les propriétés de vivacité sont indécidables pour les LCS, et d'autre part le modèle est trop pessimiste et ne satisfait quasiment aucune propriété de vivacité puiqu'il est *possible* que jamais aucun message ne soit transmis.

Pour vérifier des propriétés de vivacité, nous avons introduit une variante du modèle des LCS où les pertes de messages suivent une loi de probabilité. Cette modélisation est très naturelle dans un cadre de tolérance aux pannes, et permet d'exprimer des notions probabilistes de correction (qualitatives ou quantitatives), ce qu'un modèle possibiliste ne sait pas faire. La sémantique du modèle est alors donnée en termes de chaînes de Markov, ou de processus de décision markoviens, *à infinité d'états* pour lesquelles la vérification algorithmique pose de réels défis.

Dans ce cadre, nos travaux ont exhibé une structure particulière aux pertes de messages probabilistes : l'existence d'un attracteur fini qui permet de ramener les propriétés de vivacité à des questions d'accessibilité dans le modèle purement probabiliste où les actions du système sont gouvernées par des lois de probabilité [RI-51, RI-55].

C'est toutefois le modèle combinant non-déterminisme des actions et probabilités pour les pertes de message qui est le plus adapté aux problèmes de vérification des protocoles qui nous intéressent. Pour ce modèle nous avons montré que la décidabilité est perdue sauf si l'on se restreint à des adversaires (en général il s'agit de l'environnement) à mémoire finie. Dans ce cas, il existe des solutions algorithmiques certes très complexes, mais néanmoins à base de model-checking symbolique [RI-20]. Nous avons pu développer un outil prototype mettant ces méthodes en oeuvre et qui a permis d'analyser les propriétés de vivacité quelques protocoles simples [RI-55, TH-7].

Systèmes à canaux half-duplex Nous avions montré, avec Gérard Cécé, qu'il existait une classe de systèmes communicants, les systèmes avec une communication half-duplex, pour lesquels la plupart des propriétés d'accessibilité étaient décidables en temps polynomial ; la propriété pour un système d'être half-duplex étant elle-même décidable en temps polynomial. Nous avons continué et précisé cette étude en prouvant que le model-checking de LTL et de CTL est, par contre, indécidable pour ces systèmes [RI-56]. Nous avons également proposé un nouveau semi-algorithme avec accélérations, dans le cadre du *Regular Model Checking*, pour l'ensemble des systèmes communicants qui a la propriété de terminer sur les systèmes half-duplex. Il est remarquable que cette classe de systèmes est la seule classe connue ayant des propriétés d'accessibilité décidables en temps polynomial et possédant un test polynomial pour décider si un système quelconque fait partie de la classe.

B.2 Logiques pour la vérification des aspects infinis

B.2.1 Vérification de systèmes à compteurs plats

Travail en collaboration avec V. Goranko et G. van Drimmelen (Johannesburg, Afrique du Sud) dans le cadre d'un projet bilatéral CNRS-NRF.

Nous avons analysé comment étendre l'outil FAST pour vérifier des propriétés temporelles plus riches que la simple accessibilité. Pour ce faire, nous avons revisité certains résultats théoriques à l'origine de cet outil (platitude des modèles, applatissement des systèmes, et traduction vers une représentation symbolique basée sur l'arithmétique de Presburger) et nous les avons étendus dans le but de les inclure dans une prochaine version de FAST [CI-104]. Plus précisément, nous avons introduit une extension de $\mathsf{CTL}^\star$ dans laquelle les formules atomiques caractérisent des ensembles de configurations définissables dans l'arithmétique de Presburger. Les modèles de ce langage de spécification sont des systèmes à compteurs de Presburger pour lesquels la mise à jour des compteurs est régie par des contraintes de Presburger. Il s'agit d'un modèle qui capture naturellement celui des machines de Minsky et donc de nombreux problèmes y sont rapidement indécidables. Nous avons conçu une sous-classe de systèmes à compteurs (dits "admissibles") pour laquelle le model-checking (pour toutes les formules de la logique) est non seulement décidable mais aussi définissable dans l'arithmétique de Presburger connue pour être décidable. Il a été aussi possible de tirer profit de la platitude dans les formules en augmentant les opérateurs linéaires "next" et "until" d'opérateurs temporels définissables à partir de CQDD (automates plats avec contraintes de Presburger sur le nombre de passages dans les transitions) en imitant ce que P. Wolper a fait pour étendre LTL avec la logique ETL. Nous avons établi que cette nouvelle extension demeure décidable.

B.2.2 Propriétés qualitatives et quantitatives avec logiques temporelles

Les logiques temporelles forment une classe de langages de spécifications formelles pour énoncer des propriétés comportementales de systèmes de transition. Plus généralement, ces logiques ont pour modèles des graphes dont les noeuds et arêtes sont étiquetés. Le domaine de tels systèmes est généralement infini, par exemple de la forme $Q \times \mathbb{N}^k$ où Q est un ensemble fini d'états de contrôle et $\mathbb{N}^k$ est le domaine d'interprétation de k compteurs. On s'intéresse ici à des logiques temporelles qui peuvent spécifier des propriétés sur des séquences d'éléments de $Q \times \mathbb{N}^k$ (des parties de calculs) afin de vérifier des propriétés quantitatives sur les données (de domaine $\mathbb{N}^k$). Dans les logiques temporelles usuelles (LTL, CTL, $\mathsf{CTL}^\star$) une variable propositionnelle p représente une propriété de l'état courant du système. Par exemple, p peut être vraie lorsque la valeur de la variable x est supérieure à la valeur de la variable y après exécution de l'instruction courante. Une solution plus satisfaisante consiste à inclure dans le langage de spécification la possibilité d'exprimer directement ces contraintes entre variables du programme en abandonnant l'abstraction sous-jacente à l'usage des variables propositionnelles. Quand les variables sont typées, elles peuvent être interprétées dans des domaines spécifiques comme les entiers, les réels, les chaînes de caractères etc. Ainsi une proposition "x est supérieure à la prochaine valeur de y" s'écrit alors "$x < \mathtt{X}y$" dans l'extension proposée une fois que les modèles sont des séquences de valuations. Dans la suite, on note LTL ($\mathcal{D}$) la version de la logique LTL pour laquelle les formules atomiques sont des contraintes sur le domaine (structure relationnelle) $\mathcal{D}$. Des exemples de domaines concrets d'interprétation des variables sont $(\mathbb{N}, =, <)$ ou $(\{0,1\}^*, \subset)$.

Logiques avec contraintes de Presburger *En partie avec Deepak D'Souza (IISc, Bangalore).*

Il a été montré [RI-29] que les problèmes de model-checking et satisfaisabilité pour les logiques LTL $(\mathbb{N}, =, <)$, LTL $(\mathbb{Z}, =, <)$ et LTL $(\mathbb{R}, =, <)$ sont PSPACE-complets sachant que les entiers et les réels sont des domaines d'interprétation usuels pour les variables de programmes. La preuve pour LTL $(\mathbb{Z}, =, <)$ est très complexe [RI-29] car nous avons aussi établi que les langages de mots infinis de contraintes atomiques associés aux formules de cette logique ne sont pas toujours ω-réguliers, ce qui est une propriété tout à fait inattendue. Dans la continuité de ce travail, dans [CI-184, RI-37] une théorie du premier ordre de contraintes de périodicité IPC^+ a été introduite et il a été montré que LTL (IPC^+) admet des problèmes de model-checking et satisfaisabilité PSPACE-complets. Une application remarquable de ce résultat est la caractérisation de la complexité de problèmes relatifs à la définition de granularités temporelles (connues aussi sous le terme de "calendrier") à base d'automates. Nous avons aussi introduit une version contrainte de LTL, nommée LTL ($\text{IPC}^\star$), sur un langage de contraintes qui inclut strictement IPC^+ et celui sous-jacent à $(\mathbb{N}, <, =)$. Nous avons montré que les problèmes de model-checking et de satisfaisabilité pour LTL ($\text{IPC}^\star$) sont aussi PSPACE-complets [CI-142] ce qui généralise certains résultats de [CI-184, RI-29]. Ainsi, $\text{IPC}^\star$ apparaît comme un extension optimale à ce jour pour laquelle son plongement dans LTL est décidable en PSPACE. En corollaire, nous obtenons que le model-checking des automates relationnels intégraux de Čerans pour les propriétés linéaires est décidable en espace polynomial alors que le problème est connu pour être indécidable avec la variante de CTL correspondante. De même, le model-checking pour le fragment existentiel de $\text{CTL}^\star$ sur ces automates relationels est montré décidable dans [CI-99] en utilisant la théorie des systèmes bien-structurés. Il est donc facile de montrer que LTL avec contraintes de Presburger au niveau atomique est indécidable. Dans [CI-85] (voir aussi [CO-21]), nous avons affiné ce résultat d'indécidabilité en effectuant diverses restrictions syntaxiques sur les formules et en considérant trois problèmes distincts : le model-checking sur les DL-automates (mise à jour des compteurs avec des contraintes de différence pas nécessairement déterministes), la satisfaisabilité et le model-checking sur les automates à compteurs. Une conséquence intéressante de nos résultats est que le model-checking des automates à un compteur sur LTL (QFP) est PSPACE-complet. De même, la logique $\mathcal{L}_p$ de Comon & Cortier (CSL'00) restreinte à une variable est dans PSPACE. Un exposé invité sur les logiques du temps linéaire avec contraintes de Presburger a donné lieu à l'article de synthèse [RI-34], voir aussi la thèse de Régis Gascon [TH-1].

Contraintes de Presburger et logiques modales. *Collaboration avec D. Lugiez (LIF, Marseille)*

L'interrogation des données présentes sur le Web est rendue difficile à la fois par l'hétérogénéité des formats de données et par la masse de données devant être analysées. Certaines techniques d'interrogation s'appuient sur une modélisation des données sous forme de graphes (comme pour les documents XML par exemple). Ainsi, déterminer si un document est bien typé peut se ramener à un problème de model-checking. De plus, pour l'optimisation de requêtes, il est important de pouvoir comparer des contraintes, ce qui d'un point de vue logique est équivalent à un problème de validité. L'introduction de contraintes numériques relatives à la structure de graphes de telles données est apparue récemment avec une approche à base d'automates, même si de telles contraintes ne sont pas encore intégrées dans les langages du standard W3C comme XPath. Nous avons considéré une logique EXML qui permet d'exprimer des contraintes sur le nombre des états successeurs à l'aide de fragments de l'arithmétique de Presburger et des contraintes de régularité lorsque les fils d'un noeud sont linéairement ordonnés. Il s'agit essentiellement d'un fragment d'une logique connue dans la littérature mais sans opérateur de point fixe. On s'intéresse ici à des logiques dont la partie qualitative modale est moins riche que le mu-calcul modal très utilisé pour le model-checking mais dont les contraintes numériques de Presburger sont plus riches que celles du mu-calcul modal gradué. Il s'agit de trouver un bon

compromis entre l'expressivité de contraintes numériques et la complexité des logiques. Nous avons montré dans [CI-117] que pour $k \geq 0$ fixé, EXML restreinte aux formules avec au plus k contraintes de régularité est PSPACE-complète ainsi que EXML sans contrainte de Presburger (mais avec toutes les contraintes de régularité). De plus, nous avons trouvé une réduction entre la "Sheaves Logic" SL de D. Lugiez et S. Dal Zilio et EXML qui préserve le nombre de contraintes de régularité. Cela permet d'améliorer considérablement la complexité connue de SL dont le meilleur algorithme auparavant avait une complexité non-élémentaire.

Formalismes avec registres *En partie avec Deepak D'Souza (IISc, Bangalore), Ranko Lazić (Warwick, Royaume-Uni).*

La possibilité de stocker une valeur dans un registre et de tester ensuite la valeur d'un registre avec une donnée courante est un mécanisme que l'on retrouve dans de nombreux formalismes logiques mais aussi naturellement dans les langages de programmation. Ce type de quantification est présent par exemple dans les logiques temps-réel et dans les logiques temporelles. Ainsi, dans la logique TPTL de Alur et Henzinger, une formule $x \cdot \phi(x)$ est sémantiquement égale à $\phi(t)$ où t est la valeur courante du temps. Un même mécanisme général apparaît dans les automates à registres qui reconnaissent des mots construits sur des alphabets infinis. Nous nous intéressons ici à la vérification d'automates contraints en présence de l'opérateur *freeze*. Cet opérateur a déjà montré son utilité pour les requêtes de données semi-structurées, la vérification de systèmes temporisés ou encore la vérification de systèmes dynamiques avec ressources.

Dans [CI-148], nous avons analysé les problèmes de décidabilité et complexité pour LTL contraint en présence de l'opérateur *freeze* (cf. un premier résultat dans [RI-37]). Seuls les opérateurs du futur ont été pris en compte. On note CLTL $^{\downarrow}(\mathcal{D})$ la version contrainte de LTL avec l'opérateur *freeze* sur le domaine concret $\mathcal{D}$. Nous avons montré que si $\mathcal{D}$ est fini avec au moins deux éléments et l'égalité alors le problème de satisfaisabilité pour CLTL $^{\downarrow}(\mathcal{D})$ est EXPSPACE-complet. En l'absence de l'opérateur *freeze*, le problème est seulement dans PSPACE. Le résultat principal de ce travail a consisté à montrer que CLTL $^{\downarrow}(\mathbb{N}, =)$ est hautement indécidable (Σ_1^1-complet) ce qui est surprenant à la vue du pauvre langage de contraintes. Ce résultat est obtenu avec seulement deux registres et une variable flexible ce qui est un résultat proche de l'optimal [CI-148, RI-31]. Un autre fragment a été montré décidable dans [CI-84] en mettant en relation une logique de données avec un problème du vide pour une classe d'automates à compteurs. Nous avons par ailleurs affiné dans [CI-116] les résultats précités en considérant la logique LTL$^{\downarrow}(\sim; \mathtt{X}, \mathtt{X}^{-1}, \mathtt{U}, \mathtt{S})$ avec des opérateurs du passé et du futur, une variable flexible et en faisant la distinction entre la satisfaisabilité avec modèles finis ou avec modèles infinis. Cette logique a en effet de nombreux liens avec des classes d'automates à registres pour lesquels nous avons aussi établi des résultats de complexité pour le problème de la vacuité [CI-116]. Ainsi, nous avons étendu l'approche par automates aux logiques munies de l'opérateur *freeze*.

B.3 Outils : théorie et expérimentation

B.3.1 Algorithmique de la vérification symbolique

Face au problème de l'explosion combinatoire du nombre des états, et a fortiori pour les systèmes à infinité d'états, le calcul des ensembles d'accessibilité est nécessairement *symbolique* : des objets symboliques finis (graphes, formules, contraintes, ...) sont utilisés pour représenter et manipuler des ensembles infinis de configuration. Ici la question principale est de mettre au point les représentations symboliques et les algorithmes de calcul d'ensembles de configurations les mieux adaptés aux applications considérées. Il faut pouvoir (1) effectuer efficacement les opérations ensemblistes (union, complément, ...) sur les représentations symboliques, (2) calculer l'ensemble

des successeurs immédiats (ou des prédécesseurs) d'un ensemble représenté symboliquement et (3) comparer deux ensembles pour détecter la terminaison des calculs de points-fixes. Pour nos modèles généralistes, nous nous plaçons dans le cadre du *model-checking régulier*, où l'on manipule des ensembles réguliers de configurations, généralement représentés via des structures à base d'automates finis. Ce cadre se prête très bien aux calculs ensemblistes, et la recherche d'algorithmes optimaux pour telle ou telle sous-classe de reconnaisseurs bénéficie des acquis de la théorie des automates. Les calculs de successeurs et de prédécesseurs immédiats se ramènent généralement à des transductions d'ensembles réguliers.

Une fois ce cadre général choisi, deux difficultés restent à surmonter.

Le coût des calculs « élémentaires » : il faut pouvoir, pour une application donnée, identifier la bonne famille d'ensembles réguliers et la bonne classe d'automates pour les représenter, qui permettront des algorithmes ensemblistes suffisamment efficaces.

La convergence des calculs de points-fixes : dans le cas de systèmes infinis, les calculs itératifs des ensembles d'accessibilité, tels que la suite $\mathit{Init}, \mathrm{Post}(\mathit{Init}), \mathrm{Post}^2(\mathit{Init}), \ldots$ ne se stabilisent pas en un nombre fini d'étapes. Il est donc indispensable de mettre au point des méthodes, dites d'*accélération* (de la convergence), qui sans toujours garantir la stabilisation finale, peuvent au moins la rendre fréquente.

Nos contributions sur ces thèmes portent sur la théorie des accélérations (paragraphe B.3.1), la vérification symbolique des systèmes à compteurs (paragraphe B.3.1) et l'algorithmique des représentations symboliques (paragraphe B.3.2).

Vers une théorie des accélérations L'accélération d'une relation R consiste à calculer en un temps fini la clôture réflexive et transitive R^* de R. Si l'on peut calculer et manipuler symboliquement « post^* », c'est-à-dire l'accélération de la relation de transition entre configurations « post^* », on obtiendra toute l'information nécessaire pour les questions d'accessibilité. Pour les classes de modèles où ce calcul est impossible (ce qui est habituellement le cas), il est parfois possible d'identifier une sous-relation $R \subseteq \mathit{post}$ qu'on sait accélérer. L'injection de R^* dans un calcul de point-fixe classique permet alors de prendre en compte des suites de transitions de longueur non bornée et rend beaucoup plus plausible la stabilisation éventuelle du calcul [CI-136, CI-164]. Pour les systèmes où un contrôle fini agit sur des variables à domaine non borné, les relations correspondant à un circuit dans le graphe de contrôle (une « boucle ») sont des candidats particulièrement intéressants : d'une part il est souvent possible de les accélérer effectivement, d'autre part la pratique montre que l'injection des accélérations de boucles est très souvent suffisante pour stabiliser. Notons enfin que les techniques d'accélérations diffèrent des techniques d'élargissement. Ces dernières utilisent des approximations supérieures lors des calculs de points-fixes, ceci afin de garantir la stabilisation et quitte à sacrifier l'exactitude du calcul et à introduire des faux positifs. Les techniques d'accélérations n'introduisent, elles, aucune approximation mais sacrifient la garantie de terminaison.

Nous avons également donné dans [CI-136] un cadre théorique permettant de comprendre la portée et l'intérêt des techniques d'aplatissement utilisées par FAST ; celles-ci peuvent donc maintenant être aussi utilisées par d'autres outils supportant les accélérations comme TREX et LASH.

Vérification des systèmes hétérogènes La vérification de nombreux protocoles modernes nécessite de considérer simultanément différents types de variables à domaine non borné. Pour vérifier de tels systèmes, que nous appelons « systèmes hétérogènes », les techniques disponibles sont encore peu satisfaisantes, et en tout cas loin d'égaler les performances de techniques dédiées

à un type de données. Les problèmes à résoudre concernent essentiellement les représentations symboliques et les techniques d'accélération. Notre premier travail [CI-164] a posé le cadre et permis de composer les accélérations des compteurs, des files et des horloges. Nous avons commencé une recherche visant à vérifier les systèmes à compteurs temporisés en utilisant et combinant les techniques propres aux automates (finis) temporisés et aux systèmes à compteurs dont les ensembles d'états accessibles sont semi-linéaires (ou Presburger-définissables).

Algorithmique et accélération des systèmes à compteurs Les techniques d'accélération des systèmes à compteurs sont basées sur l'accélération des fonctions affines. Si le monoïde engendré par la matrice M d'une mise à jour « $X := M.X + V$ » est fini (cf. paragraphe B.1.2) alors l'accélération de cette mise à jour est représentable par une formule de Presburger (ou par un automate) calculable. Nous avons montré que ce résultat était préservé par composition : l'accélération de toute composée de fonctions d'un système à compteurs à monoïde fini est effectivement représentable par un automate [RI-7]. Pour implanter efficacement les techniques d'accélération, il est nécessaire de réduire le nombre de fonctions candidates à l'accélération. Soit F un ensemble fini de fonctions affines définies sur des ensembles semi-linéaires et dont le monoïde des matrices est fini. Le nombre de fonctions distinctes que l'on peut obtenir en composant k fonctions de F est exponentiel (en k) en général mais nous avons montré qu'il est polynomial quand les domaines de définition des fonctions de F sont convexes. Il est par ailleurs possible de construire cet ensemble de fonctions en temps polynomial. Ce résultat est utilisé par l'outil Fast (voir section B.3.3) pour diminuer de façon drastique le nombre de boucles considérées comme candidates à l'accélération.

B.3.2 Implantation des représentations symboliques

Algorithmique des automates binaires Parmi les différentes représentations symboliques d'ensembles de vecteurs d'entiers, celle par automates finis (remontant à Büchi) jouit de nombreux avantages. Elle a d'abord un grand pouvoir d'expression car les automates finis permettent de représenter tous les ensembles semi-linéaires (donc toutes les formules de Presburger) ; elle dispose de bonnes propriétés de clôture (en particulier par union, intersection et complémentaire) ; le vide et l'inclusion sont décidables ; et les automates finis ont une forme canonique (l'automate déterministe minimal).

Un automate représentant un ensemble $X \subseteq \mathbb{N}^p$ de vecteurs d'entiers reconnaît tous les codages (en binaire par exemple) des éléments de X. Les NDD de Boigelot et Wolper utilisent comme alphabet l'ensemble des p-uplets de $\{0,1\}^p$ et un mot de longueur n représente un vecteur à p composantes où chaque composante est un entier de n bits. Les automates binaires non ambigus (« unambiguous binary automata », ou UBA) que nous avons définis utilisent l'alphabet $\{0,1\}$: un mot de longueur pn joue le même rôle qu'un mot de longueur n dans un NDD et représente p entiers de n bits entrelacés.

Alors que les NDD et les UBA paraissent assez similaires, les UBA se prêtent mieux aux calculs nécessaires pour les accélérations car ils sont stables par résidus, ce qui n'est pas le cas des NDD. De plus, les UBA sont toujours plus petits que les NDD correspondants.

Il est par ailleurs souhaitable de pouvoir passer de la représentation par automate à celle par formule logique. On sait que les automates (UBA ou NDD) permettent de représenter plus que les ensembles semi-linéaires, et Muchnik a prouvé en 2003 qu'on peut décider si l'ensemble reconnu par un automate est bien semi-linéaire.

Nous avons étudié la taille de l'automate représentant $\mathrm{Pre}(X)$ et de celui représentant l'ensemble $\mathrm{Pre}^{\leq k}(X)$ des k-prédécesseurs (les prédécesseurs en k transitions au plus), ceci pour

pour un ensemble X également représenté par un automate binaire. Nous avons montré que le calcul d'un UBA représentant $\mathrm{Pre}(X)$ pouvait être effectué en temps polynomial. De plus, la taille asymptotique de l'automate représentant $\mathrm{Pre}^{\leq k}(X)$ est polynomiale en k pour tout système à compteurs dont le monoïde des matrices est fini [CI-177]. Lorsqu'on enlève l'hypothèse de monoïde fini, et même en se restreignant à des ensembles semi-linéaires et à des fonctions affines, la taille devient exponentielle en k [CI-179].

Nous avons également prouvé que l'enveloppe convexe d'un NDD est un polyèdre effectivement calculable en temps exponentiel [RI-60]. Ce résultat ouvre la voie à des techniques d'approximations supérieures par enveloppes convexes d'ensembles d'états accessibles ayant une complexité exponentielle dans le pire des cas (on peut donc espérer un meilleur comportement sur des études de cas qui n'atteignent pas les cas pires).

B.3.3 Outils

L'outil FAST Fast [CO-31], disponible à `www.lsv.ens-cachan.fr/fast/`, calcule automatiquement, de façon exacte, les ensembles d'états accessibles de systèmes à compteurs dans lesquels chaque transition est donnée par une fonction affine dont le domaine de définition (la garde) et les états initiaux sont exprimés par une formule de Presburger, ou même un UBA.

Fast calcule les accélérations des boucles de contrôle dont le monoïde des matrices est fini, en commençant par les boucles de longueur 1, puis 2, etc. Si l'on rencontre un ensemble de configurations représenté par un automate d'une taille dépassant une borne donnée, on revient en arrière et on essaie d'autres boucles.

Cette heuristique converge et calcule l'ensemble d'accessibilité dans 80% des 40 études de cas que nous avons examinées ; certaines sont des abstractions de programmes JAVA multithread, des protocoles de systèmes embarqués, des puzzles académiques difficiles [CO-31].

L'algorithme qui prend en entrée une fonction affine f dont le domaine de définition est donné par une formule de Presburger et qui produit un UBA représentant la relation d'accélération de R_f^* est dans 3-EXPTIME (en la taille de la fonction) et dans 5-EXPTIME en la dimension du système. Dans le cas où les fonctions affines sont des translations avec des domaines de définition convexes, on peut simplifier notablement ce calcul (*cf.* paragraphe B.3.1).

Nous avons aussi comparé les techniques d'accélération des outils Fast et TREX [CI-150]. Les expérimentations conjointes montrent que TREX a un bon comportement (semblable ou plus rapide que Fast) sur les systèmes sans paramètre, probablement car l'algorithmique polynomiale des DBM est alors plus efficace que l'algorithmique des automates binaires codant les formules de Presburger. Par contre, dès que les systèmes ont au moins un paramètre, TREX ne termine plus dans des temps raisonnables probablement car l'appel à des *solvers* externes pour manipuler les paramètres le ralentit considérablement [CI-150].

Enfin nous avons utilisé Fast pour trouver des fonctions de rang (des preuves de terminaison) plus générales que des fonctions affines [CI-2].

L'outil TOPICS et son prototype Dans le cadre du projet RNTL Averiles et de la thèse CIFRE d'Arnaud Sangnier, un analyseur de programme C est en cours d'implémentation. Cet outil, baptisé TOPICS, est dédié à la recherche de fautes mémoires dans les programmes manipulant des structures de données récursives telles les listes chaînées, voire des structures plus riches telles que les listes doublement chaînées, les listes avec pointeur de tête, etc. Cet outil, programmé en Java, reprend et prolonge une première implantation de la traduction des systèmes à pointeurs en systèmes à compteurs.

La traduction est très efficace, au moins sur les études de cas classique (retournement de liste, fusion, déallocation de liste, *etc.*) En revanche, l'outil Fast ne semble pas aujourd'hui optimal

pour l'analyse du système à compteur généré, puisqu'il demande un temps de l'ordre de la minute pour vérifier la sûreté du retournement de liste, et ne donne pas de réponse en temps et mémoire acceptable pour la fusion de liste. Cette mauvaise performance n'est probablement pas à mettre complètement sur le compte de notre traduction, puisque d'autres analyseurs de systèmes à compteurs sont capables d'examiner les systèmes que nous générons en des temps beaucoup plus courts (de l'ordre de la seconde).

L'outil TOPICS (*Translation Of Pointers Into Counters*) devrait compléter et améliorer ce premier prototype. Le parser de l'outil TOPICS est aujourd'hui opérationnel et permet, à partir d'un programme écrit dans un C relativement restreint, de construire le graphe de contrôle épuré du programme. Il prend aussi en compte certaines primitives de manipulation de threads et la traduction en produits d'automates devrait bientôt être implémentée. L'essentiel du travail sur l'outil TOPICS reste donc la réimplémentation des graphes mémoires symboliques et des opérations de post symbolique. Par ailleurs, il est envisagé d'étendre la classe des graphes mémoires pour pouvoir analyser des programmes avec des listes étendues (doublement chaînées, pointeur en tête de liste, etc.) ce qui met en jeu une algorithmique bien plus complexe que celle utilisée dans le prototype Caml.

En résumé l'outil TOPICS devrait permettre d'analyser de manière quasi automatique des programmes concurrents à nombre de mutex borné manipulant des structures de données récursives de type « listes étendues », la seule intervention de l'utilisateur consistant à préciser le motif de récursion pour chaque structure de données.

B.3.4 Expérimentations : études de systèmes embarqués

Les systèmes embarqués sont notre principal domaine d'application, ce qui explique notre participation aux projets européens FP5 IST ARTIST et FP6 REX ARTIST 2. Ces systèmes, omniprésents, mettent généralement en œuvre des protocoles distribués combinant des aspects temps-réel, paramétrés et de tolérance aux fautes. Nous traiterons dans une autre section les problèmes spécifiques à la mémoire dynamique.

B.3.5 Protocoles embarqués dans les automobiles

Le *Time-Triggered Protocol* (`TTP`) est une architecture pour des systèmes embarqués critiques. Il est soutenu par un consortium réunissant de nombreux industriels (Alcatel, Airbus, Audi, EADS, PSA, Renault, entre autres).

Le `TTP` propose un ensemble de mécanismes pour pallier certaines défaillances du système, telles que la désynchronisation des horloges, la perte de messages, l'arrêt d'un des composants, etc. Nous nous sommes concentrés sur ce dernier point (*membership algorithm*). Des travaux précédents avaient prouvé l'algorithme utilisé, mais de façon manuelle. Aucune preuve complètement automatique n'avait été proposée. Nous avons utilisé l'outil Fast sur cet algorithme non trivial et la propriété voulue a été prouvée (dans le cas d'une unique défaillance) par le logiciel sans aucune intervention de l'utilisateur. Fast a également permis de montrer que l'algorithme était valide pour deux défaillances. Il a fallu pour cela développer une nouvelle méthode d'accélération plus efficace et faire des abstractions sur le modèle [CI-180].

B.3.6 Protocoles de mémoire cache

Les protocoles de gestion des mémoires caches sont nécessaires pour pouvoir réellement utiliser les performances des processeurs car la vitesse de ceux-ci progresse plus vite que celles des mémoires. Or les protocoles de caches ont des comportements complexes. Il est donc important de

les vérifier formellement et en particulier de vérifier la propriété de cohérence : différents processus doivent avoir la même perception de leurs données partagées. De plus, il est souhaitable de pouvoir vérifier cette propriété de cohérence indépendamment du nombre n de caches, c'est-à-dire de pouvoir assurer qu'un protocole est cohérent quel que soit le nombre n de caches utilisés. Ce problème est une instance du problème général du model-checking paramétré, dont on sait qu'il est indécidable. Emerson et Namjoshi ont introduit à LICS 1998 un modèle de protocoles de caches (les *protocoles broadcast*), qui est équivalent au modèle des automates à compteurs dans lesquels le test à zéro est remplacé par une opération de transfert d'un compteur vers un autre. Nous avions montré en 1999, avec Javier Esparza et Richar Mayr, que ce modèle était bien structuré, et donc que le problème d'accessibilité d'un état de contrôle était décidable. Nous avons par la suite attaqué le problème de la vérification des principaux protocoles de caches (Berkeley, MESI, Illinois, Dragon, Firefly, Futurebus, MOESI, Synapse) différemment : plutôt que de chercher une classe décidable, nous avons appliqué les techniques d'accélération sur les automates à compteurs associés aux protocoles broadcast. Ceci est possible car ces automates à compteurs ont tous des monoïdes (de matrices) finis, il est donc toujours possible d'accélérer, c'est-à-dire de représenter les ensembles d'états de leurs boucles de contrôle par une formule de Presburger. Nous avions présenté les premiers résultats dans (Finkel & Leroux, FSTTCS '02) puis utilisé la même problématique dans [CI-136, CI-177, CI-180].

B.4 Vérification de systèmes à mémoire dynamique

La thématique de recherche sur les systèmes à mémoire dynamique s'est beaucoup développée ces dernières années suite au contrat avec EDF et à la participation au projet RNTL Averiles. Nous recensons ici les problèmes propres à cette thématique, ainsi que les différentes approches que nous avons suivies, issues du model-checking à base d'automate d'une part, et de la preuve de programmes en logique de séparation d'autre part.

B.4.1 Problématique

Les systèmes informatiques sont omniprésents tant dans les grands systèmes industriels que dans les petits produits commerciaux. Les programmes qui gèrent la couche la plus proche de l'interface matériele sont écrits dans des langages dits "bas niveaux". Ces langages offrent beaucoup de souplesse quant à la gestion par le programme de son espace mémoire, ce qui permet de nombreuses optimisations. En contrepartie, ces programmes sont particulièrement difficiles à analyser ne serait-ce même que par un humain, et malheureusement extrêmement vulnérables aux erreurs mémoires. Certifier de tels programmes est un problème ardu et néanmoins essentiel : outre les problèmes de sûreté de fonctionnement (la plupart des «bugs» de pilotes étant dûs à des fautes de manipulation mémoire), c'est aussi la sécurité et la confidentialité des programmes qui est concernée par les manipulations de pointeurs, certaines attaques reposant sur des dépassements de tableau pour modifier le comportement initial d'un programme. Ce problème est réputé ardu pour de nombreuses raisons : complexité de la sémantique de la gestion de mémoire, variété des instructions, etc. En ce qui nous concerne, même sur des classes restreintes de programmes, les difficultés majeures de l'analyse de l'allocation mémoire sont la quantité a priori non bornée de mémoire allouée durant l'exécution du programme, et l'*aliasing* : deux pointeurs différents peuvent référencer la même adresse mémoire durant l'exécution.

Dans le cas d'EDF, la demande principale était la détection d'accès hors bornes, et surtout de fuites mémoires, c'est-à-dire de cas où le programme considéré ne libère pas toute la mémoire qu'il devrait.

Plusieurs travaux scientifiques anciens ont porté sur les programmes à mémoire dynamique, notamment les analyses d'alias (points-to analysis) et les études de débordement de tableau. Notre travail s'est concentré sur une classe de programmes à mémoire dynamique dont l'étude théorique est beaucoup plus récente : les programmes manipulant des structures de données récursives allouées dynamiquement (listes, listes étendues, arbres, tableaux paramétrés, ...).

Notre but a été de donner des procédures de détection automatique de fautes mémoires. Par fautes mémoires, nous entendons notamment les violations mémoires, les fuites mémoires (pour les programmes sans ramasse-miettes, par exemple une des étude de cas d'EDF), ou encore la double désallocation. D'autres propriétés plus avancées peuvent aussi être examinées, et sollicitent une analyse plus complexe mais potentiellement plus précise. Parmi ces propriétés, citons les propriétés de forme — un programme prenant une liste ordonnée en entrée rend une liste ordonnée en sortie, les propriétés quantitative et temporelles — la longueur d'une liste n'augmente pas indéfiniment, etc. Par opposition avec les travaux plus anciens, ce cadre d´étude est souvent appellé «shape analysis» (analyse des formes mémoires générées par le programme).

Dans un premier temps, nous avons répertorié les différents outils existants. Aucun ne nous a semblé véritablement satisfaisant pour des raisons de difficultés d'utilisation, d'imprécision de l'analyse ou encore de fondements théoriques insatisfaisants. Il a donc été convenu avec EDF de développer une méthode originale, basée sur le cadre du model-checking symbolique.

Nous avons défini une représentation des programmes à analyser sous forme d'*automates à pointeurs* et une représentation symbolique du tas mémoire. Cette méthode nous a permis de gérer naturellement les problèmes exposés plus haut. Dans ce modèle simplifié, les propriétés intéressantes (absence de fuite mémoire, de *segmentation fault*) sont déjà indécidables pour des programmes manipulant trois pointeurs [CO-38].

B.4.2 Vérification par model-checking à base d'automates

La vérification par model-checking à base d'automates est un cadre théorique générique qui permet d'analyser de nombreux types de systèmes, et d'utiliser des langages de spécification — logiques temporelles, mu-calcul — ou des méthodes de recherche d'invariants de boucle — accélération, boucle d'abstraction-raffinement guidée par le contre-exemple (CEGAR) — et de bénéficier de nombreux outils développés pour l'analyse de tels modèles.

Dans notre cas, partant d'un système à pointeurs, on cherche donc à vérifier automatiquement les propriétés citées précédemment. Une approche consiste à abstraire l'ensemble infini des configurations accessibles et leur dynamique, si possible sans approximation, pour obtenir un système fini complètement analysable. La représentation symbolique utilisée pour abstraire les configurations peut elle-même faire intervenir des automates, de mots ou d'arbre, ou des graphes valués.

Analyse de forme par automates de mots et d'arbres L'approche de Peter Habermehl consiste à utiliser des automates de mots et d'arbres comme représentation symbolique de la mémoire. Dans des travaux antérieurs, il a proposé un codage sous forme de mots d'une mémoire constituée de segments de listes, cycliques ou acycliques. La dynamique du système est alors codée par des transducteurs, et la représentation symbolique d'un ensemble d'états mémoire est un langage régulier de mots codants des listes. Ce travail a permis, en suivant l'approche CEGAR, de vérifier automatiquement de nombreux programmes standards (insertion, effacement, renversement de liste, fusion, ...). Depuis cette approche a été étendue aux programmes manipulant des arbres [CI-64]. Elle repose alors sur un codage différent et une définition d'automates d'arbres étendus permettant de reconnaître des ensembles d'arbres subissant des contraintes non régulières (des contraintes arithmétiques sur le bon équilibrage de l'arbre) fort utiles dans

certains programmes de manipulation d'arbres. Une application immédiate de ce codage est la vérification automatique de programmes manipulant des arbres de recherche rouge-noirs (insertion, suppression).

Traduction des systèmes à pointeurs en systèmes à compteurs Une autre approche [CO-20, CO-4] consiste à utiliser des graphes valués pour représenter des ensembles d'états mémoires contenant des listes. L'abstraction permettant de se ramener à un nombre fini de représentations symboliques consiste à omettre les noeuds du graphe qui sont peu informatifs et à les remplacer par un compteur sur chaque segment de liste élémentaire permettant de retrouver combien de tels noeuds peu informatifs ont été omis. Cette approche permet de traduire un système à pointeurs en un système à compteurs, et d'utiliser des outils spécifiques aux systèmes à compteurs pour analyser des systèmes à pointeurs, tels que l'outil FAST, qui prend en charge seul l'accélération des boucles du système. Cette approche a conduit a un prototype, en cours de réimplémentation, qui a permis de traiter les programmes de manipulation de listes déjà vérifiés par l'approche précédente, plus des variantes faisant intervenir de façon cruciale des propriétés quantitatives sur la mémoire —par exemple une fusion de deux listes qui ne réussit que si elles ont la même longueur.

Contraintes arithmétiques dans les programmes manipulant des tableaux Les relations entre systèmes à pointeurs et systèmes à compteurs ne s'arrêtent pas aux programmes à manipulation de liste et aux aspects quantitatifs des formes mémoires manipulées. Un travail récent [CI-47] a en effet montré qu'il était possible d'utiliser les automates à compteurs pour modéliser des contraintes arithmétiques sur des tableaux et décrire de manière effective des invariants de boucle de programme manipulant des tableaux (unidimensionnels) de données.

B.4.3 Vérification par approches logiques

La vérification par les approches logiques, et notamment la logique de séparation, de programmes manipulant des pointeurs est une approche relativement différente de l'approche par model-checking à base d'automates, bien que les deux soient intimement liées. Elle consiste à définir un langage de spécification des états du programme — la représentation symbolique — puis des règles d'inférence pour des triplets de Hoare — les calculs de pre et de post liés aux opérations élémentaires du système.

Ces logiques font intervenir des connecteurs de formules plutôt originaux, dits spatiaux, qui permettent de traiter de propriétés de non aliasing. De ce fait, ces formalismes échappent aux méthodes traditionnelles de recherche de preuve automatiques existantes, par exemple pour la logique du premier ordre, du second ordre, ou la logique temporelle. Les opérateurs spatiaux surgissent naturellement dans le calcul de pre et post pour traiter le non aliasing, ils permettent le raisonnement local et modulaire. Cette approche constitue un cadre novateur pour spécifier les propriétés des programmes à pointeurs ; un prototype semi-automatique (`Smallfoot`, de Calcagno, O'Hearn *et al.*) a été récemment développé pour une version abrégée de ce type de formalisme : ce prototype vérifie qu'un programme annoté par le programmeur a été correctement annoté à chaque étape.

S'ils ont une sémantique simple et un intérêt sensible, les opérateurs spatiaux sont néanmoins difficiles à traiter, en particulier en combinaison avec de la récursion, elle aussi naturelle pour la prise en compte des boucles du programme, ou des structures mémoires récursives, ou encore en combinaison avec des opérateurs temporels, et les fragments pour lesquels on connaît des algorithmes exacts de décision sont très restreints. L'objectif de cet axe de recherche est donc de

proposer des algorithmes efficaces pour résoudre automatiquement les problèmes de satisfiabilité et model-checking pour ces logiques.

Propriétés temporelles et récursion en logique de séparation Cette approche [CI-82] consiste à développer un langage formel de spécification qui allie une composante temporelle liée à l'exécution du système et une composante propre à la gestion de la mémoire, à savoir la logique de séparation. Nous avons proposé une définition d'un tel langage formel de spécification combinant à la fois la description de plusieurs états mémoires successifs et leurs dépendances temporelles. Une fois ce langage hybride développé, l'étape suivante de ce travail consistait à concevoir de nouvelles techniques algorithmiques pour la vérification efficace de systèmes logiciels ayant une gestion dynamique de la mémoire pour des propriétés exprimables dans ce nouveau langage. Ici, il faut mettre au point des extensions de l'approche symbolique pour la vérification de systèmes infinis qui permettent d'explorer l'espace des configurations de façon optimale. Comme dans toute sa généralité, la vérification de systèmes à mémoire dynamique n'admet pas de solution algorithmique complète (indécidabilité théorique), le verrou scientifique qu'il s'agit de lever est la conception d'un langage de spécification concernant l'allocation dynamique de la mémoire qui admette des techniques algorithmiques efficaces. Des premières frontières entre décidabilité et indécidabilité ont pu être tracées, concernant à la fois le problème de satisfiabilité et de model-checking, et pour différentes classes de programmes — programmes sans mise à jour notamment — et pour différents fragments de la logique — avec ou sans arithmétique de pointeurs notamment. Une application des propriétés temporelles à la description de structures récursives a pu être envisagée. Enfin nous allons poursuivre l'étude des logiques de séparation et en particulier leur lien avec les logiques du second ordre pour lesquelles de nombreux résultats théoriques et outils logiciels sont disponibles.

Logique de séparation et concurrence Cette approche est développée par Etienne Lozes et Jules Villard, et consiste à étudier l'intérêt des opérateurs spatiaux dans la spécification de protocoles et de programmes concurrents. Un premier travail, mené par Jules Villard durant son stage de Master 2, à permis de définir et d'étudier une logique spatiale dédiée à un langage de modélisation de protocoles de communications sécurisées, le pi-calcul appliqué. On a pu établir que la logique définie donnait une caractérisation de plusieurs équivalences couramment utilisées dans la spécification de protocoles cryptographiques, telles l'équivalence statique ou l'équivalence observationnelle. Nous comptons étudier prochainement les implémentations efficaces de routines systèmes d'échange de messages, utilisées dans certains systèmes d'exploitations expérimentaux, et prévoyons de développer une approche logique pour établir la preuve automatique de la correction de certaines des optimisations utilisées.

Annexe C

Bilan scientifique détaillé : axe SECSI

Membres de l'axe SECSI

Responsables

- Hubert Comon-Lundh (PR ENS Cachan ; détaché à l'université de Tokyo depuis sept. 2007) ;
- Jean Goubault-Larrecq (PR ENS Cachan) ;

Membres permanents

- Michel Bidoit (DR CNRS, directeur du LSV jusqu'en 2005 ; directeur adjoint à la direction de la recherche au ministère de la recherche en charge des Maths-STIC, 2004-2007 ; directeur de l'UR Futurs de l'INRIA, juin-déc. 2007 ; directeur du CR Saclay-Ile-de-France de l'INRIA depuis jan. 2008) ;
- Stéphanie Delaune (CR CNRS, depuis oct. 2007) ;
- Stéphane Demri (CR CNRS)[1] ;
- Florent Jacquemard (CR INRIA) ;
- Steve Kremer (CR INRIA, depuis sept. 2004) ;
- David Nowak (CR CNRS, détaché à l'université de Tokyo depuis 2005)[2] ;
- Ralf Treinen (MCF ENS Cachan jusqu'en août 2007 ; en délégation INRIA au LSV, 2004-2006) ;

Membres non-permanents

- Myrto Arapinis (ATER ENS Cachan, 2007-2008) ;
- Laurent Mazaré (Postdoc CNRS, 2006-2007) ;
- Julien Olivain (Ingénieur associé, 2002-2005) ;
- Fabrice Parrennes (Ingénieur CDD, 2002-2003 ; 1/2 ATER ENS Cachan, 2003-2004) ;
- François-Régis Sinot (ATER ENS Cachan, 2007-2008) ;
- Angelo Troina (Postdoc ProNoBis, 2006-2007) ;

Doctorants

- Mathieu Baudet (2004-2007) ;
- Vincent Bernat (2002-2006) ;
- Sergiu Bursuc (depuis sept. 2006) ;

[1] Stéphane Demri a également participé à l'axe TEMPO et à l'axe INFINI. Il est maintenant membre de l'axe DAHU.

[2] David Nowak a également participé à l'axe INFINI.

– Elie Burztein (depuis sept. 2005) ;
– Jean-Loup Carré (depuis oct. 2006) ;
– Stéphanie Delaune (2003-2006) ;
– Pascal Lafourcade (2003-2006) ;
– Antoine Mercier (depuis oct. 2006) ;
– Benjamin Ratti (2004-2006, thèse abandonnée) ;
– Camille Vacher (depuis oct. 2007) ;
– Yu Zhang (2002-2005)

Évolution de l'axe de recherche

Les attaques contre les ordinateurs personnels (virus, spam, etc.), le paiement en ligne, les téléphones portables, ... devenant de plus en plus fréquentes, la sécurité est aujourd'hui un sujet crucial. La thématique principale de l'axe de recherche SECSI est l'utilisation de méthodes de vérification issues de la logique telles que la démonstration automatique, les automates d'arbres, les équivalences observationnelles dans les algèbres de processus et la théorie des domaines, appliquées aux protocoles visant à assurer la sécurité des systèmes. Plus particulièrement, le projet scientifique de SECSI tourne autour de la vérification de protocoles cryptgraphiques. Dans le précédent rapport quadriennal les bases des techniques de vérification automatiques de protocoles cryptographiques avaient déjà été décrites et des résultats prometteurs obtenus. Depuis ces travaux initiaux nous avons identifiés deux thèmes principaux autour desquels s'orientent nos recherches :

– Plus de *réalisme* : les modèles de protocoles utilisés pour la vérification de protocoles font généralement l'hypothèse dite du *chiffrement parfait* qui stipule qu'aucune information ne peut être obtenue sur un message en clair à partir de son chiffré, si l'on ne connait pas la clef de déchiffrement. Nous verrons que cette hypothèse très forte s'avère en fait peu réaliste dans de nombreuses situations. Nous affaiblirons cette hypothèse en considérant des extensions du modèle classique dit de Dolev-Yao. Nous travaillerons avec des modèles plus proches de la réalité, et nous obtiendrons ainsi des garanties de sécurité plus fortes.
– Plus de *propriétés* : les travaux précédents se sont focalisés sur les propriétés de secret et d'authentification. Ces propriétés sont des propriétés dites de *traces* ou d'accessibilité. Cependant des applications particulières (e.g. vote électronique, protocoles d'enchères, ...) doivent souvent vérifier des propriétés de sécurité plus subtiles. On peut citer par exemple, les propriétés d'anonymat et de résistance à la coercition dans le cadre des protocoles de vote.

Concernant le premier thème — plus de réalisme — nous avons réalisé que de nombreuses primitives cryptographiques présentent des propriétés algébriques qui peuvent être exploitées par un attaquant. Parmi les primitives les plus classiques, on trouve le « ou exclusif », l'exponentiation modulaire ou le chiffrement homomorphique. Nous avons donc étudié des modèles plus riches qui tiennent compte de telles propriétés et développé des méthodes de vérification pour ces modèles. Ces travaux sont décrits dans la section C.2. Une autre voie dans ce même sujet est l'obtention de résultats de correction *calculatoire* : on montre que des modèles symboliques peuvent être une abstraction correcte de certains modèles calculatoires plus précis, modélisant par exemple l'attaquant par des machines de Turing probabilistes polynomiales arbitraires. Ces résultats permettent donc d'obtenir des preuves automatiques de garanties de sécurité au niveau calculatoire qui étaient hors de portée des outils automatiques auparavant. Ces travaux sont décrits dans la section C.4. Finalement, de nombreux protocoles, par exemple ceux censés garantir l'anonymat, sont probabilistes. Nous avons donc initié une étude de modèles sémantiques qui arrivent à combiner probabilités et non-déterminisme (section C.5).

Concernant le deuxième point — plus de propriétés — nous avons étudié des protocoles tels que l'échange équitable, les protocoles de signature de contrat, les protocoles fondés sur des

mots de passe ou encore les protocoles de vote électronique. De nombreuses propriétés de ces protocoles s'expriment de façon naturelle au moyen d'équivalences observationnelles. Cette notion d'équivalence permet d'exprimer le fait que deux situations sont indistinguables. Cette notion est utilisée pour les propriétés d'anonymat et de résistance à la coercition dans le vote électronique, ainsi que la résistance aux attaques par dictionnaire dans les protocoles fondés sur des mots de passe. Ceci nous a entre autres amenés à étudier des techniques symboliques pour vérifier de telles équivalences. Ces travaux sont décrits dans la section C.3.

En plus de ces deux thèmes principaux nous avons continué l'étude de différentes techniques pour la vérification efficace des protocoles cryptographiques (section C.1) ; nous avons également développé des techniques formelles pour la sécurité des réseaux, en particulier la détection d'intrusion (section C.6).

C.1 Logique, automates d'arbre et sécurité

On peut affirmer que la logique — techniques de démonstration automatique, réécriture, automates d'arbres, lambda-calcul notamment — était, en 2000, le domaine d'expertise principal des membres fondateurs de l'axe SECSI. Les outils issus de la logique sont d'une expressivité et d'une efficacité remarquables en vérification de protocoles cryptographiques. Ceci est déjà vrai même lorsque l'on ne considère pas de théorie équationnelle non triviale comme à la section C.2. Nous présentons ici les résultats obtenus au LSV dans la voie, fondatrice, de la vérification automatique de protocoles cryptographiques via des techniques d'automates d'arbres, de démonstration automatique (résolution), et des techniques issues du lambda-calcul.

C.1.1 Automates d'arbres et clauses de Horn

Aux alentours de 2000, les membres de l'axe SECSI avaient exploré différentes formalisations de modèles de sécurité à la Dolev-Yao, fondées soit sur les automates d'arbres soit sur les contraintes ensemblistes. Ces formalismes peuvent en grande partie se traduire en ensembles de clauses de Horn, d'un format particulier, avec profit.

D'abord, il y a un avantage de clarté à le faire. La théorie de l'intrus parfait à la Dolev-Yao, notamment, s'écrit juste avec les clauses :

$$\begin{aligned}
\texttt{knows}(\{M\}_K) &\Leftarrow \texttt{knows}(M),\ \texttt{knows}(K)\\
\texttt{knows}(M) &\Leftarrow \texttt{knows}(\{M\}_K),\ \texttt{knows}(K)\\
\texttt{knows}(\langle M, N\rangle) &\Leftarrow \texttt{knows}(M),\ \texttt{knows}(N)\\
\texttt{knows}(M) &\Leftarrow \texttt{knows}(\langle M, N\rangle)\\
\texttt{knows}(N) &\Leftarrow \texttt{knows}(\langle M, N\rangle)
\end{aligned}$$

où $\texttt{knows}(M)$ signifie que M est un message connu de, ou déductible par l'intrus. La première clause exprime que l'intrus peut chiffrer tout message connu M par n'importe quelle clé connue K, la seconde exprime que l'intrus peut déchiffrer, et les trois suivantes que l'intrus peut former des couples $\langle M, N\rangle$ ou obtenir les composantes de tout couple.

Ensuite, le passage aux clauses de Horn permet parfois d'obtenir des résultats de décidabilité soit inconnus, soit d'une façon plus directe, en se fondant sur des résultats de complétude de raffinements de la résolution [RI-61], ou de résolution et de paramodulation [CI-118, RI-15] (voir ci-dessous pour cette dernière approche). Nous avons notamment utilisé cette approche pour vérifier des protocoles cryptographiques, en utilisant un format de clauses de Horn décidable qui s'apparentent aux contraintes ensemblistes avec tests d'égalités entre frères [In-36], puis à la classe $\mathcal{H}_1$ de Nielson, Nielson et Seidl. Nous avons en particulier donné une définition équivalente mais

plus simple de cette classe, montré que l'on pouvait effectivement calculer une approximation de tout ensemble de clauses (un cadre indécidable) sous forme d'un ensemble de clauses de $\mathcal{H}_1$ (qui est décidable), et donné une démonstration plus simple de la décidabilité en temps exponentiel de $\mathcal{H}_1$, à l'aide d'une forme de résolution ordonnée avec sélection et splitting sans splitting [RI-61]. Nous y avons aussi observé que la sous-classe $\mathcal{H}_2$, dont Nielson, Nielson et Seidl prétendait qu'elle était décidable en temps polynomial, était en fait EXPTIME-complète, et que toute extension de $\mathcal{H}_1$ permettant de coder l'égalité entre termes était indécidable.

Cette approche consistant à voir des classes d'automates d'arbres de plus en plus expressifs comme des sous-classes décidables de logiques en forme clausale a été explorée notamment dans le cadre de termes d'ordre supérieur, suivant Goubault-Larrecq (CSL'02), dont les contraintes ensemblistes d'ordre supérieur sont des ensembles de clauses formés à partir de formes restreintes de patterns à la Miller. Le travail de Ratti [Ra-41] a tenté de s'affranchir de l'utilisation de patterns, et a abouti à une petite sous-classe décidable, tout en montrant que toute extension suffisamment intéressante le long de cette voie était nécessairement indécidable. De façon peut-être plus pertinente, l'approche des automates en tant qu'ensembles de clauses a fait florès dans le cadre de théories algébriques, dont il sera question à la section C.2.

Nous avons d'autre part utilisé cette approche pour produire des preuves formelles, en Coq, de propriétés de sécurité, automatiquement [In-41] (une version améliorée et étendue de ce travail a d'autre part été récemment acceptée à la conférence Computer Security Foundations 2008). La remarque fondamentale est qu'une preuve de sécurité dans ce cadre, telle que découverte par un outil fondé sur la résolution comme notre propre outil `h1`, est un ensemble saturé de clauses, obtenu à partir d'un ensemble S de clauses représentant le protocole. Dans le cadre de $\mathcal{H}_1$, le format d'un tel ensemble saturé de clauses permet de le voir comme une forme d'automate d'arbres alternant, une représentation exponentiellement concise d'un automate d'arbres déterministe, c'est-à-dire d'un modèle fini $\mathcal{M}$. Il suffit ensuite d'écrire un model-checker pour vérifier que $\mathcal{M}$ est un modèle de S ; chaque étape du model-checker correspond à une étape de preuve par récurrence que l'on peut formaliser en Coq. Ceci a été implémenté dans l'outil `h1mc`.

L'utilisation de clauses de Horn, et spécifiquement de $\mathcal{H}_1$ et de sa procédure d'abstraction automatique d'ensembles de clauses de Horn générales, est aussi l'une des clés de notre analyseur CSur de propriétés de confidentialité de données sensibles dans du code écrit en C [CI-160]. L'intérêt est de pouvoir analyser des propriétés de secret, non plus sur des protocoles abstraits, mais sur de véritables implémentations. La principale difficulté est l'intégration de la gestion d'alias, notamment à travers les pointeurs de C, avec la modélisation d'intrus plus abstraits à la Dolev-Yao. Notamment, le fait que les intrus à la Dolev-Yao manipulent des messages abstraits, sous forme de termes, alors qu'un code C manipule des chaînes de bits, demande à faire le lien entre un message abstrait, et ses représentations concrètes possibles. Ceci est effectué au travers d'une relation de correspondance entre les deux, laquelle est axiomatisée, de même que la théorie de l'intrus, et qu'une forme légèrement généralisée d'analyse *points-to* pour les pointeurs, en clauses de Horn, dans un format qui s'abstrait bien en $\mathcal{H}_1$.

C.1.2 Automates d'arbres à mémoire avec visibilité

Les automates d'arbres à mémoire ont été introduits [RI-67] dans le cadre de la vérification de certaines classes de protocoles de sécurité. Ils généralisent à la fois les automates à piles (sur des mots) et les automates d'arbres avec test d'égalité entre sous-arbres. Si le problème du vide du langage reconnu est décidable pour ces automates, leur principale faiblesse est l'absence de bonnes propriétés de clôture, en particulier en ce qui concerne les clôtures par intersection et complémentaire.

Nous avons proposé [CI-88, RI-11] la généralisation des automates à piles avec visibilité de Alur

et Madhusudan à une sous-classe appelée VTAM des automates d'arbres de [RI-67] pour laquelle nous avons établi la clôture par toutes les opérations Booléennes (en montrant en particulier que ces VTAM peuvent être déterminisés). Les problèmes tels que le test du vide, de l'appartenance, de l'inclusion et l'universalité sont décidables pour les VTAM. En outre, nous avons proposé plusieurs extensions de VTAMs dont les transitions peuvent être contraintes par différentes sortes de tests portant sur le contenu des mémoires et qui permettent de caractériser des langages d'arbres binaire équilibrés, des arbres *red-black* ou des powerlists (listes de longueur 2^n stockés dans les feuilles d'un arbre binaire complet).

C.1.3 Relations logiques pour la création de noms et le chiffrement

Un autre style de modèles logiques permettant d'exprimer différents protocoles cryptographiques est fourni par le lambda-calcul, suivant une proposition faite par Pierce et Sumii en 2001. Ceci nécessite l'extension du lambda-calcul avec une primitive `new` de création de noms frais pour représenter la création de *nonces*, et diverses primitives de chiffrement et de déchiffrement. Ceci nécessite d'autre part de définir des critères suffisants, et suffisamment précis, impliquant l'équivalence observationnelle : ici, des notions de relations logiques, qui sont des familles de relations entre termes indexées par les types, et vérifiant certaines conditions de compatibilité.

Si l'ajout de `new` au lambda-calcul avait déjà été étudié par Pitts et Stark dans les années 1990, et si Pierce et Sumii avait déjà proposé des candidats de relations logiques, la bonne notion de relation logique restait à découvrir. Nous avons montré [CI-170] qu'une forme de relations logiques *lax*, aussi appelées relations prélogiques, à la Plotkin, Power et Sanella, caractérisait exactement l'équivalence observationnelle dans un lambda-calcul avec `new`, chiffrement et déchiffrement.

Il s'agit d'un lambda-calcul *monadique*, à la Moggi, où la monade est celle des noms frais, introduite par Stark. Dans le cas du lambda-calcul ordinaire, sans monade, déjà, les relations logiques lax sont malheureusement peu utilisables ; les relations logiques ordinaires sont en revanche complètes pour les types d'ordre un. (La complétude est trivialement fausse à l'ordre deux.) Comme la plupart des systèmes de transition sont codables à l'ordre un dans un lambda-calcul monadique, il est intéressant d'examiner si un tel théorème de complétude à l'ordre un s'étend aux lambda-calculs monadiques, en s'inspirant de l'approche de Goubault-Larrecq, Lasota, et Nowak (CSL'01, version longue [Ra-29] acceptée pour publication à MSCS, 2008) de définition de relations logiques pour des lambda-calculs monadiques généraux.

Il semble qu'un théorème de complétude s'appliquant à toute monade soit impossible à obtenir, et dépende de questions subtiles de définissabilité. Nous avons cependant obtenu des résultats de complétude à l'ordre un pour différentes monades importantes [CO-23, CO-3], incluant la monade de calculs partiels, d'exceptions, de calcul avec état, de continuations, et de choix non-déterministe. Ce dernier cas n'est complet que pour un sous-ensemble strict des programmes d'ordre un, dits d'ordre un faible. Ceci est cependant suffisant pour contenir comme corollaire le fait bien connu que la bisimulation forte caractérise l'équivalence observationnelle.

La thèse de Zhang Yu [TH-16] contient d'autres résultats non publiés, notamment la complétude dans un sous-cas des types d'ordre un des relations logiques en présence de `new` et de primitives cryptographiques, et la décidabilité de l'existence de telles relations logiques entre deux termes s, t pour un autre sous-cas des types d'ordre un. Rappelons que l'existence d'une telle relation logique entre s et t implique que s et t sont observationnellement équivalents, et que ceci permet de caractériser diverses propriétés de sécurité, dont le secret fort. La thèse [TH-16] a obtenu le prix de l'association franco-chinoise de recherche scientifique et technique (AFCRST), catégorie STIC, en 2006.

C.1.4 Automates d'arbres à contraintes pour la preuve par récurrence

En partie avec Adel Bouhoula (Sup'Com Tunis).

Nous avons mis au point une procédure pour la preuve automatique par récurrence dans des spécifications équationnelles pouvant contenir des des règles de réécriture avec des conditions et des contraintes [CO-29, Ra-34]. Les contraintes sont interprétées sur des algèbres de termes construits avec des symboles de *constructeurs* (représentant des valeurs de données), et peuvent exprimer l'égalité ou la diségalité syntaxique, des relations d'ordre ou encore des tests d'appartenance à un langage régulier d'arbres fixé. La possibilité offerte par notre procédure d'avoir des axiomes entre termes constructeurs permet de spécifier des structures de données complexes telle qu'ensembles, listes ordonnées, arbres, powerlists ... L'automatisation de la récurrence sur de telles structures de données est connu pour être un problème compliqué. Nous avons montré que la procédure est correcte et complète pour la réfutation.

Notre procédure est basée sur les grammaires d'arbres avec contraintes, qui sont utilisées dans les preuves à la fois comme un schéma de récurrence (pour la génération de sous-buts dans les étapes de récurrence) et comme outil de décision pour vérifier des critères d'effacement de sous buts. L'utilisation de grammaires comme schéma de récurrence permet la complète automatisation de la construction de tels schémas, ce qui différencie notre procédure d'approche dites par récurrence explicite où le schéma doit être spécifié par l'utilisateur.

Une implantation de cette procédure sous la forme d'un système de preuve automatique par récurrence est actuellement en cours au LSV.

Une variante de la procédure ci-dessus a été appliquée à la vérification formelle de protocoles de sécurité [CO-18, Ra-17, CO-9], par récurrence sur les traces d'événements correspondant à des envois et réceptions de messages par les participants du protocole et aux actions d'un attaquant. Deux aspects originaux de cette approche (par rapport à d'autres travaux sur la vérification de procotoles par récurrence) sont d'une part l'utilisation d'équations entre termes constructeurs pour spécifier des axiomes de destructeurs explicites (permettant de considérer des modèles de protocoles plus riches) et d'autre part l'utilisation de contraintes d'appartenance à des langages d'arbres réguliers pour spécifier une large famille de propriétés de sécurité à vérifier.

Cette approche pour la récurrence a également été appliquée à la vérification automatique de la propriété de *suffisante complétude* de spécifications algébriques [Ra-33, CO-16], qui exprime intuitivement que les fonction spécifiées sont complètement définies.

C.2 Propriétés algébriques

Un protocole est un nombre fixe de processus en parallèle qui peuvent être répliqués un nombre arbitraire de fois (ce sont les *sessions*) et instanciés par des agents. Certains agents sont supposés malhonnêtes ou compromis. L'intrus ou l'attaquant contrôle le réseau et l'on distingue généralement deux type d'attaquant. L'attaquant dit passif se contentant d'écouter les messages qui circulent sur le réseau et l'attaquant *actif*. Celui-ci peut également intercepter des messages et en émettre de nouveaux.

Compte tenu de la complexité du problème de la vérification des protocoles cryptographiques (ce problème est indécidable en générale), un certain nombre d'hypothèses simplificatrices ont été prises, permettant en particulier d'idéaliser les primitives cryptographiques en ne tenant pas compte de leurs propriétés algébriques. Cette hypothèse s'avère en fait peu réaliste dans de nombreux cas. Notamment, certaines propriétés algébriques des primitives cryptographiques sont explicitement utilisées dans certains protocoles. Citons par exemple, le protocole d'échange de

clefs de Diffie-Hellman, fondé sur l'existence d'une fonction de chiffrement $M \mapsto g^M$, telle que $(g^M)^N = (g^N)^M$. On peut également citer le protocole de porte-monnaie électronique, étude de cas fournie par France Télécom dans le cadre du projet RNTL PROUVÉ.

C'est pourquoi, après un recensement des besoins [RI-33], nous nous sommes intéressés à affaiblir cette hypothèse et nous avons étudiés l'automatisation des preuves (ou des recherches d'attaques) dans ce contexte. Selon les méthodes qu'elles mettent en oeuvre, ces résultats sont développés dans le paragraphe C.2.1 et C.2.2.

- Une des approches que nous avons développée est la mise en place d'algorithmes de décision spécifiques pour la résolution de contraintes dites de déduction (cf. paragraphe C.2.1). Cette approche, initié par Rusinowitch et Turuani en 2001 dans le cadre du chiffrement parfait, conduit lorsque le nombre de sessions est fixé à un modèle de protocoles fini en profondeur mais reste à branchement infini (l'attaquant pouvant construire des messages de taille arbitraire). Nous nous sommes intéressés à l'extension de ce résultat pour différentes théories équationnelles. Nous avons également amorcé l'étude de la notion d'équivalence observationnelle en présence de propriétés algébriques. Les résultats sont mentionnés ci-dessous et développées dans la section C.3.
- Nous avons également considéré des extensions des automates d'arbres par une théorie équationnelle et des contraintes d'égalités. Ces travaux sont des extensions naturelles aux travaux décrits dans la section C.1. L'avantage de cette approche est qu'elle permet de réutiliser (en les adaptant) des techniques existantes. Il est cependant parfois nécessaire de faire quelques abstractions (e.g. l'ordre d'exécution des différentes règles du protocole n'est pas toujours pris en compte). Ces abstractions sont sûres, mais peuvent parfois conduire à la découverte de fausses attaques (même si cela est rare dans la pratique).

Ces travaux ont été réalisés en liaison avec le projet RNTL PROUVÉ et l'ACI Sécurité Informatique « ROSSIGNOL ».

C.2.1 Contraintes de déduction

En partie avec Denis Lugiez (LIF, Marseille) et Véronique Cortier (LORIA, Nancy).

Nous nous sommes intéressés à l'extension du résultat de M. Rusinowitch et M. Turuani pour différentes théories équationnelles permettant de modéliser certaines propriétés algébriques des primitives cryptographiques, e.g. les propriétés induites par les méthodes de chiffrement par blocs, la propriété d'homomorphisme du chiffrement RSA, l'associativité et la commutativité (AC) d'opérateurs comme le *ou exclusif*, l'addition, la multiplication, ...

Dans cette voie, nous avons obtenu des algorithmes de décision pour différentes théories spécifiques relativement complexes. En particulier, un certain nombre de travaux (pour la plupart issus de la thèse de P. Lafourcade [TH-9]) ont permis de traiter des théories exploitant l'axiome d'homomorphisme ou de distributivité en présence d'opérateurs AC [CI-157, RI-47, CO-17, CI-79, CI-122, RI-28]. Une théorie particulièrement pertinente pour modéliser une étude de cas (un protocole du porte-monnaie électronique proposé par France Télécom R&D) a motivé ces travaux et a fait l'objet d'une étude [CI-83]. Compte tenu de la diversité des théories équationnelles nécessaires pour modéliser les primitives cryptographiques utilisées à l'heure actuelle, nous nous sommes très vite intéressés à mettre au point des résultats génériques. Un certain nombre de critères purement syntaxiques ont été dégagés et ont permis de mettre au point quelques classes décidables pour des théories ne mettant pas d'opérateurs AC en jeu [CI-165, CO-30, CI-133]. Ces résultats sont également développés dans les thèses de M. Baudet [TH-6] et S. Delaune [TH-11]. D'autre part, fort des études mener pour les théories liés à l'homomorphisme en présence d'opérateurs AC, nous avons pu faire dégager des conditions nécessaires pour traiter la classe des théories dites

monoidales (introduite par W. Nutt en 1990 dans le cadre des problèmes d'unification) [RI-17]. Ces conditions sont relativement précises comme le montre certains résultats d'indécidabilité [RI-35]. La thèse [TH-11] a reçu la mention "thèse remarquable" attribuée par France Télécom.

En parallèle, un programme de travail initié dans [IN-10] a également permis de dégager un certains nombre de concepts (e.g. la propriété de variants finis [CI-154]). Une première classe décidable capturant en particulier la théorie des signatures en aveugle utilisée dans les protocoles de vote (cf. paragraphe C.3) a été proposée [CI-49]. La thèse de V. Bernat est entièrement consacrée à l'étude de cette approche [TH-10]. L'utilisation de cette approche pour traiter de nouvelles théories (comme celle permettant la modélisation du protocole fourni par France Télécom dans le cadre du projet PROUVÉ) est encore à l'étude à l'heure actuelle, mais a déjà permis d'obtenir des résultats [Ra-23, CI-92].

Enfin, certaines propriétés de sécurité (e.g. secret fort, propriété d'anonymat) s'expriment en terme d'équivalence observationnelle et non en terme d'accessibilité (appelé ci-dessus déduction). En présence d'un attaquant *passif*, l'équivalence observationnelle est également appelée *équivalence statique* (nous y reviendrons dans la section C.3). Nous avons mis au point des algorithmes pour décider cette notion d'équivalence pour la classe des théories monoidales [CO-10, CI-61] et un résultat de combinaison permet de traiter des théories plus complexes [CI-66], e.g. la théorie comprenant le chiffrement et l'opérateur *ou exclusif*. Ces travaux ont été réalisés par S. Delaune au cours de son séjour post-doctoral dans l'équipe Cassis au LORIA à Nancy.

C.2.2 Automates d'arbres modulo une théorie équationnelle

En partie avec Michael Rusinowitch et Laurent Vigneron (LORIA, Nancy).

Comme il a été mentionné dans le paragraphe C.1.1, la vision des automates d'arbres en tant qu'ensembles particuliers de clauses de Horn se prête particulièrement bien à des extensions de la notion d'automates d'arbres par une théorie équationnelle et des contraintes d'égalités. Nous avons en particulier appliqué ceci au cas difficile des automates d'arbres modulo la théorie AC d'un ou plusieurs symboles associatifs et commutatifs afin de pouvoir traiter les protocoles du type Diffie-Hellman.

Automates bidirectionnels. Les bases théoriques avait été posées lors de la thèse de K. Neeraj Verma en 2003, et ont été complétées dans un article de journal [RI-27]. De nombreuses sous-classes, comme celle des automates alternants modulo AC, sont indécidables, mais nous avons montré que la vacuité de l'intersection de langages définis par des automates bidirectionnels (non alternants) était décidable. On notera que la sécurité de certaines adaptations de protocoles Diffie-Hellman en groupe se ramène à une telle question.

Les résultats de [RI-27] dépendent crucialement d'une construction, similaire à celle des arbres de Karp-Miller pour les réseaux de Petri et les VASS (i.e. systèmes d'addition de vecteurs) mais appliquée à une extension branchante des VASS, appelée ici BVASS [RI-58]. Ces derniers ont été introduit indépendamment par de Groot, Guillaume et Salvati sous le nom de VATA. Nous proposons un algorithme de décision pour les problèmes de vacuité, de bornage (boundedness), de couverture (coverability), ou d'accessibilité des états d'un BVASS [RI-58].

La haute complexité des algorithmes de décision sous-jacents à la théorie des BVASS, et plus généralement des automates bidirectionnels modulo AC, nous a d'autre part incités à chercher des procédures de décision approchées, et plus efficaces [RI-62]. L'originalité de notre approche est une procédure d'abstraction *au vol.* qui s'effectue pendant la recherche de preuve. Ce résultat a été implémenté dans l'outil MOP (MOdular Prover). L'outil arrive notamment à montrer que le protocole d'accord de clef en groupe GDH.2 (Group Diffie-Hellman 2) est sûr contre un adversaire passif, et ce jusqu'à 5 participants.

Automates d'arbres à contraintes d'égalités. Nous montrons qu'une forme généralisée du problème de l'appartenance pour les automates d'arbres est décidable pour les classes d'automates introduites, par saturation des ensemble de clauses avec une variante du calcul de paramodulation [CI-118, RI-15]. Ces automates d'arbres ont un bon potentiel d'application comme outil de décision pour la vérification de protocoles de sécurité, à la fois dans le cadre de la recherche d'attaques ou de la validation de protocoles, suivant une approche dite de *model-checking régulier* où un ensemble d'états infinis est représenté par un langage d'automate d'arbres. Ils permettent en effet de caractériser de manière exacte l'ensemble des messages circulant sur un réseau non sûr, avec un nombre fini de sessions et une version affaiblie d'attaquant (qui ne peut faire de copie de messages).

Les algorithmes développés dans ces travaux théoriques ont été implantés en Ocaml sous la forme d'une librairie TACE d'automates d'arbres et ont permis l'étude de plusieurs protocoles incluant en particulier un protocole récursif.

C.3 Protocoles complexes

Certains protocoles, pour être utilisables en pratique, doivent assurer des propriétés de sécurité complexes et ne peuvent pas être traités par les travaux mentionnés précédemment (ces derniers portaient essentiellement sur la vérification de propriétés du type accessibilité). Citons quelques exemples :

- Les protocoles reposant sur l'utilisation de mots de passe utilisateur sont particulièrement vulnérables aux *attaques par dictionnaire*. Un utilisateur choisi souvent un mot de passe facile a retenir et donc facile a deviner ! Les modèles d'attaquant standards ne capturent pas ce type d'attaques, il est donc important de mettre en place des modèles plus fins et les procédures de décision correspondantes.
- Les protocoles de vote doivent satisfaire de nombreuses propriétés (e.g. anonymat, propriété de sans-reçu, résistance à la coercition, ...). Là encore, ces propriétés ne s'expriment pas en terme d'accessibilité, mais en terme d'indistinguabilité.
- Les protocoles d'échange équitable, e.g. protocole de signature de contrat, font intervenir des propriétés d'*équité*. Elles doivent assurer qu'aucun des signataires du contrat ne peut avoir, à l'une des étapes de la signature, un avantage sur les autres. Cette propriété est assez subtile dans la mesure où il n'y a pas d'un côté les agents honnêtes et de l'autre les malhonnêtes. Ici, chaque participant se méfie de l'autre !

Nous nous sommes intéressés à ces différents problèmes en proposant des modélisations de ces propriétés de sécurité d'un genre nouveau et en développant dans un deuxième temps des procédures de décision adaptées. Ces travaux sont développés dans les paragraphes C.3.1, C.3.2 et C.3.3.

C.3.1 Protocoles avec mot de passe

De nombreux protocoles font intervenir un mot de passe choisi par un utilisateur et sont donc vulnérables aux attaques par dictionnaire. Ce type d'attaque, appelée attaque par dictionnaire, consiste pour l'intrus à deviner quel mot a été utilisé en testant tous les mots d'un dictionnaire donné jusqu'à ce qu'il reconnaisse une information caractéristique. Plusieurs modèles accompagnés de procédures de décision ont été proposés pour capturer et automatiser la rechercher de telles attaques [CI-178, RI-53, CI-133]. Une justification calculatoire a également permis de valider la définition d'attaques par dictionnaire utilisé dans [CI-133]. Ce point est développé dans le paragraphe C.4.

C.3.2 Protocoles de vote électronique

Avec Mark Ryan (Université de Birmingham, Royaume-Uni).

Les protocoles de vote électronique et les machines à voter sont déjà très largement déployés. Pourtant leur fiabilité est sérieusement mise en cause dans de nombreuses études. À l'heure actuelle, il n'y a même pas d'accord général sur une modélisation des propriétés à satisfaire et certaines d'entre elles semblent même contradictoires. Ces propriétés sont généralement décrites informellement, ce qui conduit à des raisonnement non rigoureux et problématiques.

Dans un premier temps, une étude cas sur un protocole de vote de la littérature a été menée et a permis d'identifier les lacunes des techniques et outils existants [CI-156]. Nous nous sommes ensuite intéressés aux problèmes liés à la modélisation des propriétés du type anonymat et nous avons proposé une modélisation élégante des propriétés d'anonymat en terme d'équivalence observationnelle [CO-22, CO-19, CI-121]. Le problème de vérifier ces équivalences est indécidable en général. Cependant, nous avons mis au point une notion de bisimulation symbolique décidable pour les théories dites sous-termes convergentes (dans le cadre d'un nombre borné de sessions) [CO-8, CI-56]. La notion choisie est suffisamment fine pour pouvoir valider les études de cas.

L'axe SECSI est particulièrement actif, encore aujourd'hui, sur ce sujet : un projet ARA SESUR AVOTÉ qui a débuté en janvier 2008 est entièrement consacré à cette thématique.

C.3.3 Protocoles d'échange équitable

En partie avec Rohit Chadha (Université de l'Illinois, États-Unis), Aybek Mukhamedov (Université de Birmingham, Royaume-Uni), Eike Ritter (Université de Birmingham, Royaume-Uni) et Andre Scedrov (Univerity de Pennsylvanie, États-Unis)

Les protocoles d'échange équitable sont des protocoles cryptographiques qui permettent à plusieurs entités d'échanger des biens électroniques, par exemple un paiement, de façon à ce que chaque participant envoie son bien si, et seulement si, il reçoit le bien attendu en retour. Un exemple d'échange équitable est celui des protocoles de signature de contrat numérique où les biens à échanger sont des signatures numériques sur un contrat.

Nous avons analysé plusieurs protocoles de signature de contrat [CO-60⋆, CI-290⋆, RI-52, CI-144]. La complexité de ces protocoles rend une analyse manuelle rigoureuse impossible. De plus, une automatisation efficace n'a pu être obtenue que par la mise en place de nouveaux résultats théoriques. Nous avons découvert des erreurs fondamentales dans le protocole de Garay-MacKenzie, publié en 1999. L'étude du protocole proposé par Franklin et Tsudik [CI-144] a nécessité une modélisation fine afin de prendre en compte les propriétés algébriques de l'exponentiation modulaire. Il a été nécessaire d'étendre le modèle de Dolev-Yao classique de façon similaire aux travaux présentés dans le paragraphe C.2.

Nous avons également réalisé quelques études de cas sur d'autres types de protocoles (e.g. protocoles d'enchères) [Ra-19, Ra-15] et nous nous sommes intéressés à des types d'attaques particuliers [CI-152, CI-75]. Par exemple, dans [CI-75], nous proposons une modélisation du « key conjuring ». Cette attaque consiste à utiliser un nombre aléatoire à la place d'un chiffrement. Ce nombre aléatoire a une probabilité non négligeable d'être interprété comme un chiffré valide et de permettre le déclenchement d'une fonctionnalité du protocole pourtant interdite à l'attaquant. Les protocoles cryptographiques utilisés dans les distributeurs de billets sont particulièrement à ce type d'attaques. Nous reviendrons sur ce point dans la partie projet.

C.4 Résultats de soundness

Historiquement, deux approches différentes et indépendantes ont été développées pour analyser et prouver correct des protocoles de sécurité.

- La première approche est appelée l'approche calculatoire. Dans cette approche l'adversaire est une machine de Turing probabiliste polynomiale arbitraire. Les primitives cryptographiques sont des algorithmes polynomiaux et leur sécurité est exprimée en termes de probabilité. L'avantage de cette approche est que le modèle est précis et donc les garanties de sécurité très fortes. Par contre, les preuves sont généralement très complexes et difficilement automatisables.
- La deuxième approche est appelée approche symbolique. Il s'agit de l'approche classiquement étudiée dans l'axe SECSI qui se met à un niveau d'abstraction plus élevé. L'avantage de cette approche est que les preuves de sécurité peuvent souvent être automatisées. Les abstractions faites dans cette approche sont par contre difficile à justifier.

En 2000, Abadi et Rogaway ont démontré un premier résultat de correction de l'approche symbolique par rapport à l'approche calculatoire, c'est-à-dire un résultat montrant que si un protocole est prouvé correct dans le modèle symbolique, il est également correct dans le modèle calculatoire sous des hypothèses standard sur les primitives cryptographiques. Ce type de résultat permet d'avoir le meilleur des deux approches : des preuves automatiques avec des garanties de sécurité dans un modèle calculatoire. En particulier, ce premier résultat ne tient que dans un modèle simple qui considére un adversaire passif (l'attaquant ne peut pas intéragir avec le protocole en envoyant des messages sur le réseau) et comme primitives cryptographiques le chiffrement symétrique et la paire (concaténation de messages).

Nous avons commencé à travailler sur cette thématique en 2004. Un premier pas pour rapprocher les deux approches à consister a étendre les modèles Dolev-Yao par des calculs probabilistes polynomiaux [CO-34, RI-32]. Ensuite, nous avons montré des résultats pour des adversaires plus élaborés et nous avons également considéré des primitives plus complexes en prenant en compte leurs propriétés algébriques (e.g. ou exclusif).

Adversaire passif.
En partie avec Martín Abadi (Université de Californie, Santa Cruz et Microsoft Research, Silicon Valley, États-Unis), Véronique Cortier (LORIA, Nancy) et Bogdan Warinschi (Université de Bristol, Royaume-Uni).

Nous avons défini un cadre pour comparer des implantations cryptographiques et leur idéalisations symboliques pour plusieurs notions de sécurité en présence d'un adversaire passif [CI-145]. Ces travaux ont permis de montrer la correction de théories équationnelles donnant ainsi une justification calculatoire aux travaux effectués par ailleurs et développés dans le paragraphe C.2. Nous avons obtenu de nouveaux résultats de correction pour l'« ou exclusif » le chiffrement de blocs avec listes [CI-145] et la résistance aux attaques par dictionnaires [CI-127] (voir aussi le paragraphe C.3). Ces résultats sont également développés dans la thèse de M. Baudet [TH-6]. Nous avons également montré un résultat pour une théorie qui modélise des couplages bilinéaires [CO-15]. Les couplages bilinéaires sur les courbes elliptiques ont récemment donné lieu a de nombreux protocoles très efficaces. Ce dernier résultat a été récompensé par le prix du meilleur papier au « Workshop on Issues in the Theory of Security (WITS'07) ».

Adversaire adaptatif.
Nous avons ensuite étendu le modèle de [CI-145] afin de considérer des adversaires adaptatifs, strictement plus puissants que les adversaires purement passifs [CI-69, CO-12]. Dans ce cadre, nous

avons obtenus des résultats pour des théories équationelles telles que le chiffrement symétrique, l'exponentiation modulaire, l'ou exclusif ainsi que des théories combinant chiffrement symétrique avec l'exponentiation modulaire et l'ou exclusif. Ces derniers résultats utilisent une nouvelle technique de preuve de combinaison qui permet de réutiliser des preuves de correction sur des théories individuelles. Ils définissent également un modéle symbolique pour analyser des protocoles d'échange de clés dynamiques dont ils montrent la correction calculatoire grâce aux résultats obtenus.

Adversaire actif.
Avec Véronique Cortier (LORIA, Nancy), Ralf Küsters (Université de Trier, Allemagne) et Bogdan Warinschi (Université de Bristol, Royaume-Uni).

Nous proposons un critère formel de sécurité qui est correct en présence d'un adversaire actif par rapport à une définition de secret calculatoire basée sur la notion d'indistinguabilité [CI-96]. Le secret symbolique en terme de déduction n'est en effet pas suffisant en général d'un point de vue cryptographique, en particulier en présence de fonctions de hashage. Nous présentons donc un critère de secret plus adéquat qui caractérise de façon exacte la définition calculatoire du secret pour des protocoles qui utilisent des fonctions de hashage et du chiffrement asymétrique : les protocoles qui respectent le secret sont sûrs dans le modèle calculatoire alors que tout protocole qui le viole mène directement à une attaque. De plus, ce critère est décidable en temps NP. Le résultat est basé sur des hypothèses calculatoires standard pour le chiffrement et modélise les fonctions de hashage comme des oracles aléatoires.

Les travaux ont été effectués en grande partie dans le cadre du projet ARA SSIA Formacrypt. Un projet de collaboration franco-japonais CNRS JCT/ICT vient également de démarrer en janvier 2008 sur cette thématique. Ces travaux ont aussi donné lieu à un exposé invité [IN-2] au « Workshop on the Interplay of Programming Languages and Cryptography », associé à la conférence TGC 2007. De plus, S. Kremer est un des co-fondateurs de la série de workshops « Formal and Computational Cryptography (FCC) » et en a été le président du comité de programme avec V. Cortier en 2006 [Ed-10].

C.5 Probabilités et non-déterminisme

En partie avec Catuscia Palamidessi (LIX, École Polytechnique, Saclay) et Roberto Segala (Université de Vérone, Italie) dans le cadre de l'ARC INRIA ProNoBis (Probabilities, Non-determinism, and Bisimulations for Security).

Un nouveau thème au LSV est celui de l'étude des modèles sémantiques combinant choix non déterministe et choix probabiliste. Les modèles de sécurité à la Dolev-Yao ou fondés sur les algèbres de processus ignorent les aspects probabilistes, et reposent entièrement sur des choix non-déterministes pour coder les différentes exécutions possibles. Ceci est légitime dans bon nombre de cas, ne serait-ce que parce que l'on peut souvent remplacer un tirage au hasard par un choix non déterministe sans perdre d'information importante. Les résultats développés dans le paragraphe précédent (paragraphe C.4) illustrent très bien ce propos.

Cependant, il existe des protocoles et des propriétés pour lesquelles il est nécessaire de raisonner sur les probabilités directement. On citera notamment les questions d'anonymat, illustré notamment dans le protocole des cryptographes à dîner de Chaum, où un codage non probabiliste donne une information de sécurité insuffisante. Dans un tel protocole probabiliste, cependant, il reste un certain nombre de choix non déterministes, comme celui de l'ordre dans lequel les participants honnêtes décident d'interagir entre eux. Les deux formes de choix doivent donc

cohabiter. Dans un tel cadre, la bonne notion de bisimulation, qui impliquera, voire sera équivalente à l'équivalence observationnelle, reste à définir. Rappelons que l'équivalence observationnelle est l'un des moyens classiques de définir diverses propriétés de sécurité fortes. Dans le cas purement non déterministe, les notions de bisimulations fortes ou faibles sont classiques. Dans le cas purement probabiliste, les bonnes notions viennent de Larsen et Skou (1991), et correspondent à une notion de *lumping*, utilisée indépendamment en vérification (paragraphe A.3.5).

Nous avons entrepris une étude fondamentale de modèles sémantiques mêlant à la fois non-déterminisme et probabilités, fondés tant sur la notion de capacités due au départ à G. Choquet (1953-54) et développée ensuite en économie par Gilboa ou Schmeidler, que sur la notion de prévision due à Walley (1991) en finances. Le parti pris a été d'opter pour une approche topologique, qui permette de fournir des modèles allant au-delà des espaces d'états finis. En particulier, tout cpo muni de sa topologie de Scott forme un espace topologique, et nos modèles y ont alors un sens.

La publication [RI-66] a constitué un échauffement dans notre approche de l'étude des probabilités (sans non-déterminisme) sous un angle topologique. Nous y montrons le résultat remarquable qu'une valuation continue (le pendant topologique d'une mesure) est approximable par une famille dirigée de valuations s'exprimant simplement comme des combinaisons linéaires de valuations de Dirac, si et seulement si cette valuation continue s'étend à tous les clos par le haut, pas seulement les ouverts, de l'espace sous-jacent.

C.5.1 Capacités, crédibilités, plausibilités

Nous avons commencé par étudier des modèles fondés sur la notion de capacités, et plus particulièrement de crédibilités et de plausibilités continues [CI-77]. Les premières fournissent un modèle correct et complet d'un coup de choix probabiliste suivi d'un coup de choix non déterministe démoniaque — intuitivement, dans lequel l'adversaire effectue les tirages non déterministes, de sorte soit à éviter un objectif d'accessibilité, soit à minimiser notre gain. Les secondes sont similaires, mais le non-déterminisme y est angélique. Nous l'avons appliqué à la définition de modèles de systèmes de transitions mêlant les deux types de choix, décrivant une forme de jeu stochastique à deux joueurs sur des espaces infinis, et nous avons montré qu'une notion adéquate de similarité (plus grande simulation) était capturée par une logique modale simple.

C.5.2 Prévisions

Les capacités ont néanmoins un défaut : elles ne se composent pas. Autrement dit, une suite de coups alternant entre choix probabilistes et choix déterministes n'est pas descriptible à l'aide d'une capacité. Nous avons résolu ce problème en passant, via un théorème de représentation, des capacités à certaines fonctionnelles d'ordre deux appelés prévisions, et qui en un sens calculent des moyennes généralisées (non nécessairement linéaires) de fonctions données en argument. Ces fonctionnelles sont une variante des prévisions basses cohérentes de Walley, et nous les avons nommées prévisions basses continues dans le cas démoniaque, prévisions hautes continues dans le cas angélique. Une autre notion, celle de fourchette, modélise le mélange de choix probabiliste et non déterministe chaotique. Ces outils fournissent notamment une sémantique élégante de langages fonctionnels d'ordre supérieur mêlant choix probabiliste et non déterministes, dans un style de passage à la continuation [CI-68]. Différentes tentatives avaient été effectuées pour trouver une telle sémantique, et la sémantique des prévisions est probablement l'une des plus claires.

En ce qui concerne les autres propositions de sémantiques mêlant tirage probabiliste et non déterministe, une en particulier attire notre attention, qui est due indépendamment à Mislove d'une part, à Tix, Keimel et Plotkin d'autre part. Nous avons montré que, sous des hypothèses

relativement bénignes, leurs constructions et les nôtres sont isomorphes [CI-45]. L'isomorphisme est en particulier défini à l'aide d'une généralisation de la notion de cœur d'une capacité au cas des prévisions ; le fait que le cœur d'une capacité monotone convexe soit non vide est une notion fondamentale en économie.

Il est relativement facile d'étendre la notion de similarité des systèmes de transition fondés sur les capacités à ceux fondés sur les prévisions. Plus intéressante est la notion de distance de (bi)simulation, qui permet de quantifier à quel point deux programmes, processus, ou systèmes de transitions, ont des comportements similaires ; deux processus à distance zéro étant (bi)similaires. Nous avons montré comment définir une telle notion de distance de simulation [CI-46], étendant ainsi des travaux antérieurs de Ferns, Panangaden, et Desharnais entre autres, et établi un certain nombre de théorèmes montrant la grande robustesse de la notion.

C.5.3 Le pi-calcul appliqué probabiliste

Parallèlement, nous avons défini une extension du π-calcul appliqué incluant les choix probabilistes [CI-57], et démontré que l'équivalence observationnelle y était équivalente à la bisimilarité étiquetée. Nous avons utilisé ce formalisme pour vérifier une propriété de sécurité d'un protocole de transfert oublieux d'une donnée parmi deux (1-out-of-2 oblivious transfer), ainsi que de l'anonymat dans un protocole d'onion routing.

On notera finalement que notre parti pris de considérer des modèles contenant non seulement des espaces finis mais, au-delà, des cpo et des espaces topologiques plus généraux, nous permet désormais d'envisager de formaliser des systèmes informatiques manipulant des variables réelles en particulier. Nous avons notamment été en contact avec d'autres partenaires académiques et industriels pour appliquer ces constructions à l'étude, notamment, des implémentations de contrôleurs PID ou éventuellement plus complexes, les probabilités étant nécessaires pour traiter des erreurs d'arrondi présentes dans les calculs en virgule flottante. Ceci sort naturellement, formellement, du cadre de SECSI, mais non de celui du LSV.

C.6 Sécurité système et réseau

C.6.1 Détection d'intrusions

En sécurité réactive, nous avons développé un détecteur d'intrusions temps réel, multi-sources, multi-événements appelé Orchids [CI-147, In-11], basé sur des techniques de model-checking adapté à l'analyse en ligne. (Voir aussi le paragraphe A.1.3 : il s'agit d'un problème de model-checking d'un chemin, ici le *log*, c'est-à-dire la liste des événements reçus par le détecteur.)

Orchids est un système d'analyse modulaire qui analyse à la fois des évenements systèmes et réseaux. Son moteur d'analyse utilise des techniques d'interprétation abstraite qui lui permettent d'analyser des flux d'événements réels sur plusieurs mois, tout en gardant trace de tous les débuts d'attaques possibles en cours [IN-4]. Ces performances le rendent apte à être déployé pour la surveillance de grand réseaux.

Les communications chiffrés sont un véritable challenge pour les détecteurs d'intrusions, car l'analyse du contenu des paquets est indisponible. Il est pourtant vital de pouvoir détecter les attaques sur ces flux car ils contiennent des informations sensibles. Nous avons donc developpé une technique de détection d'attaques, typiquement par buffer overflows, contre les flux chiffrés [Ra-26]. Celle-ci se fonde sur la variation d'un estimateur de l'entropie en cas d'attaque. Cet estimateur, que nous avons appelé l'estimateur de Paninski, est remarquablement précis, comme nous l'avons montré aussi bien théoriquement que par l'expérience.

C.6.2 Prédiction de vulnérabilités

Un nouveau thème de recherche en détection d'intrusions au LSV est l'utilisation de méthodes formelles pour l'analyse proactive de la sécurité des réseaux. Celle-ci permet d'améliorer la sécurité des réseaux en établissant et en analysant les scénarios d'attaques les plus dangereux a priori. Ce type d'analyse est donc complémentaire des méthodes réactives (e.g. pare-feu ou système de détection d'intrusion) visant contrer les menaces existantes. Notre analyse proactive se fonde sur la théorie des jeux, et plus particulièrement sur les jeux temporisés. Ces derniers permettent d'analyser l'interaction des intrus et des administrateurs avec le réseau en tenant compte de la dimension temporelle de leur actions. Ce type de jeux est très utilisé dans l'axe TEMPO, où l'on parlera plutôt de système que d'administrateur, et d'environnement plutôt que d'intrus (section A.2).

Les jeux temporisés classiques ne permettent pas de tenir compte de la topologie sous-jacente et de l'etat initial du réseau. Nous avons donc créé un nouveau type de jeu, appelé *anticipation game*. Ces jeux sont basés sur les *timed automaton games* de de Alfaro, Faella, Henzinger, et Majumdar, et nous utilisons une variante de la logique TATL pour vérifier diverses propriétés de sécurité et de résilience des réseaux [CO-6, CI-53]. Nous avons depuis intégré des notions de coûts et de récompenses, et raffiné notre modèle et nos algorithmes pour détecter les attaques les plus efficaces et les moins coûteuses, ainsi que les mesures de réparation les plus efficaces et les moins coûteuses [CI-5]. (Nous reverrons les automates temporisés à coûts — sans l'aspect des jeux — à la section A.1.2.)

Nous nous sommes également intéressés à la découverte automatique des topologies réseaux sous-jacentes nécessaires à notre analyse proactive. Une des difficultés majeures de cette activité de découverte est de pouvoir détecter les machines qui partagent la même adresse IP, comme dans le cas des NAT[3]. Nous avons pour ce faire developpé une technique d'analyse [CO-5] fondée sur les timestamps, qui permet d'accroître de manière significative le taux de détection de ce type de machines. Elle permet en particulier de détecter certains type de systèmes (Linux) qui jusqu'a présent échappaient aux méthodes existantes.

Nous avons ensuite combiné l'utilisation de l'entropie des flux avec d'autres discriminateurs et à de l'analyse de payload pour développer une méthode d'identification probabiliste des flux réseaux difficiles à classifier [CI-40], comme les protocoles pair à pair (P2P) ou les canaux subliminaux (covert channels).

C.7 Divers

Certaines des activités des chercheurs du LSV les mènent à obtenir des résultats en-dehors des sentiers des projets de recherche du laboratoire, pour diverses raisons. Mentionnons quelques-uns de ces résultats difficilement classables.

Goubault-Larrecq avait étudié en son temps la théorie de la preuve de la logique modale intuitionniste S4, et montré avec Goubault (Homology, Homotopy and Applications, 2003) que le calcul des termes de preuve de cette logique avait un contenu géométrique, en termes d'ensembles simpliciaux. Ce résultat a été présenté de nouveau en 2004 [In-35], et les techniques de démonstration utilisées, fondées sur des définitions d'extensions de relations logiques aux cas comonadique, ont été la source des techniques explorées dans les travaux mentionnés au paragraphe C.1.3.

Goubault-Larrecq s'est d'autre part intéressé à la logique linéaire, et plus particulièrement à la géométrie de l'interaction, à la suite d'un exposé de Colin Stirling au LSV en 2003, où Goubault-Larrecq a réalisé qu'un certain nombre de constructions de la démonstration du théorème de

[3]Network Address Translation

Sénizergues (décidabilité de l'équivalence de grammaires algébriques déterministes) étaient des constructions typiques d'espaces cohérents à la Girard. Ceci l'a mené à creuser les notions de modèles catégoriques de la logique linéaire et la géométrie de l'interaction [Ra-20]. Outre la définition générale de la catégorie de Danos-Régnier $\mathcal{DR}(M)$ d'un monoïde inversif linéaire M, en tant que cadre catégorique de la géométrie de l'interaction, le résultat principal est que s'il est facile de faire de $\mathcal{DR}(M)$ un modèle du fragment multiplicatif, il est impossible d'interpréter quelque construction additive ou exponentielle que ce soit, pour aucun M non trivial. En revanche, une construction due à Joyal et Hu, dite de complétion par cohérence, permet de construire des modèles catégoriques de toute la logique linéaire — y compris les unités — par-dessus $\mathcal{DR}$ (M). Ce travail a été soumis au numéro spécial de la revue TCS en l'honneur des soixante ans de Jean-Yves Girard.

Dans le cadre du projet européen EDOS, Treinen a développé des techniques fondées sur la démonstration automatique, et qui permettent de vérifier la cohérence de distributions de grands logiciels, comme les diverses versions de Linux. Ceci présente de nombreuses difficultés [CI-126] : outre les circularités dans les dépendances entre packages, il faut aussi tenir compte des incompatibilités entre certaines versions de packages, l'évolution dans le temps des versions, des dépendances et des incompatibilités notamment. Malgré la taille impressionnante des collections de fichiers que doivent maintenir les distributeurs Linux (comme Debian, dont Treinen est un mainteneur actif), nous avons pu développer des outils logiques permettant d'attaquer ces problèmes ardus [CI-110].

Kremer s'est également intéressé aux mathématiques du jonglage. Depuis la fin des années 80 on sait que des motifs de jonglage peuvent être décrits par des chaînes d'entiers avec des propriétés combinatoires fascinantes qui ont été étudiées par de nombreux mathématiciens et informaticiens. Dans [RI-94⋆], nous étudions les motifs de jonglage d'un point de vue reconnaissance de motifs. Inspiré par les algorithmes de reconnaissance de motifs basés sur les convolutions nous proposons un algorithme efficace pour trouver des transitions entre états de jonglage. Cet algorithme s'avère être un outil pratique dans la conception expérimental de grands motifs de jonglage. Nous donnons également une formule pour calculer le nombre de transitions d'une longueur donnée entre deux états de jonglage.

C.8 Vulgarisation

Si elle ne contribue pas directement à l'accroissement des connaissances, la vulgarisation, au sens large, participe néanmoins pleinement à la mission du chercheur de faire connaître les résultats de la science.

Dans ce cadre, on peut citer notamment les exposés invités de Goubault-Larrecq au Workshop on Classical and Quantum Information Security [In-19] sur les méthodes formelles de vérification de protocoles cryptographiques classiques (non quantiques), ou l'article d'encyclopédie [Ch-6] sur les modèles et méthodes de preuve pour la sûreté et la sécurité informatiques.

C.9 Outils

1. La bibliothèque PROUVÉ. Il s'agit d'une bibliothèque fournissant des fonctions de transformations de spécifications de protocoles cryptographiques écrits dans le langage PROUVÉ en syntaxe abstraite, et effectuant une analyse statique de la spécification.
   Url : `http://www.lsv.ens-cachan.fr/prouve/libparser/index.html`
   Licence : LGPL.
   Réalisé en Objective Caml.
   Auteur : Ralf Treinen.

2. SPORE : The Security Protocols Open Repository. Le but de cette page Web est de continuer, sur la toile, le travail initié par Clark et Jacob en 1997, en mettant à jour leur base de protocoles de sécurité. Initialement développée lors du projet RNTL EVA (2000-2003).
   Url : `http://www.lsv.ens-cachan.fr/spore/`
   Licence : none.
   Auteurs : Florent Jacquemard, Ralf Treinen, Hubert Comon-Lundh, divers autres participants.
3. Orchids. Un système de détections d'intrusions efficace, en ligne, temps réel, multi-événements, multi-sources, fondé initialement sur des idées de model-checking d'un chemin [CI-147, IN-4].
   Url : `http://www.lsv.ens-cachan.fr/orchids/`
   Licence : Cecill 2 (GPL).
   Réalisé en C.
   Auteurs : Julien Olivain, Jean Goubault-Larrecq.
4. Net-Entropy. Un vérificateur d'entropie pour les connexions réseau chiffrées [Ra-26]. L'un des senseurs originaux d'Orchids.
   Url : `http://www.lsv.ens-cachan.fr/~olivain/net-entropy/`
   Licence : GPL.
   Réalisé en C.
   Auteur : Julien Olivain.
5. EvtGen. Un simulateur d'événements discrets générique fondé sur les chaînes de Markov. Utilisé pour construire des sources d'événements artificielles mais réalistes pour tester les systèmes de détection d'intrusions, Orchids en particulier (voir plus haut). Développé au sein du projet RNTL DICO (2002-2005).
   Url : `http://www.lsv.ens-cachan.fr/~olivain/evtgen/`.
   Licence : spécifique (similaire à CSur, ci-dessous).
   Réalisé en C.
   Auteur : Julien Olivain.
6. Csur. Un analyseur statique de programmes C, dont le but est de détecter des fuites d'informations secrètes dans un modèle adéquat de sécurité, à la Dolev-Yao [CI-160], tout en tenant compte des difficultés posées par le suivi des flux d'information dans de vrais programmes, en particulier à travers l'arithmétique des pointeurs. L'outil produit des clauses qui sont ensuite fournies à `h1` (voir ci-dessus).
   Url : `http://www.lsv.ens-cachan.fr/csur/`
   Licence : spécifique (`http://www.lsv.ens-cachan.fr/csur/COPYRIGHT`).
   Réalisé en C.
   Auteurs : Fabrice Parrennes, Jean Goubault-Larrecq.
7. La suite H1 : `h1`, `pl2tptp`, `auto2pl`, `pldet`, `plpurge`, `pl2gastex`, `tptpmorph`, `linauto`, `h1trace`, `h1logstrip`, `h1mc`, `h1mon`, `h1getlog`. La suite d'outils `h1` est une collection d'outils centrés autour de la classe décidable $\mathcal{H}_1$ [RI-61]. Elle peut être vue comme une collection d'outils de manipulation d'automates d'arbres finis, ou de résolution de contraintes ensemblistes. Alors que `h1` lui-même est un démonstrateur automatique de théorèmes par résolution, ou un outil de résolution de contraintes ensemblistes, l'originalité de la suite est sa capacité à produire automatiquement des preuves formelles en Coq, soit d'insatisfiabilité (`h1trace`), soit de satisfiabilité (`h1mc`) — de façon équivalente, des preuves formelles de sécurité dans le cadre de l'analyse de protocoles cryptographiques [CI-37].
   Url : `http://www.lsv.ens-cachan.fr/~goubault/H1.dist/dh1index.html`
   Licence : GPL.
   Évaluation expérimentale en `http://www.lsv.ens-cachan.fr/~goubault/H1.dist/dh1003.`

`html`.
Réalisé en HimML et C. Version : 1.1.
Auteur : Jean Goubault-Larrecq.

8. L'analyseur de protocoles cryptographiques ISpi. ISpi vérifie des protocoles cryptographiques écrits dans une variante du spi-calcul, avec une syntaxe aussi compatible que possible avec l'outil ProVerif de B. Blanchet. Produit des clauses fournies à `h1`. Développé au sein du projet RNTL PROUVÉ.
   Url : `http://www.lsv.ens-cachan.fr/~goubault/ISpi/`
   Licence : GPL.
   Réalisé en HimML. Version : 1.0. Non terminé.
   Auteur : Jean Goubault-Larrecq.
9. TACE : une bibliothèque d'automates d'arbres à contraintes. Décide de problèmes d'appartenance, de non-vacuité et de non-vacuité de l'intersection pour diverses classes d'automates d'arbres finis bottom-up modulo une théorie équationnelle, avec des tests équationnels arbitraires. De même que `h1`, peut être vu comme un démonstrateur automatique de théorèmes pour des fragments décidables spécifiques de la logique du premier ordre, cette fois-ci avec l'égalité [CI-118].
   Url : `http://tace.gforge.inria.fr/`
   Licence : Public Domain.
   Réalisé en Objective Caml.
   Auteurs : Florent Jacquemard, Camille Vacher.
10. Le système NetQi [Lo-2]. NetQi aide à évaluer le degré de résistance, ainsi que de résilience, de réseaux informatiques face aux attaques et aux pannes, en suivant le modèle des jeux d'anticipation [CI-53, CI-5].
    Licence : spécifique.
    Url : `http://www.netqi.org/`
    Réalisé en C.
    Auteur : Elie Bursztein.
11. YAPA. Décide la déductibilité et l'équivalence statique de frames du spi-calcul, modulo des théories équationnelles présentées sous forme de systèmes de réécriture canoniques. Il s'agit d'un outil de base pour décider plusieurs propriétés de sécurité, notamment la résistance aux attaques hors-ligne contre des secrets faibles (mots de passe).
    Url : `http://www.lsv.ens-cachan.fr/~baudet/yapa/index.html`
    Réalisé en Objective Caml.
    Auteur : Mathieu Baudet.

C.10 Projets

PROUVÉ Projet RNTL (Réseau National de recherche et d'innovation en Technologies Logicielles), 2003-2006, 3 ans.
Intitulé : *Protocoles cryptographiques : Outils de Vérification automatique.*
Web : `http://www.lsv.ens-cachan.fr/prouve/`
Coordinateurs : Ralf Treinen et Michael Rusinowitch (LORIA).
Partenaires du LSV : CRIL Technology, France Télécom R&D, LORIA, VERIMAG.

Rossignol Projet de l'ACI "Sécurité Informatique", 2003-2006, 3 ans.
Intitulé : *Sémantique de la vérification des protocoles cryptographiques : théories et applications.*

Web : http://www.cmi.univ-mrs.fr/~lugiez/aci-rossignol.html
Coordinateur : Denis Lugiez (LIF).
Partenaires du LSV : LIF, INRIA Futurs, VERIMAG.

PSI-Robuste Projet de l'ACI Cryptologie, 2002-2005, 3 ans.
Intitulé : *Protection des systèmes d'information : analyse statique de la robustesse des moyens cryptographiques et défense dynamique par techniques modernes de détection d'intrusions.*
Web : http://www.lsv.ens-cachan.fr/~{}goubault/PSIrobuste.html
Porteur : Jean Goubault-Larrecq.

Formacrypt Projet ARA SSIA, 2006-2008, 3 ans.
Intitulé : *Preuves formelles et sémantique probabiliste en cryptographie.*
Web : http://www.di.ens.fr/~blanchet/formacrypt/
Coordinateur : Bruno Blanchet (LIENS).
Partenaires du LSV : LIENS, LORIA.

ProNoBis Action de recherche concertée (ARC) de l'INRIA, 2006-2007, 2 ans.
Intitulé : *Probabilities, Non-Determinism, and Bisimulation in Security.*
Web : http://www.lsv.ens-cachan.fr/~goubault/ProNobis/index.html
Coordinateur : Jean Goubault-Larrecq (EPI SECSI).
Partenaires : EPI Comète INRIA Saclay, ENS Cachan, EPITA, Queen Mary U. of London, U. Paris 7, U. di Verona, U. Oxford.

AVoté Projet ANR SeSur, 2007-2009, 3 ans.
Intitulé : *Analyse formelle de protocoles de vote électronique.*
Web : http://www.lsv.ens-cachan.fr/anr-avote/
Coordinateur : Véronique Cortier (LORIA).
Partenaires : France Télécom, LORIA, Verimag.

Liste complète des publications

La liste ci-dessous regroupe l'ensemble de la production scientifique des membres du laboratoire (au 1er octobre 2008) pour la période 2004–2008, triées par catégories (livres, articles, ...). Elle contient de nombreuses références qui sont des publications *extérieures au laboratoire* car parues alors que leur auteur n'était pas encore membre du LSV. Ces références sont regroupées en fin de liste (dans chaque catégorie) et marquées d'une étoile.

Une analyse succincte de cette production est disponible dans le corps du bilan scientifique, à la page 11.

Livres

[LI-1] Hubert Comon-Lundh, Max Dauchet, Rémi Gilleron, Cristof Löding, Florent Jacquemard, Denis Lugiez, Sophie Tison, and Marc Tommasi. *Tree Automata Techniques and Applications*. November 2007.

[LI-2] Benedikt Bollig. *Formal Models of Communicating Systems — Languages, Automata, and Monadic Second-Order Logic*. Springer, June 2006.

[LI-3] Michel Bidoit and Peter D. Mosses. *CASL User Manual — Introduction to Using the Common Algebraic Specification Language*, volume 2900 of *Lecture Notes in Computer Science*. Springer, 2004.

Chapitres de livres

[Ch-1] Volker Diekert and Paul Gastin. First-order definable languages. *In* Jörg Flum, Erich Grädel, and Thomas Wilke, editors, *Logic and Automata: History and Perspectives*, volume 2 of *Texts in Logic and Games*, pages 261–306. Amsterdam University Press, 2008.

[Ch-2] Volker Diekert and Paul Gastin. Local safety and local liveness for distributed systems. *In Perspectives in Concurrency Theory*. 2008. To appear.

[Ch-3] Manfred Droste and Paul Gastin. Weighted automata and weighted logics. *In* Werner Kuich, Heiko Vogler, and Manfred Droste, editors, *Handbook of Weighted Automata*, EATCS Monographs in Theoretical Computer Science. Springer, 2008. To appear.

[Ch-4] Patricia Bouyer and François Laroussinie. Model checking timed automata. *In* Stephan Merz and Nicolas Navet, editors, *Modeling and Verification of Real-Time Systems*, pages 111–140. ISTE Ltd. – John Wiley & Sons, Ltd., January 2008.

[Ch-5] Serge Haddad and Patrice Moreaux. Verification of probabilistic systems methods and tools. *In* Stephan Merz and Nicolas Navet, editors, *Modeling and Verification of Real-Time Systems*, pages 289–318. ISTE Ltd. – John Wiley & Sons, Ltd., January 2008.

[Ch-6] Jean Goubault-Larrecq. Preuve et vérification pour la sécurité et la sûreté. *In* Jacky Akoka and Isabelle Comyn-Wattiau, editors, *Encyclopédie de l'informatique et des systèmes d'information*, chapter I.6, pages 683–703. Vuibert, December 2006.

[Ch-7] Patricia Bouyer and François Laroussinie. Vérification par automates temporisés. *In* Nicolas Navet, editor, *Systèmes temps-réel 1 : techniques de description et de vérification*, pages 121–150. Hermès, June 2006.

[Ch-8] Philippe Schnoebelen. The verification of probabilistic lossy channel systems. *In* Christel Baier, Boudewijn R. Haverkort, Holger Hermanns, Joost-Pieter Katoen, Markus Siegle, and Frits Vaandrager, editors, *Validation of Stochastic Systems: A Guide to Current Research*, volume 2925 of *Lecture Notes in Computer Science*, pages 445–465. Springer, 2004.

[Ch-9] F. Bréant, Jean-Michel Couvreur, Frédéric Gilliers, Fabrice Kordon, Isabelle Mounier, Emmanuel Paviot-Adet, Denis Poitrenaud, Dan M. Regep, and Grégoire Sutre. Modeling and verifying behavioral aspects. *In* Fabrice Kordon and Michel Lemoine, editors, *Formal Methods for Embedded Distributed Systems: How to Master the Complexity*, chapter 6, pages 171–211. Kluwer Academic Publishers, June 2004.

[Ch-10⋆] Céline Boutrous-Saab, Serge Haddad, and Valérie Monfort. Interopérabilité et services web. *In* Serge Haddad, Fabrice Kordon, and Laure Petrucci, editors, *Méthodes formelles pour les systèmes réartis et coopératifs*, chapter 12, pages 289–315. Hermès, November 2006.

[Ch-11⋆] Serge Haddad. Panorama de la vérification. *In* Serge Haddad, Fabrice Kordon, and Laure Petrucci, editors, *Méthodes formelles pour les systèmes réartis et coopératifs*, chapter 6, pages 121–138. Hermès, November 2006.

[Ch-12⋆] Serge Haddad and Patrice Moreaux. Vérification de systèmes probabilisés : méthodes et outils. *In* Nicolas Navet, editor, *Systèmes temps-réel 1 : techniques de description et de vérification*, pages 261–292. Hermès, June 2006.

[Ch-13⋆] Benedikt Bollig and Martin Leucker. Verifying qualitative properties of probabilistic programs. *In* Christel Baier, Boudewijn R. Haverkort, Holger Hermanns, Joost-Pieter Katoen, Markus Siegle, and Frits Vaandrager, editors, *Validation of Stochastic Systems: A Guide to Current Research*, volume 2925 of *Lecture Notes in Computer Science*, pages 124–146. Springer, 2004.

Participation à des ouvrages collectifs

[OC-1] Bruno Bouyssonouse and Joseph Sifakis, editors. *Embedded Systems Design: The ARTIST Roadmap for Research and Development*, volume 3436 of *Lecture Notes in Computer Science*. Springer, 2005.

Édition d'ouvrages collectifs

[Ed-1] Liqun Chen, Steve Kremer, and Mark D. Ryan, editors. *Formal Protocol Verification Applied*, volume 07421 of *Dagstuhl Seminar Proceedings*.

[Ed-2] Peter Habermehl and Tomáš Vojnar, editors. *Proceedings of the 10th International Workshop on Verification of Infinite State Systems (INFINITY'08)*, Electronic Notes in Theoretical Computer Science. Elsevier Science Publishers, 2008. To appear.

[Ed-3] Stéphane Demri and Christian S. Jensen, editors. *Proceedings of the 15th International Symposium on Temporal Representation and Reasoning (TIME'08)*. IEEE Computer Society Press.

[Ed-4] Monica Nesi and Ralf Treinen, editors. *Preliminary Proceedings of the 2nd International Workshop on Security and Rewriting Techniques (SecReT'07)*.

[Ed-5] Hubert Comon-Lundh, Claude Kirchner, and Hélène Kirchner, editors. *Rewriting, Computation and Proof — Essays Dedicated to Jean-Pierre Jouannaud on the Occasion of his 60th Birthday*, volume 4600 of *Lecture Notes in Computer Science*. Springer.

[Ed-6] Siva Anantharaman, Paul Gastin, Gaétan Hains, John Mullins, and Michaël Rusinowitch, editors. *Selected papers of the International Workshop on Security Analysis of Systems: Formalisms and Tools (SASYFT'04)*, volume 11(1).

[Ed-7] Franck Cassez and François Laroussinie. Contrôle des applications temps-réel : modèles temporisés et hybrides. *Technique et Science Informatiques*, 25(3), 2006.

[Ed-8] Eugène Asarin and Patricia Bouyer, editors. *Proceedings of the 4th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'06)*, volume 4202 of *Lecture Notes in Computer Science*. Springer.

[Ed-9] Patricia Bouyer and P. Madhusudan, editors. *Proceedings of the 3rd Workshop on Games in Design and Verification (GDV'06)*.

[Ed-10] Véronique Cortier and Steve Kremer, editors. *Proceedings of the 2nd Workshop on Formal and Computational Cryptography (FCC'06)*.

[Ed-11] Philippe Schnoebelen, editor. *Proceedings of the 5th International Workshop on Verification of Infinite State Systems (INFINITY'03)*, volume 98 of *Electronic Notes in Theoretical Computer Science*. Elsevier Science Publishers, August 2004.

[Ed-12⋆] Jacques Farré, Igor Litovsky, and Sylvain Schmitz, editors. *Revised Selected Papers of the 10th International Conference on Implementation and Application of Automata (CIAA'05)*, volume 3845 of *Lecture Notes in Computer Science*. Springer, 2006.

[Ed-13⋆] Serge Haddad, Fabrice Kordon, and Laure Petrucci, editors. *Méthodes formelles pour les systèmes répartis et coopératifs*. Traité IC2 – série informatique et systèmes d'information. Hermès, November 2006.

Articles dans des revues internationales avec comité de lecture

[RI-1] Martin De Wulf, Laurent Doyen, Nicolas Markey, and Jean-François Raskin. Robust safety of timed automata. *Formal Methods in System Design*, 2008. To appear.

[RI-2] Stéphanie Delaune, Steve Kremer, and Mark D. Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 2008. To appear.

[RI-3] Stéphane Demri and Régis Gascon. Verification of qualitative $\mathbb{Z}$ constraints. *Theoretical Computer Science*, 2008. To appear.

[RI-4] Stéphane Demri and Ranko Lazić. LTL with the freeze quantifier and register automata. *ACM Transactions on Computational Logic*, 2008. To appear.

[RI-5] Panos Papadimitratos, Marcin Poturalski, Patrick Schaller, Pascal Lafourcade, David Basin, Srdjan Čapkun, and Jean-Pierre Hubaux. Secure neighborhood discovery: A fundamental element for mobile ad hoc networking. *IEEE Communications Magazine*, 2008. To appear.

[RI-6] Sami Taktak, Emmanuelle Encrenaz, and Jean-Lou Desbarbieux. A tool for automatic detection of deadlocks in wormhole networks on chip. *ACM Transactions on Design Automation of Electronic Systems*, 2008. To appear.

[RI-7] Sébastien Bardin, Alain Finkel, Jérôme Leroux, and Laure Petrucci. FAST: Acceleration from theory to practice. *International Journal on Software Tools for Technology Transfer*, 10(5):401–424, October 2008.

[RI-8] Béatrice Bérard, Franck Cassez, Serge Haddad, Didier Lime, and Olivier H. Roux. When are timed automata weakly timed bisimilar to time Petri nets? *Theoretical Computer Science*, 403(2-3):202–220, September 2008.

[RI-9] Benedikt Bollig and Dietrich Kuske. Muller message-passing automata and logics. *Information and Computation*, 206(9-10):1084–1094, September-October 2008.

[RI-10] Patricia Bouyer, Kim G. Larsen, and Nicolas Markey. Model checking one-clock priced timed automata. *Logical Methods in Computer Science*, 4(2:9), June 2008.

[RI-11] Hubert Comon-Lundh, Florent Jacquemard, and Nicolas Perrin. Visibly tree automata with memory and constraints. *Logical Methods in Computer Science*, 4(2:8), June 2008.

[RI-12] Volker Diekert, Paul Gastin, and Manfred Kufleitner. A survey on small fragments of first-order logic over finite words. *International Journal of Foundations of Computer Science*, 19(3):513–548, June 2008.

[RI-13] Manfred Droste and Paul Gastin. On aperiodic and star-free formal power series in partially commuting variables. *Theory of Computing Systems*, 42(4):608–631, May 2008.

[RI-14] François Laroussinie, Nicolas Markey, and Ghassan Oreiby. On the expressiveness and complexity of ATL. *Logical Methods in Computer Science*, 4(2:7), May 2008.

[RI-15] Florent Jacquemard, Michaël Rusinowitch, and Laurent Vigneron. Tree automata with equality constraints modulo equational theories. *Journal of Logic and Algebraic Programming*, 75(2):182–208, April 2008.

[RI-16] Patricia Bouyer, Ed Brinksma, and Kim G. Larsen. Optimal infinite scheduling for multi-priced timed automata. *Formal Methods in System Design*, 32(1):2–23, February 2008.

[RI-17] Stéphanie Delaune, Pascal Lafourcade, Denis Lugiez, and Ralf Treinen. Symbolic protocol analysis for monoidal equational theories. *Information and Computation*, 206(2-4):312–351, February-April 2008.

[RI-18] Patricia Bouyer, Serge Haddad, and Pierre-Alain Reynier. Timed Petri nets and timed automata: On the discriminating power of Zeno sequences. *Information and Computation*, 206(1):73–107, January 2008.

[RI-19] Stéphane Demri and Ewa Orłowska. Relative nondeterministic information logic is EXPTIME-complete. *Fundamenta Informaticae*, 75(1-4):163–178, 2007.

[RI-20] Christel Baier, Nathalie Bertrand, and Philippe Schnoebelen. Verifying nondeterministic probabilistic channel systems against ω-regular linear-time properties. *ACM Transactions on Computational Logic*, 9(1), December 2007.

[RI-21] Paul Gastin and Dietrich Kuske. Uniform satisfiability in PSPACE for local temporal logics over Mazurkiewicz traces. *Fundamenta Informaticae*, 80(1-3):169–197, November 2007.

[RI-22] Béatrice Bérard, Paul Gastin, and Antoine Petit. Timed substitutions for regular signal-event languages. *Formal Methods in System Design*, 31(2):101–134, October 2007.

[RI-23] Patricia Bouyer, Thomas Brihaye, Véronique Bruyère, and Jean-François Raskin. On the optimal reachability problem on weighted timed automata. *Formal Methods in System Design*, 31(2):135–175, October 2007.

[RI-24] Laura Bozzelli. Complexity results on branching-time pushdown model checking. *Theoretical Computer Science*, 379(1-2):286–297, June 2007.

[RI-25] Manfred Droste and Paul Gastin. Weighted automata and weighted logics. *Theoretical Computer Science*, 380(1-2):69–86, June 2007.

[RI-26] François Laroussinie and Jeremy Sproston. State explosion in almost-sure probabilistic reachability. *Information Processing Letters*, 102(6):236–241, June 2007.

[RI-27] Kumar N. Verma and Jean Goubault-Larrecq. Alternating two-way AC-tree automata. *Information and Computation*, 205(6):817–869, June 2007.

[RI-28] Pascal Lafourcade, Denis Lugiez, and Ralf Treinen. Intruder deduction for the equational theory of Abelian groups with distributive encryption. *Information and Computation*, 205(4):581–623, April 2007.

[RI-29] Stéphane Demri and Deepak D'Souza. An automata-theoretic approach to constraint LTL. *Information and Computation*, 205(3):380–415, March 2007.

[RI-30] Stéphane Demri and David Nowak. Reasoning about transfinite sequences. *International Journal of Foundations of Computer Science*, 18(1):87–112, February 2007.

[RI-31] Stéphane Demri, Ranko Lazić, and David Nowak. On the freeze quantifier in constraint LTL: Decidability and complexity. *Information and Computation*, 205(1):2–24, January 2007.

[RI-32] Mathieu Baudet. Random polynomial-time attacks and Dolev-Yao models. *Journal of Automata, Languages and Combinatorics*, 11(1):7–21, 2006.

[RI-33] Véronique Cortier, Stéphanie Delaune, and Pascal Lafourcade. A survey of algebraic properties used in cryptographic protocols. *Journal of Computer Security*, 14(1):1–43, 2006.

[RI-34] Stéphane Demri. Linear-time temporal logics with Presburger constraints: An overview. *Journal of Applied Non-Classical Logics*, 16(3-4):311–347, 2006.

[RI-35] Stéphanie Delaune. An undecidability result for AGh. *Theoretical Computer Science*, 368(1-2):161–167, December 2006.

[RI-36] Volker Diekert and Paul Gastin. Pure future local temporal logics are expressively complete for Mazurkiewicz traces. *Information and Computation*, 204(11):1597–1619, November 2006.

[RI-37] Stéphane Demri. LTL over integer periodicity constraints. *Theoretical Computer Science*, 360(1-3):96–123, August 2006.

[RI-38] Antonín Kučera and Philippe Schnoebelen. A general approach to comparing infinite-state systems with their finite-state specifications. *Theoretical Computer Science*, 358(2-3):315–333, August 2006.

[RI-39] Nicolas Markey and Jean-François Raskin. Model checking restricted sets of timed paths. *Theoretical Computer Science*, 358(2-3):273–292, August 2006.

[RI-40] Alexander Rabinovich and Philippe Schnoebelen. BTL_2 and the expressive power of $ECTL^+$. *Information and Computation*, 204(7):1023–1044, July 2006.

[RI-41] Gerd Behrmann, Patricia Bouyer, Kim G. Larsen, and Radek Pelánek. Lower and upper bounds in zone-based abstractions of timed automata. *International Journal on Software Tools for Technology Transfer*, 8(3):204–215, June 2006.

[RI-42] Patricia Bouyer, Thomas Brihaye, and Nicolas Markey. Improved undecidability results on weighted timed automata. *Information Processing Letters*, 98(5):188–194, June 2006.

[RI-43] Stéphane Demri, François Laroussinie, and Philippe Schnoebelen. A parametric analysis of the state explosion problem in model checking. *Journal of Computer and System Sciences*, 72(4):547–575, June 2006.

[RI-44] Volker Diekert and Paul Gastin. From local to global temporal logics over Mazurkiewicz traces. *Theoretical Computer Science*, 356(1-2):126–135, May 2006.

[RI-45] Alain Finkel, Gilles Geeraerts, Jean-François Raskin, and Laurent Van Begin. On the ω-language expressive power of extended Petri nets. *Theoretical Computer Science*, 356(3):374–386, May 2006.

[RI-46] Michel Bidoit and Rolf Hennicker. Constructor-based observational logic. *Journal of Logic and Algebraic Programming*, 67(1-2):3–51, April-May 2006.

[RI-47] Stéphanie Delaune. Easy intruder deduction problems with homomorphisms. *Information Processing Letters*, 97(6):213–218, March 2006.

[RI-48] François Laroussinie, Nicolas Markey, and Philippe Schnoebelen. Efficient timed model checking for discrete-time systems. *Theoretical Computer Science*, 353(1-3):249–271, March 2006.

[RI-49] Nicolas Markey and Philippe Schnoebelen. Mu-calculus path checking. *Information Processing Letters*, 97(6):225–230, March 2006.

[RI-50] Laurent Fribourg, Stéphane Messika, and Claudine Picaronny. Coupling and self-stabilization. *Distributed Computing*, 18(3):221–232, February 2006.

[RI-51] Christel Baier, Nathalie Bertrand, and Philippe Schnoebelen. A note on the attractor-property of infinite-state Markov chains. *Information Processing Letters*, 97(2):58–63, January 2006.

[RI-52] Rohit Chadha, Steve Kremer, and Andre Scedrov. Formal analysis of multi-party contract signing. *Journal of Automated Reasoning*, 36(1-2):39–83, January 2006.

[RI-53] Stéphanie Delaune and Florent Jacquemard. Decision procedures for the security of protocols with probabilistic encryption against offline dictionary attacks. *Journal of Automated Reasoning*, 36(1-2):85–124, January 2006.

[RI-54] Patricia Bouyer and Fabrice Chevalier. On conciseness of extensions of timed automata. *Journal of Automata, Languages and Combinatorics*, 10(4):393–405, 2005.

[RI-55] Parosh Aziz Abdulla, Nathalie Bertrand, Alexander Rabinovich, and Philippe Schnoebelen. Verification of probabilistic systems with faulty communication. *Information and Computation*, 202(2):141–165, November 2005.

[RI-56] Gérard Cécé and Alain Finkel. Verification of programs with half-duplex communication. *Information and Computation*, 202(2):166–190, November 2005.

[RI-57] Denis Lugiez and Philippe Schnoebelen. Decidable first-order transition logics for PA-processes. *Information and Computation*, 203(1):75–113, November 2005.

[RI-58] Kumar N. Verma and Jean Goubault-Larrecq. Karp-Miller trees for a branching extension of VASS. *Discrete Mathematics & Theoretical Computer Science*, 7(1):217–230, November 2005.

[RI-59] Stéphane Demri. A reduction from DLP to PDL. *Journal of Logic and Computation*, 15(5):767–785, October 2005.

[RI-60] Alain Finkel and Jérôme Leroux. The convex hull of a regular set of integer vectors is polyhedral and effectively computable. *Information Processing Letters*, 96(1):30–35, October 2005.

[RI-61] Jean Goubault-Larrecq. Deciding $\mathcal{H}_1$ by resolution. *Information Processing Letters*, 95(3):401–408, August 2005.

[RI-62] Jean Goubault-Larrecq, Muriel Roger, and Kumar N. Verma. Abstraction and resolution modulo AC: How to verify Diffie-Hellman-like protocols automatically. *Journal of Logic and Algebraic Programming*, 64(2):219–251, August 2005.

[RI-63] Sophie Pinchinat and Stéphane Riedweg. A decidable class of problems for control under partial observation. *Information Processing Letters*, 95(4):454–465, August 2005.

[RI-64] The Artist Education Group. Guidelines for a graduate curriculum on embedded software and systems. *ACM Transactions in Embedded Computing Systems*, 4(3):587–611, August 2005.

[RI-65] Stéphane Demri and Hans de Nivelle. Deciding regular grammar logics with converse through first-order logic. *Journal of Logic, Language and Information*, 14(3):289–319, June 2005.

[RI-66] Jean Goubault-Larrecq. Extensions of valuations. *Mathematical Structures in Computer Science*, 15(2):271–297, April 2005.

[RI-67] Hubert Comon and Véronique Cortier. Tree automata with one memory, set constraints and cryptographic protocols. *Theoretical Computer Science*, 331(1):143–214, February 2005.

[RI-68] Volker Diekert and Paul Gastin. Local temporal logic is expressively complete for cograph dependence alphabets. *Information and Computation*, 195(1-2):30–52, November 2004.

[RI-69] Alain Finkel, Pierre McKenzie, and Claudine Picaronny. A well-structured framework for analysing Petri net extensions. *Information and Computation*, 195(1-2):1–29, November 2004.

[RI-70] Patricia Bouyer, Catherine Dufourd, Emmanuel Fleury, and Antoine Petit. Updatable timed automata. *Theoretical Computer Science*, 321(2-3):291–345, August 2004.

[RI-71] Béatrice Bérard, Patricia Bouyer, and Antoine Petit. Analysing the PGM protocol with Uppaal. *International Journal of Production Research*, 42(14):2773–2791, July 2004.

[RI-72] Patricia Bouyer. Forward analysis of updatable timed automata. *Formal Methods in System Design*, 24(3):281–320, May 2004.

[RI-73] Nicolas Markey. Past is for free: On the complexity of verifying linear temporal properties with past. *Acta Informatica*, 40(6-7):431–458, May 2004.

[RI-74] Hubert Comon-Lundh and Véronique Cortier. Security properties: Two agents are sufficient. *Science of Computer Programming*, 50(1-3):51–71, March 2004.

[RI-75] Marie Duflot, Laurent Fribourg, and Claudine Picaronny. Randomized dining philosophers without fairness assumption. *Distributed Computing*, 17(1):65–76, February 2004.

[RI-76] Nicolas Markey and Philippe Schnoebelen. A PTIME-complete matching problem for SLP-compressed words. *Information Processing Letters*, 90(1):3–6, January 2004.

[RI-77⋆] Pierre-Cyrille Héam. A note on partially ordered tree automata. *Information Processing Letters*, 108(4):242–246, October 2008.

[RI-78⋆] Yohan Boichut and Pierre-Cyrille Héam. A theorerical limit for safety verification techniques with regular fix-point computations. *Information Processing Letters*, 108(1):1–2, September 2008.

[RI-79⋆] Filippo Furfaro, Giuseppe M. Mazzeo, Domenico Saccà, and Cristina Sirangelo. Compressed hierarchical binary histograms for summarizing multi-dimensional data. *International Journal on Knowledge and Information Systems*, 15(3):335–380, June 2008.

[RI-80⋆] Michel Bidoit, Donald Sannella, and Andrzej Tarlecki. Observational interpretation of CASL specifications. *Mathematical Structures in Computer Science*, 18(2):325–371, April 2008.

[RI-81⋆] Ahmed Bouajjani, Peter Habermehl, and Tomáš Vojnar. Verification of parametric concurrent systems with prioritised FIFO resource management. *Formal Methods in System Design*, 32(2):129–172, April 2008.

[RI-82⋆] Thomas Brihaye. Words and bisimulation of dynamical systems. *Discrete Mathematics & Theoretical Computer Science*, 9(2), 2007.

[RI-83⋆] Serge Haddad and Denis Poitrenaud. Recursive Petri nets – Theory and application to discrete event systems. *Acta Informatica*, 44(7-8):463–508, December 2007.

[RI-84⋆] Thomas Chatain and Victor Khomenko. On the well-foundedness of adequate orders used for construction of complete unfolding prefixes. *Information Processing Letters*, 104(4):129–136, November 2007.

[RI-85⋆] Dietmar Berwanger, Erich Grädel, and Giacomo Lenzi. The variable hierarchy of the μ-calculus is strict. *Theory of Computing Systems*, 40(4):437–466, June 2007.

[RI-86⋆] Luc Segoufin. Static analysis of XML processing with data values. *SIGMOD Records*, 36(1):31–38, March 2007.

[RI-87⋆] Serge Haddad and Patrice Moreaux. Sub-stochastic matrix analysis for bounds computation — Theoretical results. *European Journal of Operational Research*, 167(2), January 2007.

[RI-88⋆] Graham Steel. Formal analysis of PIN block attacks. *Theoretical Computer Science*, 367(1-2):257–270, November 2006.

[RI-89⋆] Benedikt Bollig and Martin Leucker. Message-passing automata are expressively equivalent to EMSO logic. *Theoretical Computer Science*, 358(2-3):150–172, August 2006.

[RI-90⋆] Luis Caires and Étienne Lozes. Elimination of quantifiers and undecidability in spatial logics for concurrency. *Theoretical Computer Science*, 358(2-3):293–314, August 2006.

[RI-91⋆] David Nowak. Synchronous structures. *Information and Computation*, 204(8):1295–1324, August 2006.

[RI-92⋆] François-Régis Sinot. Call-by-need in token-passing nets. *Mathematical Structures in Computer Science*, 16(4):639–666, August 2006.

[RI-93⋆] Anca Muscholl, Mathias Samuelides, and Luc Segoufin. Complementing deterministic tree-walking automata. *Information Processing Letters*, 99(1):33–39, July 2006.

[RI-94⋆] Jean Cardinal, Steve Kremer, and Stefan Langerman. Juggling with pattern matching. *Theory of Computing Systems*, 39(3):425–437, June 2006.

[RI-95⋆] Emmanuel Beffara and François Maurel. Concurrent nets: a study of prefixing in process calculi. *Theoretical Computer Science*, 356(3):356–373, May 2006.

[RI-96⋆] Serge Abiteboul, Luc Segoufin, and Victor Vianu. Representing and querying XML with incomplete information. *ACM Transactions on Database Systems*, 31(1):208–254, March 2006.

[RI-97⋆] Thomas Brihaye. A note on the undecidability of the reachability problem for o-minimal dynamical systems. *Mathematical Logic Quarterly*, 52(2):165–170, March 2006.

[RI-98⋆] Thomas Brihaye, Véronique Bruyère, and Jean-François Raskin. On model-checking timed automata with stopwatch observers. *Information and Computation*, 204(3):408–433, March 2006.

[RI-99⋆] Serge Haddad and Jean-François Pradat-Peyre. New efficient Petri nets reductions for parallel programs verification. *Parallel Processing Letters*, 16(1):101–116, March 2006.

[RI-100⋆] Daniel Hirschkoff, Étienne Lozes, and Davide Sangiorgi. On the expressiveness of the ambient logic. *Logical Methods in Computer Science*, 2(2), March 2006.

[RI-101⋆] Anca Muscholl, Thomas Schwentick, and Luc Segoufin. Active context-free games. *Theory of Computing Systems*, 39(1):237–276, February 2006.

[RI-102⋆] Graham Steel and Alan Bundy. Attacking group protocols by refuting incorrect inductive conjectures. *Journal of Automated Reasoning*, 36(1-2):149–176, January 2006.

[RI-103⋆] Thomas Brihaye and Christian Michaux. On the expressiveness and decidability of o-minimal hybrid systems. *Journal of Complexity*, 21(4):447–478, August 2005.

[RI-104⋆] Elie Bursztein. TCP timestamp to count hosts behind NAT. *Phrack Magazine*, 63(3):linenoise 0x03–2, August 2005.

[RI-105⋆] Maribel Fernández, Ian Mackie, and François-Régis Sinot. Closed reduction: Explicit substitutions without α-conversion. *Mathematical Structures in Computer Science*, 15(2):343–381, April 2005.

[RI-106⋆] Maribel Fernández, Ian Mackie, and François-Régis Sinot. Lambda-calculus with director strings. *Applicable Algebra in Engineering, Communication and Computing*, 15(6):393–437, April 2005.

[RI-107⋆] François-Régis Sinot. Director strings revisited: A generic approach to the efficient representation of free variables in higher-order rewriting. *Journal of Logic and Computation*, 15(2):201–218, April 2005.

[RI-108⋆] Georg Gottlob, Christoph Koch, Reinhard Pichler, and Luc Segoufin. The complexity of XPath query evaluation and XML typing. *Journal of the ACM*, 52(2):284–335, March 2005.

[RI-109⋆] Serge Haddad, Patrice Moreaux, Matteo Sereno, and Manuel Silva. Product-form and stochastic Petri nets: A structural approach. *Performance Evaluation*, 59(4):313–336, March 2005.

[RI-110⋆] Nathalie Bertrand, Irène Charon, Olivier Hudry, and Antoine Lobstein. 1-identifying codes on trees. *Australasian Journal of Combinatorics*, 31:21–35, February 2005.

[RI-111⋆] Étienne Lozes. Elimination of spatial connectives in static spatial logics. *Theoretical Computer Science*, 330(3):475–499, February 2005.

[RI-112⋆] Nicole Bidoit, Sandra de Amo, and Luc Segoufin. Order independent temporal properties. *Journal of Logic and Computation*, 14(2):277–298, 2004.

[RI-113⋆] Dietmar Berwanger and Erich Grädel. Fixed-point logics and solitaire games. *Theory of Computing Systems*, 37(6):675–694, December 2004.

[RI-114⋆] Weiwen Xu. Automatic generation of symbolic model for parameterized synchronous systems. *Journal of Computer Science and Technology*, 19(6):812–819, November 2004.

[RI-115⋆] Nathalie Bertrand, Irène Charon, Olivier Hudry, and Antoine Lobstein. Identifying and locating-dominating codes on chains and cycles. *European Journal of Combinatorics*, 25(7):969–987, October 2004.

[RI-116⋆] Joyce El Haddad and Serge Haddad. A fault-tolerant communication mechanism for cooperative robots. *International Journal of Production Research*, 42(14):2793–2808, July 2004.

[RI-117⋆] Paul Gastin and Michael W. Mislove. A simple process algebra based on atomic actions with resources. *Mathematical Structures in Computer Science*, 14(1):1–55, February 2004.

Articles dans d'autres revues

[RE-1] Rémy Chevallier, Emmanuelle Encrenaz-Tiphène, Laurent Fribourg, and Weiwen Xu. Timing analysis of an embedded memory: SPSMALL. *WSEAS Transactions on Circuits and Systems*, 5(7):973–978, July 2006.

[RE-2] Patricia Bouyer and Fabrice Chevalier. On the control of timed and hybrid systems. *EATCS Bulletin*, 89:79–96, June 2006.

[RE-3⋆] Gérard Cécé, Pierre-Cyrille Héam, and Yann Mainier. Clôtures transitives de semi-commutations et model-checking régulier. *Technique et Science Informatiques*, 27(1-2):7–28, 2008.

[RE-4⋆] Adam Antonik, Michael Huth, Kim G. Larsen, Ulrik Nyman, and Andrzej Wasowski. 20 years of mixed and modal specifications. *EATCS Bulletin*, 95:94–129, June 2008.

[RE-5⋆] Sami Evangelista, Serge Haddad, and Jean-François Pradat-Peyre. De nouvelles réductions colorées pour la validation de logiciels. *Revue Électronique des Sciences et Technologies de l'Automatique*, 1, 2004.

[RE-6⋆] Serge Haddad, Lynda Mokdad, and Patrice Moreaux. Évaluation de performance des systèmes stochastiques à événements discrets non Markoviens — une nouvelle approche. *Revue Électronique des Sciences et Technologies de l'Automatique*, 1, 2004.

[RE-7⋆] Frédéric Loulergue, Frédéric Gava, Myrto Arapinis, and Frédéric Dabrowski. Semantics and implementation of minimally synchronous parallel ML. *International Journal of Computer & Information Science*, 5(3):182–199, September 2004.

Communications invitées dans une conférence internationale

[IN-1] Patricia Bouyer. Model-checking timed temporal logics. *In* Carlos Areces and Stéphane Demri, editors, *Proceedings of the 4th Workshop on Methods for Modalities (M4M-5), Cachan, France, November 2007*, Electronic Notes in Theoretical Computer Science. Elsevier Science Publishers, 2008. To appear.

[IN-2] Steve Kremer. Computational soundness of equational theories (tutorial). *In* Gilles Barthe and Cédric Fournet, editors, *Revised Selected Papers from the 3rd Symposium on Trustworthy Global Computing (TGC'07), Sophia-Antipolis, France, November 2007*, volume 4912 of *Lecture Notes in Computer Science*, pages 363–382. Springer, 2008.

[IN-3] Hubert Comon-Lundh. Challenges in the automated verification of security protocols. *In* Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *Proceedings of the 4th International Joint Conference on Automated Reasoning (IJCAR'08), Sydney, Australia, August 2008*, volume 5195 of *Lecture Notes in Artificial Intelligence*, pages 396–409. Springer-Verlag.

[IN-4] Jean Goubault-Larrecq and Julien Olivain. Orchids, and bad weeds. *In* Martin Leucker, editor, *Proceedings of the 8th Workshop on Runtime Verification (RV'08), Budapest, Hungary, March 2008*, Lecture Notes in Computer Science. Springer. To appear.

[IN-5] Philippe Schnoebelen. Model checking branching-time logics. *In* Valentin Goranko and X. Sean Wang, editors, *Proceedings of the 14th International Symposium on Temporal Representation and Reasoning (TIME'07), Alicante, Spain, June 2007*, page 5. IEEE Computer Society Press.

[IN-6] Patricia Bouyer. Weighted timed automata: Model-checking and games. *In* Steve Brookes and Michael Mislove, editors, *Proceedings of the 22nd Conference on Mathematical Foundations of Programming Semantics (MFPS'06), Genova, Italy, May 2006*, volume 158 of *Electronic Notes in Theoretical Computer Science*, pages 3–17. Elsevier Science Publishers. Invited paper.

[IN-7] François Laroussinie. Timed modal logics for the verification of real-time systems. *In* Holger Schlingloff, editor, *Proceedings of the 4th Workshop on Methods for Modalities (M4M-4), Berlin, Germany, December 2005*, volume 194 of *Informatik Bericht*, pages 293–305. Humboldt Universität zu Berlin. Invited paper.

[IN-8] Patricia Bouyer. Timed automata — From theory to implementation. Invited tutorial, 1st International Conference on the Quantitative Evaluation of System (QEST'04), Twente, The Netherlands, September 2004.

[IN-9] Michel Bidoit and Rolf Hennicker. Glass box and black box views of state-based system specifications. *In* Charles Rattray, Savitri Maharaj, and Carron Shankland, editors, *Proceedings of the 10th International Conference on Algebraic Methodology and Software Technology (AMAST'04), Stirling, UK, July 2004*, volume 3116 of *Lecture Notes in Computer Science*, page 19. Springer. Invited talk.

[IN-10] Hubert Comon-Lundh. Intruder theories (ongoing work). *In* Igor Walukiewicz, editor, *Proceedings of the 7th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'04), Barcelona, Spain, March 2004*, volume 2987 of *Lecture Notes in Computer Science*, pages 1–4. Springer. Invited talk.

Autres communications sur invitation, tutorials, ...

[In-1] Patricia Bouyer. Probabilities in timed automata. Invited talk, Workshop Automata and Verification (AV'08), Mons, Belgium, August 2008.

[In-2] Nicolas Markey. Infinite runs in weighted times games with energy constraints. Invited talk, Workshop Automata and Verification (AV'08), Mons, Belgium, August 2008.

[In-3] Philippe Schnoebelen. The complexity of lossy channel systems. Invited talk, Workshop Automata and Verification (AV'08), Mons, Belgium, August 2008.

[In-4] Nicolas Markey. Timed systems – model checking and games. Invited tutorial, 8th School on Modelling and Verifying Parallel Processes (MOVEP'08), Nouan-le-Fuzelier, France, June 2008.

[In-5] Patricia Bouyer. Model-checking timed temporal logics. *In* Tei-Wei Kuo and Samuel Cruz-Lara, editors, *Proceedings of the 4th Taiwanese-French Conference on Information Technology (TFIT'08), Taipei, Taiwan, ROC, March 2008*, pages 132–142.

[In-6] Stéphanie Delaune, Steve Kremer, and Graham Steel. Formal analysis of PKCS#11. *In* Tei-Wei Kuo and Samuel Cruz-Lara, editors, *Proceedings of the 4th Taiwanese-French Conference on Information Technology (TFIT'08), Taipei, Taiwan, ROC, March 2008*, pages 267–278.

[In-7] Franck Cassez and Nicolas Markey. Contrôle et implémentation des systèmes temporisés. *In Actes de la 5ème École Temps-Réel (ETR'07), Nantes, France, September 2007*, pages 111–123.

[In-8] Stéphanie Delaune. Modélisation des propriétés d'anonymat dans le pi-calcul appliqué. Invited talk, Sécurité Informatique et Vote ÉlecTrOnique, Tunis, Tunisie, April 2007.

[In-9] Fabrice Chevalier. Decision procedures for timed logics. Invited talk, Advances and Issues in Timed Systems, Kolkata, India, December 2006.

[In-10] Paul Gastin. Refinements and abstractions of signal-event (timed) languages. Invited talk, Advances and Issues in Timed Systems, Kolkata, India, December 2006.

[In-11] Jean Goubault-Larrecq. The ORCHIDS intrusion prevention system. Invited talk, Annual Adaptive and Resilient Computing Systems Workshop (AARCS'06), Santa Fe, New Mexico, USA, November 2006.

[In-12] Philippe Schnoebelen. De nouvelles applications pour le model-checking. Invited lecture, Journées à l'occasion des 20 ans du LIPN, Villetaneuse, France, November 2006.

[In-13] Nicolas Markey. Verification of multi-agent systems with ATL. Invited talk, FNRS meeting on "Synthesis and Verification", October 2006.

[In-14] François Laroussinie and Nicolas Markey. Expressiveness of temporal logics. Introductory course, 18th European Summer School in Logic, Language and Information (ESSLLI'06), Málaga, Spain, July-August 2006.

[In-15] Steve Kremer. Formal verification of cryptographic protocols. Invited tutorial, 7th School on Modelling and Verifying Parallel Processes (MOVEP'06), Bordeaux, France, June 2006. 5 pages.

[In-16] Paul Gastin. Distributed synthesis: synchronous and asynchronous semantics. Invited talk, 34ème École de Printemps en Informatique Théorique, Ile de Ré, France, May 2006.

[In-17] Paul Gastin. Refinements and abstractions of signal-event (timed) languages. Invited talk, 22nd Conference on Mathematical Foundations of Programming Semantics (MFPS'06), May 2006.

[In-18] Paul Gastin. Weigthed logics and weighted automata. Invited talk, Workshop Weighted Automata: Theory and Applications, Leipzig, Germany, March 2006.

[In-19] Jean Goubault-Larrecq. A biased survey of models and methods for verifying cryptographic protocols. Invited talk, Workshop on Classical and Quantum Information Security, Pasadena, California, USA, December 2005.

[In-20] Paul Gastin. On the synthesis of distributed controllers. Invited talk, Workshop Perspectives in Verification, in honor of Wolfgang Thomas on the occasion of his Doctorate Honoris Causa, Cachan, France, November 2005.

[In-21] Karine Altisen, Patricia Bouyer, Thierry Cachat, Franck Cassez, and Guillaume Gardey. Introduction au contrôle des systèmes temps-réel. *In* Hassane Alla and Éric Rutten, editors, *Actes du 5ème Colloque sur la Modélisation des Systèmes Réactifs (MSR'05), Autrans, France, October 2005*, pages 367–380. Hermès. Invited paper.

[In-22] Karine Altisen, Nicolas Markey, Pierre-Alain Reynier, and Stavros Tripakis. Implémentabilité des automates temporisés. *In* Hassane Alla and Éric Rutten, editors, *Actes du 5ème Colloque sur la Modélisation des Systèmes Réactifs (MSR'05), Autrans, France, October 2005*, pages 395–406. Hermès. Invited paper.

[In-23] Patricia Bouyer, Fabrice Chevalier, Moez Krichen, and Stavros Tripakis. Observation partielle des systèmes temporisés. *In* Hassane Alla and Éric Rutten, editors, *Actes du 5ème Colloque sur la Modélisation des Systèmes Réactifs (MSR'05), Autrans, France, October 2005*, pages 381–393. Hermès. Invited paper.

[In-24] Patricia Bouyer. Foundations of timed systems. *In Proc. of the ARTIST2 Summer School on Component & Modelling, Testing & Verification, and Statical Analysis of Embedded Systems, Nässlingen, Sweden, September-October 2005*.

[In-25] Patricia Bouyer. An introduction to timed automata. *In Actes de la 3ème École Temps-Réel (ETR'05), Nancy, France, September 2005*, pages 111–123.

[In-26] Patricia Bouyer. Optimal timed games. Invited talk, 5th International Workshop on Automated Verification of Critical Systems (AVoCS'05), Warwick, UK, September 2005.

[In-27] Patricia Bouyer. Optimal reachability timed games. Invited talk, 7th International Workshop on Verification of Infinite State Systems (INFINITY'05), San Francisco, USA, August 2005.

[In-28] Patricia Bouyer. Weighted timed automata: Model-checking and games. Invited talk, Workshop CORTOS'06, Bonn, Germany, August 2005.

[In-29] Stéphane Demri. On the complexity of information logics. Invited talk, Workshop on Logical and Algebraic Foundations of Rough Sets, Regina, Canada, August 2005.

[In-30] Patricia Bouyer. Partial observation of timed systems. Invited talk, 2nd Workshop on Games in Design and Verification, Edinburgh, Scotland, July 2005.

[In-31] Patricia Bouyer. Synthesis of timed systems. Invited lecture, Spring School on Infinite Games and Their Applications, Bonn, Germany, March 2005.

[In-32] Patricia Bouyer. Timed automata and extensions: Decidability limits. Invited talk, 5èmes Journées Systèmes Infinis (JSI'05), Cachan, France, March 2005.

[In-33] Patricia Bouyer. Timed automata — From theory to implementation. Invited tutorial, 6th Winter School on Modelling and Verifying Parallel Processes (MOVEP'04), Brussels, Belgium, December 2004. 27 pages.

[In-34] Paul Gastin. Basics of model checking. Invited tutorial, 6th Winter School on Modelling and Verifying Parallel Processes (MOVEP'04), Brussels, Belgium, December 2004.

[In-35] Jean Goubault-Larrecq. On computational interpretations of the modal logic S4. Invited talk, 2nd Workshop on the Logic for Pragmatics (WoLP'04), Créteil, France, July 2004.

[In-36] Jean Goubault-Larrecq. On cryptographic protocols, regular tree languages, and automated deduction. Invited talk, Workshop on Security of Systems: Formalism and Tools (SASYFT'04), Orleans, France, June 2004.

[In-37] David Nowak. Logical relations for monadic types. Invited talk, International Workshop on Formal Methods and Security (IWFMS'04), Nanjing, China, May 2004.

[In-38] Patricia Bouyer. Timed models for concurrent systems. Invited lecture, 32nd Spring School on Theoretical Computer Science (Concurrency Theory), Luminy, France, April 2004.

[In-39] Paul Gastin. Specifications for distributed systems. Invited lecture, 32nd Spring School on Theoretical Computer Science (Concurrency Theory), Luminy, France, April 2004.

[In-40] Patricia Bouyer. Automates temporisés, de la théorie à l'implémentation. Invited talk, Journées Formalisation des Activit?s Concurrentes (FAC'04), Toulouse, France, March 2004.

[In-41] Jean Goubault-Larrecq. Une fois qu'on n'a pas trouvé de preuve, comment le faire comprendre à un assistant de preuve ? *In* Valérie Ménissier-Morain, editor, *Actes des 15èmes Journées Francophones sur les Langages Applicatifs (JFLA'04), Sainte-Marie-de-Ré, France, January 2004*, pages 1–40. INRIA. Invited paper.

Communications dans des conférences internationales avec comité de lecture

[CI-1] Florent Bouchy, Alain Finkel, and Arnaud Sangnier. Reachability in timed counter systems. *In* Peter Habermehl and Tomáš Vojnar, editors, *Proceedings of the 10th International Workshop on Verification of Infinite State Systems (INFINITY'08), Toronto, Canada, August 2008*, Electronic Notes in Theoretical Computer Science. Elsevier Science Publishers, 2008. To appear.

[CI-2] Emmanuelle Encrenaz and Alain Finkel. Automatic verification of counter systems with ranking functions. *In* P. Madhusudan and Vineet Kahlon, editors, *Proceedings of the 9th International Workshop on Verification of Infinite State Systems (INFINITY'07), Lisbon, Portugal, September 2007*, Electronic Notes in Theoretical Computer Science. Elsevier Science Publishers, 2008. To appear.

[CI-3] Myrto Arapinis, Stéphanie Delaune, and Steve Kremer. From one session to many: Dynamic tags for security protocols. *In* Iliano Cervesato, editor, *Proceedings of the 15th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'08), Doha, Qatar, November 2008*, Lecture Notes in Artificial Intelligence. Springer. To appear.

[CI-4] Marco Beccuti, Daniele Codetta-Raiteri, Giuliana Franceschinis, and Serge Haddad. Non deterministic repairable fault trees for computing optimal repair strategy. *In* Tijani Chahed, Stavros Toumpis, and Uri Yechiali, editors, *Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS'08), Athens, Greece, October 2008*. ACM Press. To appear.

[CI-5] Elie Bursztein. Extending anticipation games with location, penalty and timeline. *In* Pierpaolo Degano, Joshua Guttman, and Fabio Martinelli, editors, *Proceedings of the 5th International Workshop on Formal Aspects in Security and Trust (FAST'08), Malaga, Spain, October 2008*, Lecture Notes in Computer Science. Springer. To appear.

[CI-6] Hubert Comon-Lundh and Véronique Cortier. Computational soundness of observational equivalence. *In Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS'08), Alexandria, Virginia, USA, October 2008*. ACM Press. To appear.

[CI-7] Étienne André, Thomas Chatain, Emmanuelle Encrenaz, and Laurent Fribourg. An inverse method for parametric timed automata. *In* Vesa Halava and Igor Potapov, editors, *Proceedings of the 2nd Workshop on Reachability Problems (RP'08), Liverpool, UK, September 2008*, Electronic Notes in Theoretical Computer Science, pages 27–43. Elsevier Science Publishers. To appear.

[CI-8] Mohamed Faouzi Atig, Benedikt Bollig, and Peter Habermehl. Emptiness of multi-pushdown automata is 2ETIME-complete. *In* Masami Ito and Masafumi Toyama, editors, *Proceedings of the 12th International Conference on Developments in Language Theory (DLT'08), Kyoto, Japan, September 2008*, volume 5257 of *Lecture Notes in Computer Science*, pages 121–133. Springer.

[CI-9] Nathalie Bertrand, Patricia Bouyer, Thomas Brihaye, and Nicolas Markey. Quantitative model-checking of one-clock timed automata under probabilistic semantics. *In Proceedings of the 5th International Conference on Quantitative Evaluation of Systems (QEST'08), Saint Malo, France, September 2008*, pages 55–64. IEEE Computer Society Press.

[CI-10] Patricia Bouyer, Thomas Brihaye, Marcin Jurdziński, Ranko Lazić, and Michał Rutkowski. Average-price and reachability-price games on hybrid automata with strong resets. *In* Franck Cassez and Claude Jard, editors, *Proceedings of the 6th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'08), Saint-Malo, France, September 2008*, volume 5215 of *Lecture Notes in Computer Science*, pages 63–77. Springer.

[CI-11] Patricia Bouyer, Uli Fahrenberg, Kim G. Larsen, Nicolas Markey, and Jiří Srba. Infinite runs in weighted timed automata with energy constraints. *In* Franck Cassez and Claude Jard, editors, *Proceedings of the 6th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'08), Saint-Malo, France, September 2008*, volume 5215 of *Lecture Notes in Computer Science*, pages 33–47. Springer.

[CI-12] Rémi Brochenin, Stéphane Demri, and Étienne Lozes. On the almighty wand. *In* Michael Kaminski and Simone Martini, editors, *Proceedings of the 16th Annual EACSL Conference on Computer Science Logic (CSL'08), Bertinoro, Italy, September 2008*, volume 5213 of *Lecture Notes in Computer Science*, pages 323–338. Springer.

[CI-13] Najla Chamseddine, Marie Duflot, Laurent Fribourg, Claudine Picaronny, and Jeremy Sproston. Computing expected absorption times for parametric determinate probabilistic timed automata. *In Proceedings of the 5th International Conference on Quantitative Evaluation of Systems (QEST'08), Saint Malo, France, September 2008*, pages 254–263. IEEE Computer Society Press.

[CI-14] Thomas Place. Characterization of logics over ranked tree languages. *In* Michael Kaminski and Simone Martini, editors, *Proceedings of the 16th Annual EACSL Conference on Computer Science Logic (CSL'08), Bertinoro, Italy, September 2008*, volume 5213 of *Lecture Notes in Computer Science*, pages 401–415. Springer.

[CI-15] S. Akshay, Benedikt Bollig, Paul Gastin, Madhavan Mukund, and K. Narayan Kumar. Distributed timed automata with independently evolving clocks. *In* Franck van Breugel and Marsha Chechik, editors, *Proceedings of the 19th International Conference on Concurrency Theory (CONCUR'08), Toronto, Canada, August 2008*, volume 5201 of *Lecture Notes in Computer Science*, pages 82–97. Springer.

[CI-16] Paolo Baldan, Thomas Chatain, Stefan Haar, and Barbara Koenig. Unfolding-based diagnosis of systems with an evolving topology. *In* Franck van Breugel and Marsha Chechik, editors, *Proceedings of the 19th International Conference on Concurrency Theory (CONCUR'08), Toronto, Canada, August 2008*, volume 5201 of *Lecture Notes in Computer Science*, pages 203–217. Springer.

[CI-17] Adel Bouhoula and Florent Jacquemard. Automated induction with constrained tree automata. *In* Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *Proceedings of the 4th International Joint Conference on Automated Reasoning (IJCAR'08), Sydney, Australia, August 2008*, volume 5195 of *Lecture Notes in Artificial Intelligence*, pages 539–553. Springer-Verlag.

[CI-18] Pierre Chambart and Philippe Schnoebelen. Mixing lossy and perfect fifo channels. *In* Franck van Breugel and Marsha Chechik, editors, *Proceedings of the 19th International Conference on Concurrency Theory (CONCUR'08), Toronto, Canada, August 2008*, volume 5201 of *Lecture Notes in Computer Science*, pages 340–355. Springer.

[CI-19] Alain Finkel and Arnaud Sangnier. Reversal-bounded counter machines revisited. *In* Edward Ochmański and Jerzy Tyszkiewicz, editors, *Proceedings of the 33rd International Symposium on Mathematical Foundations of Computer Science (MFCS'08), Toruń, Poland, August 2008*, volume 5162 of *Lecture Notes in Computer Science*, pages 323–334. Springer.

[CI-20] Steve Kremer, Antoine Mercier, and Ralf Treinen. Proving group protocols secure against eavesdroppers. *In* Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *Proceedings of the 4th International Joint Conference on Automated Reasoning (IJCAR'08), Sydney, Australia, August 2008*, volume 5195 of *Lecture Notes in Artificial Intelligence*, pages 116–131. Springer-Verlag.

[CI-21] Étienne Lozes and Jules Villard. A spatial equational logic for the applied π-calculus. *In* Franck van Breugel and Marsha Chechik, editors, *Proceedings of the 19th International Conference on Concurrency Theory (CONCUR'08), Toronto, Canada, August 2008*, volume 5201 of *Lecture Notes in Computer Science*, pages 387–401. Springer.

[CI-22] Carlos Areces, Diego Figueira, Santiago Figueira, and Sergio Mera. Expressive power and decidability for memory logics. *In* Wilfrid Hodges and Ruy de Queiroz, editors, *Proceedings of the 15th Workshop on Logic, Language, Information and Computation (WoLLIC'08), Edinburgh, Scotland, UK, July 2008*, volume 5110 of *Lecture Notes in Computer Science*, pages 56–68. Springer.

[CI-23] Mikołaj Bojańczyk and Luc Segoufin. Tree languages defined in first-order logic with one quantifier alternation. *In* Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP'08) – Part II, Reykjavik, Iceland, July 2008*, volume 5126 of *Lecture Notes in Computer Science*, pages 233–245. Springer.

[CI-24] Ahmed Bouajjani, Peter Habermehl, Lukáš Holík, Tayssir Touili, and Tomáš Vojnar. Antichain-based universality and inclusion testing over nondeterministic finite tree automata. *In* Oscar H. Ibarra and Bala Ravikumar, editors, *Proceedings of the 13th International Conference on Implementation and Application of Automata (CIAA'08), San Francisco, California, USA, July 2008*, volume 5148 of *Lecture Notes in Computer Science*, pages 57–67. Springer-Verlag.

[CI-25] Patricia Bouyer, Nicolas Markey, Joël Ouaknine, and James Worrell. On expressiveness and complexity in real-time model checking. *In* Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP'08) – Part II, Reykjavik, Iceland, July 2008*, volume 5126 of *Lecture Notes in Computer Science*, pages 124–135. Springer.

[CI-26] Adrià Gascón, Guillem Godoy, and Florent Jacquemard. Closure of tree automata languages under innermost rewriting. *In* Aart Middeldorp, editor, *Proceedings of the 8th International Workshop on Reduction Strategies in Rewriting and Programming (WRS'08), Castle of Hagenberg, Austria, July 2008*, Electronic Notes in Theoretical Computer Science. Elsevier Science Publishers. To appear.

[CI-27] Florent Jacquemard and Michaël Rusinowitch. Closure of Hedge-automata languages by Hedge rewriting. *In* Andrei Voronkov, editor, *Proceedings of the 19th International Conference on Rewriting Techniques and Applications (RTA'08), Hagenberg, Austria, July 2008*, volume 5117 of *Lecture Notes in Computer Science*, pages 157–171. Springer.

[CI-28] Serge Abiteboul, Luc Segoufin, and Victor Vianu. Static analysis of active XML services. *In* Maurizio Lenzerini and Domenico Lembo, editors, *Proceedings of the 26th Annual ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'08), Vancouver, Canada, June 2008*, pages 221–230. ACM Press.

[CI-29] Christel Baier, Nathalie Bertrand, Patricia Bouyer, Thomas Brihaye, and Marcus Größer. Almost-sure model checking of infinite paths in one-clock timed automata. *In Proceedings of the 23rd Annual IEEE Symposium on Logic in Computer Science (LICS'08), Pittsburgh, PA, USA, June 2008*, pages 217–226. IEEE Computer Society Press.

[CI-30] Mikołaj Bojańczyk, Luc Segoufin, and Howard Straubing. Piecewise testable tree languages. *In Proceedings of the 23rd Annual IEEE Symposium on Logic in Computer Science (LICS'08), Pittsburgh, PA, USA, June 2008*, pages 442–451. IEEE Computer Society Press.

[CI-31] Florent Bouchy, Alain Finkel, and Jérôme Leroux. Decomposition of decidable first-order logics over integers and reals. *In* Stéphane Demri and Christian S. Jensen, editors, *Proceedings of the 15th International Symposium on Temporal Representation and Reasoning (TIME'08), Montréal, Canada, June 2008*, pages 147–155. IEEE Computer Society Press.

[CI-32] Thomas Brihaye, Mohamed Ghannem, Nicolas Markey, and Lionel Rieg. Good friends are hard to find! *In* Stéphane Demri and Christian S. Jensen, editors, *Proceedings of the 15th International Symposium on Temporal Representation and Reasoning (TIME'08), Montréal, Canada, June 2008*, pages 32–40. IEEE Computer Society Press.

[CI-33] Pierre Chambart and Philippe Schnoebelen. The ordinal recursive complexity of lossy channel systems. *In Proceedings of the 23rd Annual IEEE Symposium on Logic in Computer Science (LICS'08), Pittsburgh, PA, USA, June 2008*, pages 205–216. IEEE Computer Society Press.

[CI-34] Stéphanie Delaune, Steve Kremer, and Mark D. Ryan. Composition of password-based protocols. *In Proceedings of the 21st IEEE Computer Security Foundations Symposium (CSF'08), Pittsburgh, PA, USA, June 2008*, pages 239–251. IEEE Computer Society Press.

[CI-35] Stéphanie Delaune, Steve Kremer, and Graham Steel. Formal analysis of PKCS#11. *In Proceedings of the 21st IEEE Computer Security Foundations Symposium (CSF'08), Pittsburgh, PA, USA, June 2008*, pages 331–344. IEEE Computer Society Press.

[CI-36] Stéphanie Delaune, Mark D. Ryan, and Ben Smyth. Automatic verification of privacy properties in the applied pi-calculus. *In* Yuecel Karabulut, John Mitchell, Peter Herrmann, and Christian Damsgaard Jensen, editors, *Proceedings of the 2nd Joint iTrust and PST Conferences on Privacy, Trust Management and Security (IFIPTM'08), Trondheim, Norway, June 2008*, volume 263 of *IFIP Conference Proceedings*, pages 263–278. Springer.

[CI-37] Jean Goubault-Larrecq. Towards producing formally checkable security proofs, automatically. *In Proceedings of the 21st IEEE Computer Security Foundations Symposium (CSF'08), Pittsburgh, PA, USA, June 2008*, pages 224–238. IEEE Computer Society Press.

[CI-38] Balder ten Cate and Luc Segoufin. XPath, transitive closure logic, and nested tree walking automata. *In* Maurizio Lenzerini and Domenico Lembo, editors, *Proceedings of the 26th Annual ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'08), Vancouver, Canada, June 2008*, pages 251–260. ACM Press.

[CI-39] Béatrice Bérard, Serge Haddad, Lom Messan Hillah, Fabrice Kordon, and Yann Tierry-Mieg. Collision avoidance in intelligent transport systems: Towards an application of control theory. *In Proceedings of the 9th Workshop on Discrete Event Systems (WODES'08), Gothenburg, Sweden, May 2008*, pages 346–351.

[CI-40] Elie Bursztein. Probabilistic protocol identification for hard to classify protocol. *In* Jose A. Onieva, Damien Sauveron, Serge Chaumette, Dieter Gollmann, and Konstantinos Markantonakis, editors, *Proceedings of the 2nd International Workshop on Information Security Theory and Practices (WISTP'08), Sevilla, Spain, May 2008*, volume 5019 of *Lecture Notes in Computer Science*, pages 49–63. Springer. Best paper award.

[CI-41] Emmanuelle Encrenaz and Laurent Fribourg. Time separation of events: An inverse method. *In* Catuscia Palamidessi and Franck Valencia, editors, *Proceedings of the LIX Colloquium on Emerging Trends in Concurrency Theory (LIX'06), Palaiseau, France, November 2006*, volume 209 of *Electronic Notes in Theoretical Computer Science*, pages 135–148. Elsevier Science Publishers, April 2008.

[CI-42] Patricia Bouyer, Nicolas Markey, and Pierre-Alain Reynier. Robust analysis of timed automata *via* channel machines. *In* Roberto Amadio, editor, *Proceedings of the 11th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'08), Budapest, Hungary, March-April 2008*, volume 4962 of *Lecture Notes in Computer Science*, pages 157–171. Springer.

[CI-43] Pierre Chambart and Philippe Schnoebelen. The ω-regular Post embedding problem. *In* Roberto Amadio, editor, *Proceedings of the 11th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'08), Budapest, Hungary, March-April 2008*, volume 4962 of *Lecture Notes in Computer Science*, pages 97–111. Springer.

[CI-44] Stéphane Demri, Ranko Lazić, and Arnaud Sangnier. Model checking freeze LTL over one-counter automata. *In* Roberto Amadio, editor, *Proceedings of the 11th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'08), Budapest, Hungary, March-April 2008*, volume 4962 of *Lecture Notes in Computer Science*, pages 490–504. Springer.

[CI-45] Jean Goubault-Larrecq. Prevision domains and convex powercones. *In* Roberto Amadio, editor, *Proceedings of the 11th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'08), Budapest, Hungary, March-April 2008*, volume 4962 of *Lecture Notes in Computer Science*, pages 318–333. Springer.

[CI-46] Jean Goubault-Larrecq. Simulation hemi-metrics between infinite-state stochastic games. *In* Roberto Amadio, editor, *Proceedings of the 11th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'08), Budapest, Hungary, March-April 2008*, volume 4962 of *Lecture Notes in Computer Science*, pages 50–65. Springer.

[CI-47] Peter Habermehl, Radu Iosif, and Tomáš Vojnar. What else is decidable about arrays? *In* Roberto Amadio, editor, *Proceedings of the 11th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'08), Budapest, Hungary, March-April 2008*, volume 4962 of *Lecture Notes in Computer Science*, pages 474–489. Springer.

[CI-48] Patricia Bouyer, Nicolas Markey, Joël Ouaknine, Philippe Schnoebelen, and James Worrell. On termination for faulty channel machines. *In* Susanne Albers and Pascal Weil, editors, *Proceedings of the 25th Annual Symposium on Theoretical Aspects of Computer Science (STACS'08), Bordeaux, France, February 2008*, pages 121–132.

[CI-49] Vincent Bernat and Hubert Comon-Lundh. Normal proofs in intruder theories. *In* Mitsu Okada and Ichiro Satoh, editors, *Revised Selected Papers of the 11th Asian Computing Science Conference (ASIAN'06), Tokyo, Japan, December 2006*, volume 4435 of *Lecture Notes in Computer Science*, pages 151–166. Springer, January 2008.

[CI-50] S. Akshay, Benedikt Bollig, and Paul Gastin. Automata and logics for timed message sequence charts. *In* V. Arvind and Sanjiva Prasad, editors, *Proceedings of the 27th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'07), New Delhi, India, December 2007*, volume 4855 of *Lecture Notes in Computer Science*, pages 290–302. Springer.

[CI-51] Christel Baier, Nathalie Bertrand, Patricia Bouyer, Thomas Brihaye, and Marcus Größer. Probabilistic and topological semantics for timed automata. *In* V. Arvind and Sanjiva Prasad, editors, *Proceedings of the 27th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'07), New Delhi, India, December 2007*, volume 4855 of *Lecture Notes in Computer Science*, pages 179–191. Springer.

[CI-52] Benedikt Bollig, Dietrich Kuske, and Ingmar Meinecke. Propositional dynamic logic for message-passing systems. *In* V. Arvind and Sanjiva Prasad, editors, *Proceedings of the 27th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'07), New Delhi, India, December 2007*, volume 4855 of *Lecture Notes in Computer Science*, pages 303–315. Springer.

[CI-53] Elie Bursztein and Jean Goubault-Larrecq. A logical framework for evaluating network resilience against faults and attacks. *In* Iliano Cervesato, editor, *Proceedings of the 12th Asian Computing Science Conference (ASIAN'07), Doha, Qatar, December 2007*, volume 4846 of *Lecture Notes in Computer Science*, pages 212–227. Springer.

[CI-54] Pierre Chambart and Philippe Schnoebelen. Post embedding problem is not primitive recursive, with applications to channel systems. *In* V. Arvind and Sanjiva Prasad, editors, *Proceedings of the 27th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'07), New Delhi, India, December 2007*, volume 4855 of *Lecture Notes in Computer Science*, pages 265–276. Springer.

[CI-55] Véronique Cortier, Jérémie Delaitre, and Stéphanie Delaune. Safely composing security protocols. *In* V. Arvind and Sanjiva Prasad, editors, *Proceedings of the 27th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'07), New Delhi, India, December 2007*, volume 4855 of *Lecture Notes in Computer Science*, pages 352–363. Springer.

[CI-56] Stéphanie Delaune, Steve Kremer, and Mark D. Ryan. Symbolic bisimulation for the applied pi-calculus. *In* V. Arvind and Sanjiva Prasad, editors, *Proceedings of the 27th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'07), New Delhi, India, December 2007*, volume 4855 of *Lecture Notes in Computer Science*, pages 133–145. Springer.

[CI-57] Jean Goubault-Larrecq, Catuscia Palamidessi, and Angelo Troina. A probabilistic applied pi-calculus. *In* Zhong Shao, editor, *Proceedings of the 5th Asian Symposium on Programming Languages and Systems (APLAS'07), Singapore, November-December 2007*, volume 4807 of *Lecture Notes in Computer Science*, pages 175–290. Springer.

[CI-58] Alessandro Abate, Yu Bai, Nathalie Sznajder, Carolyn Talcott, and Ashish Tiwari. Quantitative and probabilisitic modeling in pathway logic. *In Proceedings of the IEEE 7th International Symposium on Bioinformatics and Bioengineering (BIBE'07), Boston, Massachusetts, USA, October 2007*, pages 922–929. IEEE Computer Society Press.

[CI-59] Patricia Bouyer and Nicolas Markey. Costs are expensive! *In* Jean-François Raskin and P. S. Thiagarajan, editors, *Proceedings of the 5th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'07), Salzburg, Austria, October 2007*, volume 4763 of *Lecture Notes in Computer Science*, pages 53–68. Springer.

[CI-60] Fabrice Chevalier, Deepak D'Souza, and Pavithra Prabhakar. Counter-free input determined timed automata. *In* Jean-François Raskin and P. S. Thiagarajan, editors, *Proceedings of the 5th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'07), Salzburg, Austria, October 2007*, volume 4763 of *Lecture Notes in Computer Science*, pages 82–97. Springer.

[CI-61] Véronique Cortier and Stéphanie Delaune. Deciding knowledge in security protocols for monoidal equational theories. *In* Nachum Dershowitz and Andrei Voronkov, editors, *Proceedings of the 14th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'07), Yerevan, Armenia, October 2007*, volume 4790 of *Lecture Notes in Artificial Intelligence*, pages 196–210. Springer.

[CI-62] Stéphanie Delaune, Hai Lin, and Christopher Lynch. Protocol verification via rigid/ flexible resolution. *In* Nachum Dershowitz and Andrei Voronkov, editors, *Proceedings of the 14th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'07), Yerevan, Armenia, October 2007*, volume 4790 of *Lecture Notes in Artificial Intelligence*, pages 242–256. Springer.

[CI-63] Stéphane Demri and Alexander Rabinovich. The complexity of temporal logic with until and since over ordinals. *In* Nachum Dershowitz and Andrei Voronkov, editors, *Proceedings of the 14th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'07), Yerevan, Armenia, October 2007*, volume 4790 of *Lecture Notes in Artificial Intelligence*, pages 531–545. Springer.

[CI-64] Peter Habermehl, Radu Iosif, Adam Rogalewicz, and Tomáš Vojnar. Proving termination of tree manipulating programs. *In* Kedar Namjoshi and Tomohiro Yoneda, editors, *Proceedings of the 5th International Symposium on Automated Technology for Verification and Analysis (ATVA'07), Tokyo, Japan, October 2007*, volume 4762 of *Lecture Notes in Computer Science*, pages 145–161. Springer.

[CI-65] S. Akshay, Madhavan Mukund, and K. Narayan Kumar. Checking coverage for infinite collections of timed scenarios. *In* Luís Caires and Vasco T. Vasconcelos, editors, *Proceedings of the 18th International Conference on Concurrency Theory (CONCUR'07), Lisbon, Portugal, September 2007*, volume 4703 of *Lecture Notes in Computer Science*, pages 181–196. Springer.

[CI-66] Mathilde Arnaud, Véronique Cortier, and Stéphanie Delaune. Combining algorithms for deciding knowledge in security protocols. *In* Franck Wolter, editor, *Proceedings of the 6th International Symposium on Frontiers of Combining Systems (FroCoS'07), Liverpool, UK, September 2007*, volume 4720 of *Lecture Notes in Artificial Intelligence*, pages 103–117. Springer.

[CI-67] Thomas Brihaye, François Laroussinie, Nicolas Markey, and Ghassan Oreiby. Timed concurrent game structures. *In* Luís Caires and Vasco T. Vasconcelos, editors, *Proceedings of the 18th International Conference on Concurrency Theory (CONCUR'07), Lisbon, Portugal, September 2007*, volume 4703 of *Lecture Notes in Computer Science*, pages 445–459. Springer.

[CI-68] Jean Goubault-Larrecq. Continuous previsions. *In* Jacques Duparc and Thomas A. Henzinger, editors, *Proceedings of the 16th Annual EACSL Conference on Computer Science Logic (CSL'07), Lausanne, Switzerland, September 2007*, volume 4646 of *Lecture Notes in Computer Science*, pages 542–557. Springer.

[CI-69] Steve Kremer and Laurent Mazaré. Adaptive soundness of static equivalence. *In* Joachim Biskup and Javier Lopez, editors, *Proceedings of the 12th European Symposium on Research in Computer Security (ESORICS'07), Dresden, Germany, September 2007*, volume 4734 of *Lecture Notes in Computer Science*, pages 610–625. Springer.

[CI-70] Puneet Bhateja, Paul Gastin, Madhavan Mukund, and K. Narayan Kumar. Local testing of message sequence charts is difficult. *In* Erzsébet Csuhaj-Varjú and Zoltán Ésik, editors, *Proceedings of the 16th International Symposium on Fundamentals of Computation Theory (FCT'07), Budapest, Hungary, August 2007*, volume 4639 of *Lecture Notes in Computer Science*, pages 76–87. Springer.

[CI-71] Ahmed Bouajjani, Peter Habermehl, Yan Jurski, and Mihaela Sighireanu. Rewriting systems with data – A framework for reasoning about systems with unbounded structures over infinite data domains. *In* Erzsébet Csuhaj-Varjú and Zoltán Ésik, editors, *Proceedings of the 16th International Symposium on Fundamentals of Computation Theory (FCT'07), Budapest, Hungary, August 2007*, volume 4639 of *Lecture Notes in Computer Science*, pages 1–22. Springer.

[CI-72] Patricia Bouyer, Nicolas Markey, Joël Ouaknine, and James Worrell. The cost of punctuality. *In Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science (LICS'07), Wrocław, Poland, July 2007*, pages 109–118. IEEE Computer Society Press.

[CI-73] Cécile Braunstein and Emmanuelle Encrenaz. Using CTL formulae as component abstraction in a design and verification flow. *In* Twan Basten and Sandeep Shukla, editors, *Proceedings of the 7th International Conference on Application of Concurrency to System Design (ACSD'07), Bratislava, Slovak Republik, July 2007*, pages 80–89. IEEE Computer Society Press.

[CI-74] Thomas Brihaye, Thomas A. Henzinger, Vinayak Prabhu, and Jean-François Raskin. Minimum-time reachability in timed games. *In* Lars Arge, Christian Cachin, Tomasz Jurdziński, and Andrzej Tarlecki, editors, *Proceedings of the 34th International Colloquium on Automata, Languages and Programming (ICALP'07), Wrocław, Poland, July 2007*, volume 4596 of *Lecture Notes in Computer Science*, pages 825–837. Springer.

[CI-75] Véronique Cortier, Stéphanie Delaune, and Graham Steel. A formal theory of key conjuring. *In Proceedings of the 20th IEEE Computer Security Foundations Symposium (CSF'07), Venice, Italy, July 2007*, pages 79–93. IEEE Computer Society Press.

[CI-76] Paul Gastin and Pierre Moro. Minimal counter-example generation for SPIN. *In* Dragan Bošnački and Stefan Edelkamp, editors, *Proceedings of the 14th International SPIN*

Workshop on Model Checking Software (SPIN'07), Berlin, Germany, July 2007, volume 4595 of *Lecture Notes in Computer Science*, pages 24–38. Springer.

[CI-77] Jean Goubault-Larrecq. Continuous capacities on continuous state spaces. *In* Lars Arge, Christian Cachin, Tomasz Jurdziński, and Andrzej Tarlecki, editors, *Proceedings of the 34th International Colloquium on Automata, Languages and Programming (ICALP'07), Wrocław, Poland, July 2007*, volume 4596 of *Lecture Notes in Computer Science*, pages 764–776. Springer.

[CI-78] Jean Goubault-Larrecq. On Noetherian spaces. *In Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science (LICS'07), Wrocław, Poland, July 2007*, pages 453–462. IEEE Computer Society Press.

[CI-79] Pascal Lafourcade. Intruder deduction for the equational theory of *exclusive-or* with commutative and distributive encryption. *In* Maribel Fernández and Claude Kirchner, editors, *Proceedings of the 1st International Workshop on Security and Rewriting Techniques (SecReT'06), Venice, Italy, July 2006*, volume 171(4) of *Electronic Notes in Theoretical Computer Science*, pages 37–57. Elsevier Science Publishers, July 2007.

[CI-80] Benedikt Bollig and Ingmar Meinecke. Weighted distributed systems and their logics. *In* Sergei N. Artemov and Anil Nerode, editors, *Proceedings of the Symposium on Logical Foundations of Computer Science (LFCS'07), New-York, NY, USA, June 2007*, volume 4514 of *Lecture Notes in Computer Science*, pages 54–68. Springer.

[CI-81] Patricia Bouyer, Thomas Brihaye, and Fabrice Chevalier. Weighted o-minimal hybrid systems are more decidable than weighted timed automata! *In* Sergei N. Artemov and Anil Nerode, editors, *Proceedings of the Symposium on Logical Foundations of Computer Science (LFCS'07), New-York, NY, USA, June 2007*, volume 4514 of *Lecture Notes in Computer Science*, pages 69–83. Springer.

[CI-82] Rémi Brochenin, Stéphane Demri, and Étienne Lozes. Reasoning about sequences of memory states. *In* Sergei N. Artemov and Anil Nerode, editors, *Proceedings of the Symposium on Logical Foundations of Computer Science (LFCS'07), New-York, NY, USA, June 2007*, volume 4514 of *Lecture Notes in Computer Science*, pages 100–114. Springer.

[CI-83] Sergiu Bursuc, Hubert Comon-Lundh, and Stéphanie Delaune. Deducibility constraints, equational theory and electronic money. *In* Hubert Comon-Lundh, Claude Kirchner, and Hélène Kirchner, editors, *Rewriting, Computation and Proof — Essays Dedicated to Jean-Pierre Jouannaud on the Occasion of his 60th Birthday, Cachan, France, June 2007*, volume 4600 of *Lecture Notes in Computer Science*, pages 196–212. Springer.

[CI-84] Stéphane Demri, Deepak D'Souza, and Régis Gascon. Decidable temporal logic with repeating values. *In* Sergei N. Artemov and Anil Nerode, editors, *Proceedings of the Symposium on Logical Foundations of Computer Science (LFCS'07), New-York, NY, USA, June 2007*, volume 4514 of *Lecture Notes in Computer Science*, pages 180–194. Springer.

[CI-85] Stéphane Demri and Régis Gascon. The effects of bounding syntactic resources on Presburger LTL (extended abstract). *In* Valentin Goranko and X. Sean Wang, editors, *Proceedings of the 14th International Symposium on Temporal Representation and Reasoning (TIME'07), Alicante, Spain, June 2007*, pages 94–104. IEEE Computer Society Press.

[CI-86] Benedikt Bollig, Joost-Pieter Katoen, Carsten Kern, and Martin Leucker. Replaying play in and play out: Synthesis of design models from scenarios by learning. *In* Orna Grumberg and Michael Huth, editors, *Proceedings of the 13th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'07), Braga, Portugal, March 2007*, volume 4424 of *Lecture Notes in Computer Science*, pages 435–450. Springer.

[CI-87] Patricia Bouyer, Kim G. Larsen, and Nicolas Markey. Model-checking one-clock priced timed automata. *In* Helmut Seidl, editor, *Proceedings of the 10th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'07), Braga, Portugal, March 2007*, volume 4423 of *Lecture Notes in Computer Science*, pages 108–122. Springer.

[CI-88] Hubert Comon-Lundh, Florent Jacquemard, and Nicolas Perrin. Tree automata with memory, visibility and structural constraints. *In* Helmut Seidl, editor, *Proceedings of the 10th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'07), Braga, Portugal, March 2007*, volume 4423 of *Lecture Notes in Computer Science*, pages 168–182. Springer.

[CI-89] Davide D'Aprile, Susanna Donatelli, Arnaud Sangnier, and Jeremy Sproston. From time Petri nets to timed automata: An untimed approach. *In* Orna Grumberg and Michael Huth, editors, *Proceedings of the 13th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'07), Braga, Portugal, March 2007*, volume 4424 of *Lecture Notes in Computer Science*, pages 216–230. Springer.

[CI-90] Marcin Jurdziński, François Laroussinie, and Jeremy Sproston. Model checking probabilistic timed automata with one or two clocks. *In* Orna Grumberg and Michael Huth, editors, *Proceedings of the 13th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'07), Braga, Portugal, March 2007*, volume 4424 of *Lecture Notes in Computer Science*, pages 170–184. Springer.

[CI-91] François Laroussinie, Nicolas Markey, and Ghassan Oreiby. On the expressiveness and complexity of ATL. *In* Helmut Seidl, editor, *Proceedings of the 10th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'07), Braga, Portugal, March 2007*, volume 4423 of *Lecture Notes in Computer Science*, pages 243–257. Springer.

[CI-92] Sergiu Bursuc, Hubert Comon-Lundh, and Stéphanie Delaune. Associative-commutative deducibility constraints. *In* Wolfgang Thomas and Pascal Weil, editors, *Proceedings of the 24th Annual Symposium on Theoretical Aspects of Computer Science (STACS'07), Aachen, Germany, February 2007*, volume 4393 of *Lecture Notes in Computer Science*, pages 634–645. Springer.

[CI-93] Patricia Bouyer, Kim G. Larsen, Nicolas Markey, and Jacob Illum Rasmussen. Almost optimal strategies in one-clock priced timed automata. *In* Naveen Garg and S. Arun-Kumar, editors, *Proceedings of the 26th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'06), Kolkata, India, December 2006*, volume 4337 of *Lecture Notes in Computer Science*, pages 345–356. Springer.

[CI-94] Laura Bozzelli, Mojmír Křetínský, Vojtěch Řehák, and Jan Strejček. On decidability of LTL model checking for process rewrite systems. *In* Naveen Garg and S. Arun-Kumar, editors, *Proceedings of the 26th Conference on Foundations of Software Technology and*

Theoretical Computer Science (FSTTCS'06), Kolkata, India, December 2006, volume 4337 of *Lecture Notes in Computer Science*, pages 248–259. Springer.

[CI-95] Fabrice Chevalier, Deepak D'Souza, and Pavithra Prabhakar. On continuous timed automata with input-determined guards. *In* Naveen Garg and S. Arun-Kumar, editors, *Proceedings of the 26th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'06), Kolkata, India, December 2006*, volume 4337 of *Lecture Notes in Computer Science*, pages 369–380. Springer.

[CI-96] Véronique Cortier, Steve Kremer, Ralf Küsters, and Bogdan Warinschi. Computationally sound symbolic secrecy in the presence of hash functions. *In* Naveen Garg and S. Arun-Kumar, editors, *Proceedings of the 26th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'06), Kolkata, India, December 2006*, volume 4337 of *Lecture Notes in Computer Science*, pages 176–187. Springer.

[CI-97] Paul Gastin, Nathalie Sznajder, and Marc Zeitoun. Distributed synthesis for well-connected architectures. *In* Naveen Garg and S. Arun-Kumar, editors, *Proceedings of the 26th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'06), Kolkata, India, December 2006*, volume 4337 of *Lecture Notes in Computer Science*, pages 321–332. Springer.

[CI-98] Christel Baier, Nathalie Bertrand, and Philippe Schnoebelen. On computing fixpoints in well-structured regular model checking, with applications to lossy channel systems. *In* Miki Hermann and Andrei Voronkov, editors, *Proceedings of the 13th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'06), Phnom Penh, Cambodia, November 2006*, volume 4246 of *Lecture Notes in Artificial Intelligence*, pages 347–361. Springer.

[CI-99] Laura Bozzelli and Régis Gascon. Branching time temporal logic extended with Presburger constraints. *In* Miki Hermann and Andrei Voronkov, editors, *Proceedings of the 13th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'06), Phnom Penh, Cambodia, November 2006*, volume 4246 of *Lecture Notes in Artificial Intelligence*, pages 197–211. Springer.

[CI-100] Sami Taktak, Emmanuelle Encrenaz, and Jean-Lou Desbarbieux. A tool for automatic detection of deadlock in wormhole networks on chip. *In Proceedings of the IEEE High Level Design Verification and Test Workshop (HLDVT'06), Monterey, California, USA, November 2006*, pages 203–210. IEEE Computer Society Press.

[CI-101] Houda Bel mokadem, Béatrice Bérard, Patricia Bouyer, and François Laroussinie. Timed temporal logics for abstracting transient states. *In* Susanne Graf and Wenhui Zhang, editors, *Proceedings of the 4th International Symposium on Automated Technology for Verification and Analysis (ATVA'06), Beijing, ROC, October 2006*, volume 4218 of *Lecture Notes in Computer Science*, pages 337–351. Springer.

[CI-102] Puneet Bhateja, Paul Gastin, and Madhavan Mukund. A fresh look at testing for asynchronous communication. *In* Susanne Graf and Wenhui Zhang, editors, *Proceedings of the 4th International Symposium on Automated Technology for Verification and Analysis (ATVA'06), Beijing, ROC, October 2006*, volume 4218 of *Lecture Notes in Computer Science*, pages 369–383. Springer.

[CI-103] Patricia Bouyer, Serge Haddad, and Pierre-Alain Reynier. Timed unfoldings for networks of timed automata. *In* Susanne Graf and Wenhui Zhang, editors, *Proceedings of the 4th International Symposium on Automated Technology for Verification and Analysis (ATVA'06), Beijing, ROC, October 2006*, volume 4218 of *Lecture Notes in Computer Science*, pages 292–306. Springer.

[CI-104] Stéphane Demri, Alain Finkel, Valentin Goranko, and Govert van Drimmelen. Towards a model-checker for counter systems. *In* Susanne Graf and Wenhui Zhang, editors, *Proceedings of the 4th International Symposium on Automated Technology for Verification and Analysis (ATVA'06), Beijing, ROC, October 2006*, volume 4218 of *Lecture Notes in Computer Science*, pages 493–507. Springer.

[CI-105] Christel Baier, Nathalie Bertrand, and Philippe Schnoebelen. Symbolic verification of communicating systems with probabilistic message losses: liveness and fairness. *In* Elie Najm, Jean-François Pradat-Peyre, and Véronique Viguié Donzeau-Gouge, editors, *Proceedings of 26th IFIP WG6.1 International Conference on Formal Techniques for Networked and Distributed Systems (FORTE'06), Paris, France, September 2006*, volume 4229 of *Lecture Notes in Computer Science*, pages 212–227. Springer.

[CI-106] Béatrice Bérard, Paul Gastin, and Antoine Petit. Intersection of regular signal-event (timed) languages. *In* Eugène Asarin and Patricia Bouyer, editors, *Proceedings of the 4th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'06), Paris, France, September 2006*, volume 4202 of *Lecture Notes in Computer Science*, pages 52–66. Springer.

[CI-107] Béatrice Bérard, Paul Gastin, and Antoine Petit. Refinements and abstractions of signal-event (timed) languages. *In* Eugène Asarin and Patricia Bouyer, editors, *Proceedings of the 4th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'06), Paris, France, September 2006*, volume 4202 of *Lecture Notes in Computer Science*, pages 67–81. Springer.

[CI-108] Rémy Chevallier, Emmanuelle Encrenaz-Tiphène, Laurent Fribourg, and Weiwen Xu. Verification of the generic architecture of a memory circuit using parametric timed automata. *In* Eugène Asarin and Patricia Bouyer, editors, *Proceedings of the 4th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'06), Paris, France, September 2006*, volume 4202 of *Lecture Notes in Computer Science*, pages 113–127. Springer.

[CI-109] François Laroussinie, Nicolas Markey, and Ghassan Oreiby. Model checking timed ATL for durational concurrent game structures. *In* Eugène Asarin and Patricia Bouyer, editors, *Proceedings of the 4th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'06), Paris, France, September 2006*, volume 4202 of *Lecture Notes in Computer Science*, pages 245–259. Springer.

[CI-110] Fabio Mancinelli, Jaap Boender, Roberto Di Cosmo, Jérôme Vouillon, Berke Durak, Xavier Leroy, and Ralf Treinen. Managing the complexity of large free and open source package-based software distributions. *In Proceedings of the 21st IEEE/ACM International Conference on Automated Software Engineering (ASE'06), Tokyo, Japan, September 2006*, pages 199–208. IEEE Computer Society Press.

[CI-111] Ichiro Mitsuhashi, Michio Oyamaguchi, and Florent Jacquemard. The confluence problem for flat TRSs. *In* Jacques Calmet, Tetsuo Ida, and Dongming Wang, editors, *Proceedings*

of the 8th International Conference on Artificial Intelligence and Symbolic Computation (AISC'06), Beijing, China, September 2006, volume 4120 of *Lecture Notes in Artificial Intelligence*, pages 68–81. Springer.

[CI-112] Manuel Baclet and Claire Pagetti. Around Hopcroft's algorithm. *In* Oscar H. Ibarra and Hsu-Chun Yen, editors, *Proceedings of the 11th International Conference on Implementation and Application of Automata (CIAA'06), Taipei, Taiwan, ROC, August 2006*, volume 4094 of *Lecture Notes in Computer Science*, pages 114–125. Springer-Verlag.

[CI-113] Sébastien Bardin, Jérôme Leroux, and Gérald Point. FAST Extended Release. *In* Thomas Ball and Robert B. Jones, editors, *Proceedings of the 18th International Conference on Computer Aided Verification (CAV'06), Seattle, Washington, USA, August 2006*, volume 4144 of *Lecture Notes in Computer Science*, pages 63–66. Springer.

[CI-114] Patricia Bouyer, Laura Bozzelli, and Fabrice Chevalier. Controller synthesis for MTL specifications. *In* Christel Baier and Holger Hermanns, editors, *Proceedings of the 17th International Conference on Concurrency Theory (CONCUR'06), Bonn, Germany, August 2006*, volume 4137 of *Lecture Notes in Computer Science*, pages 450–464. Springer.

[CI-115] Patricia Bouyer, Thomas Brihaye, and Fabrice Chevalier. Control in o-minimal hybrid systems. *In Proceedings of the 21st Annual IEEE Symposium on Logic in Computer Science (LICS'06), Seattle, Washington, USA, August 2006*, pages 367–378. IEEE Computer Society Press.

[CI-116] Stéphane Demri and Ranko Lazić. LTL with the freeze quantifier and register automata. *In Proceedings of the 21st Annual IEEE Symposium on Logic in Computer Science (LICS'06), Seattle, Washington, USA, August 2006*, pages 17–26. IEEE Computer Society Press.

[CI-117] Stéphane Demri and Denis Lugiez. Presburger modal logic is only PSPACE-complete. *In* Ulrich Furbach and Natarajan Shankar, editors, *Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR'06), Seattle, Washington, USA, August 2006*, volume 4130 of *Lecture Notes in Artificial Intelligence*, pages 541–556. Springer-Verlag.

[CI-118] Florent Jacquemard, Michaël Rusinowitch, and Laurent Vigneron. Tree automata with equality constraints modulo equational theories. *In* Ulrich Furbach and Natarajan Shankar, editors, *Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR'06), Seattle, Washington, USA, August 2006*, volume 4130 of *Lecture Notes in Artificial Intelligence*, pages 557–571. Springer-Verlag.

[CI-119] Nathalie Bertrand and Philippe Schnoebelen. A short visit to the STS hierarchy. *In* Jos Baeten and Iain Phillips, editors, *Proceedings of the 12th International Workshop on Expressiveness in Concurrency (EXPRESS'05), San Francisco, CA, USA, August 2005*, volume 154(3) of *Electronic Notes in Theoretical Computer Science*, pages 59–69. Elsevier Science Publishers, July 2006.

[CI-120] Patricia Bouyer, Serge Haddad, and Pierre-Alain Reynier. Timed Petri nets and timed automata: On the discriminating power of Zeno sequences. *In* Michele Buglesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP'06) —*

Part II, Venice, Italy, July 2006, volume 4052 of *Lecture Notes in Computer Science*, pages 420–431. Springer.

[CI-121] Stéphanie Delaune, Steve Kremer, and Mark D. Ryan. Coercion-resistance and receipt-freeness in electronic voting. *In Proceedings of the 19th IEEE Computer Security Foundations Workshop (CSFW'06), Venice, Italy, July 2006*, pages 28–39. IEEE Computer Society Press.

[CI-122] Stéphanie Delaune, Pascal Lafourcade, Denis Lugiez, and Ralf Treinen. Symbolic protocol analysis in presence of a homomorphism operator and *exclusive or*. *In* Michele Buglesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP'06) — Part II, Venice, Italy, July 2006*, volume 4052 of *Lecture Notes in Computer Science*, pages 132–143. Springer.

[CI-123] Michel Bidoit and Rolf Hennicker. Proving behavioral refinements of COL-specifications. *In* Kokichi Futatsugi, Jean-Pierre Jouannaud, and José Meseguer, editors, *Algebra, Meaning and Computation — Essays dedicated to Joseph A. Goguen on the Occasion of His 65th Birthday, San Diego, California, USA, June 2006*, volume 4060 of *Lecture Notes in Computer Science*, pages 333–354. Springer.

[CI-124] Patricia Bouyer, Serge Haddad, and Pierre-Alain Reynier. Extended timed automata and time Petri nets. *In* Kees Goossens and Laure Petrucci, editors, *Proceedings of the 6th International Conference on Application of Concurrency to System Design (ACSD'06), Turku, Finland, June 2006*, pages 91–100. IEEE Computer Society Press.

[CI-125] Cécile Braunstein and Emmanuelle Encrenaz. Formalizing the incremental design and verification process of a pipelined protocol converter. *In Proceedings of the 17th International Workshop on Rapid System Prototyping (RSP'06), Chania, Crete, June 2006*, pages 103–109. IEEE Computer Society Press.

[CI-126] Jaap Boender, Roberto Di Cosmo, Berke Durak, Xavier Leroy, Fabio Mancinelli, Mario Morgado, David Pinheiro, Ralf Treinen, Paulo Trezentos, and Jérôme Vouillon. News from the EDOS project: improving the maintenance of free software distributions. *In* Olivier Berger, editor, *Proceedings of the International Workshop on Free Software (IWFS'06), Porto Allegre, Brazil, April 2006*, pages 199–207.

[CI-127] Martín Abadi, Mathieu Baudet, and Bogdan Warinschi. Guessing attacks and the computational soundness of static equivalence. *In* Luca Aceto and Anna Ingólfsdóttir, editors, *Proceedings of the 9th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'06), Vienna, Austria, March 2006*, volume 3921 of *Lecture Notes in Computer Science*, pages 398–412. Springer.

[CI-128] Benedikt Bollig, Carsten Kern, Markus Schlütter, and Volker Stolz. MSCan: A tool for analyzing MSC specifications. *In* Holger Hermanns and Jens Palsberg, editors, *Proceedings of the 12th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'06), Vienna, Austria, March 2006*, volume 3920 of *Lecture Notes in Computer Science*, pages 455–458. Springer.

[CI-129] Patricia Bouyer, Nicolas Markey, and Pierre-Alain Reynier. Robust model-checking of linear-time properties in timed automata. *In* Jose R. Correa, Alejandro Hevia, and Marcos Kiwi, editors, *Proceedings of the 7th Latin American Symposium on Theoretical*

Informatics (LATIN'06), Valdivia, Chile, March 2006, volume 3887 of *Lecture Notes in Computer Science*, pages 238–249. Springer.

[CI-130] Jean-Pierre Jouannaud and Weiwen Xu. Automatic complexity analysis for programs extracted from Coq proof. *In* Stuart Allen, John Crossley, Kung-Kiu Lau, and Iman Poernomo, editors, *Proceedings of the Workshop on the Constructive Logic for Automated Software Engineering (CLASE'05), Edinburgh, UK, April 2005*, volume 153(1) of *Electronic Notes in Theoretical Computer Science*, pages 35–53. Elsevier Science Publishers, March 2006.

[CI-131] Patricia Bouyer, Fabrice Chevalier, and Nicolas Markey. On the expressiveness of TPTL and MTL. *In* R. Ramanujam and Sandeep Sen, editors, *Proceedings of the 25th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'05), Hyderabad, India, December 2005*, volume 3821 of *Lecture Notes in Computer Science*, pages 432–443. Springer.

[CI-132] Sophie Pinchinat and Stéphane Riedweg. On the architectures in decentralized supervisory control. *In Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'05), Seville, Spain, December 2005*, pages 12–17. IEEE Computer Society Press.

[CI-133] Mathieu Baudet. Deciding security of protocols against off-line guessing attacks. *In Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS'05), Alexandria, Virginia, USA, November 2005*, pages 16–25. ACM Press.

[CI-134] Patricia Bouyer, François Laroussinie, and Pierre-Alain Reynier. Diagonal constraints in timed automata: Forward analysis of timed systems. *In* Paul Pettersson and Wang Yi, editors, *Proceedings of the 3rd International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'05), Uppsala, Sweden, September-October 2005*, volume 3829 of *Lecture Notes in Computer Science*, pages 112–126. Springer, November 2005.

[CI-135] Silvano Dal Zilio and Régis Gascon. Resource bound certification for a tail-recursive virtual machine. *In* Kwangkeun Yi, editor, *Proceedings of the 3rd Asian Symposium on Programming Languages and Systems (APLAS'05), Tsukuba, Japan, November 2005*, volume 3780 of *Lecture Notes in Computer Science*, pages 247–263. Springer.

[CI-136] Sébastien Bardin, Alain Finkel, Jérôme Leroux, and Philippe Schnoebelen. Flat acceleration in symbolic model checking. *In* Doron A. Peled and Yih-Kuen Tsay, editors, *Proceedings of the 3rd International Symposium on Automated Technology for Verification and Analysis (ATVA'05), Taipei, Taiwan, ROC, October 2005*, volume 3707 of *Lecture Notes in Computer Science*, pages 474–488. Springer.

[CI-137] Michel Bidoit and Rolf Hennicker. Externalized and internalized notions of behavioral refinement. *In* Dang Van Hung and Martin Wirsing, editors, *Proceedings of the 2nd International Colloquium on Theoretical Aspects of Computing (ICTAC'05), Hanoi, Vietnam, October 2005*, volume 3722 of *Lecture Notes in Computer Science*, pages 334–350. Springer.

[CI-138] Stéphane Demri and David Nowak. Reasoning about transfinite sequences (extended abstract). *In* Doron A. Peled and Yih-Kuen Tsay, editors, *Proceedings of the 3rd International Symposium on Automated Technology for Verification and Analysis (ATVA'05),*

Taipei, Taiwan, ROC, October 2005, volume 3707 of *Lecture Notes in Computer Science*, pages 248–262. Springer.

[CI-139] Houda Bel mokadem, Béatrice Bérard, Vincent Gourcuff, Jean-Marc Roussel, and Olivier de Smet. Verification of a timed multitask system with Uppaal. *In* Lucia Lo Bello and Thilo Sauter, editors, *Proceedings of the 10th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'05), Catania, Italy, September 2005*, pages 347–354. IEEE Industrial Electronics Society.

[CI-140] Houda Bel mokadem, Béatrice Bérard, Patricia Bouyer, and François Laroussinie. A new modality for almost everywhere properties in timed automata. *In* Martín Abadi and Luca de Alfaro, editors, *Proceedings of the 16th International Conference on Concurrency Theory (CONCUR'05), San Francisco, CA, USA, August 2005*, volume 3653 of *Lecture Notes in Computer Science*, pages 110–124. Springer.

[CI-141] Patricia Bouyer, Franck Cassez, and François Laroussinie. Modal logics for timed control. *In* Martín Abadi and Luca de Alfaro, editors, *Proceedings of the 16th International Conference on Concurrency Theory (CONCUR'05), San Francisco, CA, USA, August 2005*, volume 3653 of *Lecture Notes in Computer Science*, pages 81–94. Springer.

[CI-142] Stéphane Demri and Régis Gascon. Verification of qualitative $\mathbb{Z}$-constraints. *In* Martín Abadi and Luca de Alfaro, editors, *Proceedings of the 16th International Conference on Concurrency Theory (CONCUR'05), San Francisco, CA, USA, August 2005*, volume 3653 of *Lecture Notes in Computer Science*, pages 518–532. Springer.

[CI-143] Paul Gastin and Dietrich Kuske. Uniform satisfiability problem for local temporal logics over Mazurkiewicz traces. *In* Martín Abadi and Luca de Alfaro, editors, *Proceedings of the 16th International Conference on Concurrency Theory (CONCUR'05), San Francisco, CA, USA, August 2005*, volume 3653 of *Lecture Notes in Computer Science*, pages 533–547. Springer.

[CI-144] Aybek Mukhamedov, Steve Kremer, and Eike Ritter. Analysis of a multi-party fair exchange protocol and formal proof of correctness in the strand space model. *In* Andrew S. Patrick and Moti Yung, editors, *Revised Papers from the 9th International Conference on Financial Cryptography and Data Security (FC'05), Roseau, The Commonwealth Of Dominica, February-March 2005*, volume 3570 of *Lecture Notes in Computer Science*, pages 255–269. Springer, August 2005.

[CI-145] Mathieu Baudet, Véronique Cortier, and Steve Kremer. Computationally sound implementations of equational theories against passive adversaries. *In* Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP'05), Lisboa, Portugal, July 2005*, volume 3580 of *Lecture Notes in Computer Science*, pages 652–663. Springer.

[CI-146] Manfred Droste and Paul Gastin. Weighted automata and weighted logics. *In* Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP'05), Lisboa, Portugal, July 2005*, volume 3580 of *Lecture Notes in Computer Science*, pages 513–525. Springer.

[CI-147] Julien Olivain and Jean Goubault-Larrecq. The Orchids intrusion detection tool. *In* Kousha Etessami and Sriram Rajamani, editors, *Proceedings of the 17th International Conference on Computer Aided Verification (CAV'05), Edinburgh, Scotland, UK, July 2005*, volume 3576 of *Lecture Notes in Computer Science*, pages 286–290. Springer.

[CI-148] Stéphane Demri, Ranko Lazić, and David Nowak. On the freeze quantifier in constraint LTL: Decidability and complexity. *In Proceedings of the 12th International Symposium on Temporal Representation and Reasoning (TIME'05), Burlington, Vermont, USA, June 2005*, pages 113–121. IEEE Computer Society Press.

[CI-149] Sophie Pinchinat and Stéphane Riedweg. You can always compute maximally permissive controllers under partial observation when they exist. *In Proceedings of the 24th American Control Conference (ACC'05), Portland, Oregon, USA, June 2005*, pages 2287–2292.

[CI-150] Christophe Darlot, Alain Finkel, and Laurent Van Begin. About Fast and TReX accelerations. *In* Michael R. A. Huth, editor, *Proceedings of the 4th International Workshop on Automated Verification of Critical Systems (AVoCS'04), London, UK, August-September 2004*, volume 128(6) of *Electronic Notes in Theoretical Computer Science*, pages 87–103. Elsevier Science Publishers, May 2005.

[CI-151] Marie Duflot, Laurent Fribourg, Thomas Hérault, Richard Lassaigne, Frédéric Magniette, Stéphane Messika, Sylvain Peyronnet, and Claudine Picaronny. Probabilistic model checking of the CSMA/CD protocol using PRISM and APMC. *In* Michael R. A. Huth, editor, *Proceedings of the 4th International Workshop on Automated Verification of Critical Systems (AVoCS'04), London, UK, August-September 2004*, volume 128(6) of *Electronic Notes in Theoretical Computer Science*, pages 195–214. Elsevier Science Publishers, May 2005.

[CI-152] Steve Kremer and Mark D. Ryan. Analysing the vulnerability of protocols to produce known-pair and chosen-text attacks. *In* Riccardo Focardi and Gianluigi Zavattaro, editors, *Proceedings of the 2nd International Workshop on Security Issues in Coordination Models, Languages and Systems (SecCo'04), London, UK, August 2004*, volume 128(5) of *Electronic Notes in Theoretical Computer Science*, pages 84–107. Elsevier Science Publishers, May 2005.

[CI-153] Patricia Bouyer, Fabrice Chevalier, and Deepak D'Souza. Fault diagnosis using timed automata. *In* Vladimiro Sassone, editor, *Proceedings of the 8th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'05), Edinburgh, U.K., April 2005*, volume 3441 of *Lecture Notes in Computer Science*, pages 219–233. Springer.

[CI-154] Hubert Comon-Lundh and Stéphanie Delaune. The finite variant property: How to get rid of some algebraic properties. *In* Jürgen Giesl, editor, *Proceedings of the 16th International Conference on Rewriting Techniques and Applications (RTA'05), Nara, Japan, April 2005*, volume 3467 of *Lecture Notes in Computer Science*, pages 294–307. Springer.

[CI-155] Alain Finkel, Gilles Geeraerts, Jean-François Raskin, and Laurent Van Begin. On the omega-language expressive power of extended Petri nets. *In* Jos Baeten and Flavio Corradini, editors, *Proceedings of the 11th International Workshop on Expressiveness in Concurrency (EXPRESS'04), London, UK, August 2004*, volume 128(2) of *Electronic*

Notes in Theoretical Computer Science, pages 87–101. Elsevier Science Publishers, April 2005.

[CI-156] Steve Kremer and Mark D. Ryan. Analysis of an electronic voting protocol in the applied pi-calculus. *In* Mooly Sagiv, editor, *Programming Languages and Systems — Proceedings of the 14th European Symposium on Programming (ESOP'05), Edinburgh, U.K., April 2005*, volume 3444 of *Lecture Notes in Computer Science*, pages 186–200. Springer.

[CI-157] Pascal Lafourcade, Denis Lugiez, and Ralf Treinen. Intruder deduction for AC-like equational theories with homomorphisms. *In* Jürgen Giesl, editor, *Proceedings of the 16th International Conference on Rewriting Techniques and Applications (RTA'05), Nara, Japan, April 2005*, volume 3467 of *Lecture Notes in Computer Science*, pages 308–322. Springer.

[CI-158] François Laroussinie and Jeremy Sproston. Model checking durational probabilistic systems. *In* Vladimiro Sassone, editor, *Proceedings of the 8th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'05), Edinburgh, U.K., April 2005*, volume 3441 of *Lecture Notes in Computer Science*, pages 140–154. Springer.

[CI-159] Patricia Bouyer, Franck Cassez, Emmanuel Fleury, and Kim G. Larsen. Synthesis of optimal strategies using HyTech. *In* Luca De Alfaro, editor, *Proceedings of the Workshop on Games in Design and Verification (GDV'04), Boston, Massachusetts, USA, July 2004*, volume 119(1) of *Electronic Notes in Theoretical Computer Science*, pages 11–31. Elsevier Science Publishers, February 2005.

[CI-160] Jean Goubault-Larrecq and Fabrice Parrennes. Cryptographic protocol analysis on real C code. *In* Radhia Cousot, editor, *Proceedings of the 6th International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI'05), Paris, France, January 2005*, volume 3385 of *Lecture Notes in Computer Science*, pages 363–379. Springer.

[CI-161] Michel Bidoit, Donald Sannella, and Andrzej Tarlecki. Toward component-oriented formal software development: An algebraic approach. *In* Martin Wirsing, Alexander Knapp, and Simonetta Balsamo, editors, *Revised Papers of the 9th International Workshop on Radical Innovations of Software and Systems Engineering in the Future (RISSEF'02), Venice, Italy, October 2002*, volume 2941 of *Lecture Notes in Computer Science*, pages 75–90. Springer, 2004.

[CI-162] Patricia Bouyer, Franck Cassez, Emmanuel Fleury, and Kim G. Larsen. Optimal strategies in priced timed game automata. *In* Kamal Lodaya and Meena Mahajan, editors, *Proceedings of the 24th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'04), Chennai, India, December 2004*, volume 3328 of *Lecture Notes in Computer Science*, pages 148–160. Springer.

[CI-163] Paul Gastin, Benjamin Lerman, and Marc Zeitoun. Distributed games with causal memory are decidable for series-parallel systems. *In* Kamal Lodaya and Meena Mahajan, editors, *Proceedings of the 24th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'04), Chennai, India, December 2004*, volume 3328 of *Lecture Notes in Computer Science*, pages 275–286. Springer.

[CI-164] Sébastien Bardin and Alain Finkel. Composition of accelerations to verify infinite heterogeneous systems. *In* Farn Wang, editor, *Proceedings of the 2nd International Symposium on Automated Technology for Verification and Analysis (ATVA'04), Taipei, Taiwan, ROC, October-November 2004*, volume 3299 of *Lecture Notes in Computer Science*, pages 248–262. Springer.

[CI-165] Stéphanie Delaune and Florent Jacquemard. A decision procedure for the verification of security protocols with explicit destructors. *In* Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick McDaniel, editors, *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS'04), Washington, D.C., USA, October 2004*, pages 278–287. ACM Press.

[CI-166] Laurent Fribourg, Stéphane Messika, and Claudine Picaronny. Coupling and self-stabilization. *In* Rachid Guerraoui, editor, *Proceedings of the 18th International Symposium on Distributed Computing (DISC'04), Amsterdam, The Netherlands, October 2004*, volume 3274 of *Lecture Notes in Computer Science*, pages 201–215. Springer.

[CI-167] Jérôme Leroux. Disjunctive invariants for numerical systems. *In* Farn Wang, editor, *Proceedings of the 2nd International Symposium on Automated Technology for Verification and Analysis (ATVA'04), Taipei, Taiwan, ROC, October-November 2004*, volume 3299 of *Lecture Notes in Computer Science*, pages 93–107. Springer.

[CI-168] Michel Bidoit, Rolf Hennicker, Alexander Knapp, and Hubert Baumeister. Glass-box and black-box views on object-oriented specifications. *In Proceedings of the 2nd IEEE International Conference on Software Engineering and Formal Methods (SEFM'04), Beijing, China, September 2004*, pages 208–217. IEEE Computer Society Press.

[CI-169] Martin De Wulf, Laurent Doyen, Nicolas Markey, and Jean-François Raskin. Robustness and implementability of timed automata. *In* Yassine Lakhnech and Sergio Yovine, editors, *Proceedings of the Joint Conferences Formal Modelling and Analysis of Timed Systems (FORMATS'04) and Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT'04), Grenoble, France, September 2004*, volume 3253 of *Lecture Notes in Computer Science*, pages 118–133. Springer.

[CI-170] Jean Goubault-Larrecq, Sławomir Lasota, David Nowak, and Yu Zhang. Complete lax logical relations for cryptographic lambda-calculi. *In* Jerzy Marcinkowski and Andrzej Tarlecki, editors, *Proceedings the 18th International Workshop on Computer Science Logic (CSL'04), Karpacz, Poland, September 2004*, volume 3210 of *Lecture Notes in Computer Science*, pages 400–414. Springer.

[CI-171] Nicolas Markey and Philippe Schnoebelen. Symbolic model checking for simply-timed systems. *In* Yassine Lakhnech and Sergio Yovine, editors, *Proceedings of the Joint Conferences Formal Modelling and Analysis of Timed Systems (FORMATS'04) and Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT'04), Grenoble, France, September 2004*, volume 3253 of *Lecture Notes in Computer Science*, pages 102–117. Springer.

[CI-172] Antonín Kučera and Philippe Schnoebelen. A general approach to comparing infinite-state systems with their finite-state specifications. *In* Philippa Gardner and Nobuko Yoshida, editors, *Proceedings of the 15th International Conference on Concurrency Theory (CONCUR'04), London, UK, August 2004*, volume 3170 of *Lecture Notes in Computer Science*, pages 372–386. Springer.

[CI-173] François Laroussinie, Nicolas Markey, and Philippe Schnoebelen. Model checking timed automata with one or two clocks. *In* Philippa Gardner and Nobuko Yoshida, editors, *Proceedings of the 15th International Conference on Concurrency Theory (CONCUR'04), London, UK, August 2004*, volume 3170 of *Lecture Notes in Computer Science*, pages 387–401. Springer.

[CI-174] Jérôme Leroux. The affine hull of a binary automaton is computable in polynomial time. *In* Philippe Schnoebelen, editor, *Proceedings of the 5th International Workshop on Verification of Infinite State Systems (INFINITY'03), Marseilles, France, September 2003*, volume 98 of *Electronic Notes in Theoretical Computer Science*, pages 89–104. Elsevier Science Publishers, August 2004.

[CI-175] Jérôme Leroux and Grégoire Sutre. On flatness for 2-dimensional vector addition systems with states. *In* Philippa Gardner and Nobuko Yoshida, editors, *Proceedings of the 15th International Conference on Concurrency Theory (CONCUR'04), London, UK, August 2004*, volume 3170 of *Lecture Notes in Computer Science*, pages 402–416. Springer.

[CI-176] Nicolas Markey and Jean-François Raskin. Model checking restricted sets of timed paths. *In* Philippa Gardner and Nobuko Yoshida, editors, *Proceedings of the 15th International Conference on Concurrency Theory (CONCUR'04), London, UK, August 2004*, volume 3170 of *Lecture Notes in Computer Science*, pages 432–447. Springer.

[CI-177] Alain Finkel and Jérôme Leroux. Image computation in infinite state model checking. *In* Rajeev Alur and Doron A. Peled, editors, *Proceedings of the 16th International Conference on Computer Aided Verification (CAV'04), Boston, Massachusetts, USA, July 2004*, volume 3114 of *Lecture Notes in Computer Science*, pages 361–371. Springer.

[CI-178] Stéphanie Delaune and Florent Jacquemard. A theory of dictionary attacks and its complexity. *In Proceedings of the 17th IEEE Computer Security Foundations Workshop (CSFW'04), Asilomar, Pacific Grove, California, USA, June 2004*, pages 2–15. IEEE Computer Society Press.

[CI-179] Alain Finkel and Jérôme Leroux. Polynomial time image computation with interval-definable counters systems. *In* Susanne Graf and Laurent Mounier, editors, *Proceedings of the 11th International SPIN Workshop on Model Checking Software (SPIN'04), Barcelona, Spain, April 2004*, volume 2989 of *Lecture Notes in Computer Science*, pages 182–197. Springer.

[CI-180] Sébastien Bardin, Alain Finkel, and Jérôme Leroux. FASTer acceleration of counter automata in practice. *In* Kurt Jensen and Andreas Podelski, editors, *Proceedings of the 10th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'04), Barcelona, Spain, March 2004*, volume 2988 of *Lecture Notes in Computer Science*, pages 576–590. Springer.

[CI-181] Gerd Behrmann, Patricia Bouyer, Kim G. Larsen, and Radek Pelánek. Lower and upper bounds in zone based abstractions of timed automata. *In* Kurt Jensen and Andreas Podelski, editors, *Proceedings of the 10th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'04), Barcelona, Spain, March 2004*, volume 2988 of *Lecture Notes in Computer Science*, pages 312–326. Springer.

[CI-182] Patricia Bouyer, Ed Brinksma, and Kim G. Larsen. Staying alive as cheaply as possible. *In* Rajeev Alur and George J. Pappas, editors, *Proceedings of the 7th International Conference on Hybrid Systems: Computation and Control (HSCC'04), Philadelphia, Pennsylvania, USA, March 2004*, volume 2993 of *Lecture Notes in Computer Science*, pages 203–218. Springer.

[CI-183] Jennifer M. Davoren, Vaughan Coulthard, Nicolas Markey, and Thomas Moor. Non-deterministic temporal logics for general flow systems. *In* Rajeev Alur and George J. Pappas, editors, *Proceedings of the 7th International Conference on Hybrid Systems: Computation and Control (HSCC'04), Philadelphia, Pennsylvania, USA, March 2004*, volume 2993 of *Lecture Notes in Computer Science*, pages 280–295. Springer.

[CI-184] Stéphane Demri. LTL over integer periodicity constraints. *In* Igor Walukiewicz, editor, *Proceedings of the 7th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'04), Barcelona, Spain, March 2004*, volume 2987 of *Lecture Notes in Computer Science*, pages 121–135. Springer.

[CI-185⋆] Yohan Boichut, Pierre-Cyrille Héam, and Olga Kouchnarenko. Tree automata for detecting attacks on protocols with algebraic cryptographic primitives. *In* P. Madhusudan and Vineet Kahlon, editors, *Proceedings of the 9th International Workshop on Verification of Infinite State Systems (INFINITY'07), Lisbon, Portugal, September 2007*, Electronic Notes in Theoretical Computer Science. Elsevier Science Publishers, 2008. To appear.

[CI-186⋆] Graham Steel. The importance of non-theorems and counterexamples in program verification. *In* Bertrand Meyer and Jim Woodcock, editors, *Revised Selected Papers and Discussions of the 1st IFIP TC2/WG2.3 Conference Verified Software—Theories, Tools, and Experiments (VSTTE'05), Zurich, Switzerland, October 2005*, volume 4171 of *Lecture Notes in Computer Science*, pages 491–495. Springer, 2008.

[CI-187⋆] Mohammad Mahdi Jaghoori, Delphine Longuet, and Frank S. de Boer. Schedulability and compatibility of real-time asynchronous objects. *In Proceedings of the 29th IEEE Real-Time Systems Symposium (RTSS'08), Barcelona, Spain, November 2008*. IEEE Computer Society Press. To appear.

[CI-188⋆] Romé Courbis, Pierre-Cyrille Héam, and Olga Kouchnarenko. Handling left-quadratic rules when completing tree automata. *In* Vesa Halava and Igor Potapov, editors, *Proceedings of the 2nd Workshop on Reachability Problems (RP'08), Liverpool, UK, September 2008*, Electronic Notes in Theoretical Computer Science. Elsevier Science Publishers. To appear.

[CI-189⋆] Pierre-Cyrille Héam, Olga Kouchnarenko, and Jérôme Voinot. Component simulation-based substitutivity managing QoS aspects. *In* Carlos Canal and Corina Pasareanu, editors, *Proceedings of the 5th International Workshop on Formal Aspects of Component Software (FACS'08), Malaga, Spain, September 2008*, Electronic Notes in Theoretical Computer Science. Elsevier Science Publishers. To appear.

[CI-190⋆] Adam Antonik, Michael Huth, Kim G. Larsen, Ulrik Nyman, and Andrzej Wasowski. EXPTIME-complete decision problems for mixed and modal specifications. *In* Daniele Gorla and Thomas Hildebrandt, editors, *Proceedings of the 15th International Workshop on Expressiveness in Concurrency (EXPRESS'08), Toronto, Canada, August 2008*, Electronic Notes in Theoretical Computer Science. Elsevier Science Publishers. To appear.

[CI-191⋆] Dietmar Berwanger, Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger, and Sangram Raje. Strategy construction for parity games with imperfect information. *In* Franck van Breugel and Marsha Chechik, editors, *Proceedings of the 19th International Conference on Concurrency Theory (CONCUR'08), Toronto, Canada, August 2008*, volume 5201 of *Lecture Notes in Computer Science*, pages 325–339. Springer.

[CI-192⋆] Michel Bidoit and Rolf Hennicker. An algebraic semantics for contract-based software components. *In* José Meseguer and Grigore Rosu, editors, *Proceedings of the 12th International Conference on Algebraic Methodology and Software Technology (AMAST'08), Urbana, Illinois, USA, July 2008*, volume 5140 of *Lecture Notes in Computer Science*, pages 216–231. Springer.

[CI-193⋆] Yohan Boichut, Roméo Courbis, Pierre-Cyrille Héam, and Olga Kouchnarenko. Finer is better: Abstraction refinement for rewriting approximations. *In* Andrei Voronkov, editor, *Proceedings of the 19th International Conference on Rewriting Techniques and Applications (RTA'08), Hagenberg, Austria, July 2008*, volume 5117 of *Lecture Notes in Computer Science*, pages 48–62. Springer.

[CI-194⋆] Leonid Libkin and Cristina Sirangelo. Data exchange and schema mappings in open and closed worlds. *In* Maurizio Lenzerini and Domenico Lembo, editors, *Proceedings of the 26th Annual ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'08), Vancouver, Canada, June 2008*, pages 139–148. ACM Press.

[CI-195⋆] Sandra Alves, Mário Florido, Ian Mackie, and François-Régis Sinot. Minimality in a linear calculus with iteration. *In* Jürgen Giesl, editor, *Proceedings of the 7th International Workshop on Reduction Strategies in Rewriting and Programming (WRS'07), Paris, France, June 2007*, volume 204 of *Electronic Notes in Theoretical Computer Science*, pages 163–179. Elsevier Science Publishers, April 2008.

[CI-196⋆] Sylvain Schmitz. An experimental ambiguity detection tool. *In* Anthony Sloane and Adrian Johnstone, editors, *Proceedings of the 7th Workshop on Language Descriptions, Tools, and Applications (LDTA'07), Braga, Portugal, July 2007*, volume 203(2) of *Electronic Notes in Theoretical Computer Science*, pages 69–84. Elsevier Science Publishers, April 2008.

[CI-197⋆] François-Régis Sinot. Complete laziness: a natural semantics. *In* Jürgen Giesl, editor, *Proceedings of the 7th International Workshop on Reduction Strategies in Rewriting and Programming (WRS'07), Paris, France, June 2007*, volume 204 of *Electronic Notes in Theoretical Computer Science*, pages 129–145. Elsevier Science Publishers, April 2008.

[CI-198⋆] Adam Antonik, Michael Huth, Kim G. Larsen, Ulrik Nyman, and Andrzej Wasowski. Complexity of decision problems for mixed and modal transition systems. *In* Roberto Amadio, editor, *Proceedings of the 11th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'08), Budapest, Hungary, March-April 2008*, volume 4962 of *Lecture Notes in Computer Science*, pages 112–126. Springer.

[CI-199⋆] François-Régis Sinot. Sub-λ-calculi, classified. *In* Ian Mackie and Detlef Plump, editors, *Proceedings of the 4th International Workshop on Term Graph Rewriting (TERMGRAPH'07), Braga, Portugal, March 2007*, volume 203(1) of *Electronic Notes in Theoretical Computer Science*, pages 123–133. Elsevier Science Publishers, March 2008.

[CI-200⋆] Myrto Arapinis and Marie Duflot. Bounding messages for free in security protocols. *In* V. Arvind and Sanjiva Prasad, editors, *Proceedings of the 27th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'07), New Delhi, India, December 2007*, volume 4855 of *Lecture Notes in Computer Science*, pages 376–387. Springer.

[CI-201⋆] Marion Daubignard, Romain Janvier, Yassine Lakhnech, and Laurent Mazaré. Game-based criterion partition applied to computational soundness of adaptive security. *In* Theo Dimitrakos, Fabio Martinelli, Peter Y. A. Ryan, and Steve Schneider, editors, *Revised Selected Papers of the 4th International Workshop on Formal Aspects in Security and Trust (FAST'06), Hamilton, Ontario, Canada, August 2006*, volume 4691 of *Lecture Notes in Computer Science*, pages 47–64. Springer, October 2007.

[CI-202⋆] Laura Recalde, Serge Haddad, and Manuel Silva. Continuous Petri nets: Expressive power and decidability issues. *In* Kedar Namjoshi and Tomohiro Yoneda, editors, *Proceedings of the 5th International Symposium on Automated Technology for Verification and Analysis (ATVA'07), Tokyo, Japan, October 2007*, volume 4762 of *Lecture Notes in Computer Science*, pages 362–377. Springer.

[CI-203⋆] Adam Antonik and Michael Huth. On the complexity of semantic self-minimization. *In* Michael Goldsmith and Bill Roscoe, editors, *Proceedings of the 7th International Workshop on Automated Verification of Critical Systems (AVoCS'07), Oxford, UK, September 2007*, Electronic Notes in Theoretical Computer Science. Elsevier Science Publishers. To appear.

[CI-204⋆] Susanna Donatelli, Serge Haddad, and Jeremy Sproston. $\mathrm{CSL}^{\mathrm{TA}}$: an expressive logic for continuous-time Markov chains. *In Proceedings of the 4th International Conference on Quantitative Evaluation of Systems (QEST'07), Edinburgh, Scotland, September 2007*, pages 31–40. IEEE Computer Society Press.

[CI-205⋆] Delphine Longuet and Marc Aiguier. Specification-based testing for CoCasl's modal specifications. *In* Till Mossakowski, Ugo Montanari, and Magne Haveraaen, editors, *Proceedings of the 2nd International Conference on Algebra and Coalgebra in Computer Science (CALCO'07), Bergen, Norway, August 2007*, volume 4624 of *Lecture Notes in Computer Science*, pages 356–371. Springer-Verlag.

[CI-206⋆] Mathias Samuelides and Luc Segoufin. Complexity of pebble tree-walking automata. *In* Erzsébet Csuhaj-Varjú and Zoltán Ésik, editors, *Proceedings of the 16th International Symposium on Fundamentals of Computation Theory (FCT'07), Budapest, Hungary, August 2007*, volume 4639 of *Lecture Notes in Computer Science*, pages 458–469. Springer.

[CI-207⋆] Horatiu Cirstea, Germain Faure, Maribel Fernández, Ian Mackie, and François-Régis Sinot. From functional programs to interaction nets via the rewriting calculus. *In* Sergio Antoy, editor, *Proceedings of the 6th International Workshop on Reduction Strategies in Rewriting and Programming (WRS'06), Seattle, Washington, USA, August 2006*, volume 174(10) of *Electronic Notes in Theoretical Computer Science*, pages 39–56. Elsevier Science Publishers, July 2007.

[CI-208⋆] Sylvain Schmitz. Conservative ambiguity detection in context-free grammars. *In* Lars Arge, Christian Cachin, Tomasz Jurdziński, and Andrzej Tarlecki, editors, *Proceedings of the 34th International Colloquium on Automata, Languages and Programming (ICALP'07), Wrocław, Poland, July 2007*, volume 4596 of *Lecture Notes in Computer Science*, pages 692–703. Springer.

[CI-209⋆] Marc Aiguier and Delphine Longuet. Test selection criteria for modal specifications of reactive systems. *In Proceedings of the 1st Joint IEEE/IFIP Symposium on Theoretical Aspects of Software Engineering (TASE'07), Shanghai, China, June 2007*, pages 159–170. IEEE Computer Society Press.

[CI-210⋆] Marco Beccuti, Giuliana Franceschinis, and Serge Haddad. Markov decision Petri net and Markov decision well-formed net formalisms. *In* Jetty Kleijn and Alex Yakovlev, editors, *Proceedings of the 28th International Conference on Applications and Theory of Petri Nets (ICATPN'07), Siedlce, Poland, June 2007*, volume 4546 of *Lecture Notes in Computer Science*, pages 43–62. Springer.

[CI-211⋆] Serge Haddad and Pascal Poizat. Transactional reduction of component compositions. *In* John Derrick and Jüri Vain, editors, *Proceedings of 27th IFIP WG6.1 International Conference on Formal Techniques for Networked and Distributed Systems (FORTE'07), Tallinn, Estonia, June 2007*, volume 4574 of *Lecture Notes in Computer Science*, pages 341–357. Springer.

[CI-212⋆] Marc Aiguier, Agnès Arnould, Pascale Le Gall, and Delphine Longuet. Test selection criteria for quantifier-free first-order specifications. *In* Farhad Arbab and Marjan Sirjani, editors, *Proceedings of the International Symposium on Fundamentals of Software Engineering (FSEN'07), Tehran, Iran, April 2007*, volume 4767 of *Lecture Notes in Computer Science*, pages 144–159. Springer.

[CI-213⋆] Florent Kirchner and François-Régis Sinot. Rule-based operational semantics for an imperative language. *In* Mariel Fernández and Ralf Lämmel, editors, *Proceedings of the 7th International Workshop on Rule Based Programming (RULE'06), Seattle, Washington, USA, August 2006*, volume 174(1) of *Electronic Notes in Theoretical Computer Science*, pages 35–47. Elsevier Science Publishers, April 2007.

[CI-214⋆] Véronique Cortier, Gavin Keighren, and Graham Steel. Automatic analysis of the security of XOR-based key management schemes. *In* Orna Grumberg and Michael Huth, editors, *Proceedings of the 13th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'07), Braga, Portugal, March 2007*, volume 4424 of *Lecture Notes in Computer Science*, pages 538–552. Springer.

[CI-215⋆] Dietmar Berwanger. Admissibility in infinite games. *In* Wolfgang Thomas and Pascal Weil, editors, *Proceedings of the 24th Annual Symposium on Theoretical Aspects of Computer Science (STACS'07), Aachen, Germany, February 2007*, volume 4393 of *Lecture Notes in Computer Science*, pages 188–199. Springer.

[CI-216⋆] Alan Nash, Luc Segoufin, and Victor Vianu. Determinacy and rewriting of conjunctive queries using views: A progress report. *In* Thomas Schwentick and Dan Suciu, editors, *Proceedings of the 11th International Conference on Database Theory (ICDT'07), Barcelona, Spain, January 2007*, volume 4353 of *Lecture Notes in Computer Science*, pages 59–73. Springer-Verlag.

[CI-217⋆] Luc Segoufin and Cristina Sirangelo. Constant-memory validation of streaming XML documents against DTDs. *In* Thomas Schwentick and Dan Suciu, editors, *Proceedings of the 11th International Conference on Database Theory (ICDT'07), Barcelona, Spain, January 2007*, volume 4353 of *Lecture Notes in Computer Science*, pages 299–313. Springer-Verlag.

[CI-218⋆] Jeremy Bryans, Maciej Koutny, Laurent Mazaré, and Peter Y. A. Ryan. Opacity generalised to transition systems. *In* Theo Dimitrakos, Fabio Martinelli, Peter Y. A. Ryan, and Steve A. Schneider, editors, *Revised Selected Papers of the 3rd International Workshop on Formal Aspects in Security and Trust (FAST'05), Newcastle upon Tyne, UK, July 2005*, volume 3866 of *Lecture Notes in Computer Science*, pages 81–95. Springer, 2006.

[CI-219⋆] Yohan Boichut, Pierre-Cyrille Héam, and Olga Kouchnarenko. Handling algebraic properties in automatic analysis of security protocols. *In* Kamel Barkaoui, Ana Cavalcanti, and Antonio Cerone, editors, *Proceedings of the 3rd International Colloquium on Theoretical Aspects of Computing (ICTAC'06), Tunis, Tunisia, November 2006*, volume 4281 of *Lecture Notes in Computer Science*, pages 153–167. Springer.

[CI-220⋆] Franck Cassez, Thomas Chatain, and Claude Jard. Symbolic unfoldings for networks of timed automata. *In* Susanne Graf and Wenhui Zhang, editors, *Proceedings of the 4th International Symposium on Automated Technology for Verification and Analysis (ATVA'06), Beijing, ROC, October 2006*, volume 4218 of *Lecture Notes in Computer Science*, pages 307–321. Springer.

[CI-221⋆] Dietmar Berwanger and David Janin. Automata on directed graphs: Vertex versus edge marking. *In* Andrea Corradini, Hartmut Ehrig, Ugo Montanari, Leila Ribeiro, and Grzegorz Rozenberg, editors, *Proceedings of the 3rd International Conference on Graph Transformations (ICGT'06), Natal, Rio Grande do Norte, Brazil, September 2006*, volume 4178 of *Lecture Notes in Computer Science*, pages 46–60. Springer.

[CI-222⋆] Serge Haddad, Laura Recalde, and Manuel Silva. On the computational power of timed differentiable Petri nets. *In* Eugène Asarin and Patricia Bouyer, editors, *Proceedings of the 4th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'06), Paris, France, September 2006*, volume 4202 of *Lecture Notes in Computer Science*, pages 230–244. Springer.

[CI-223⋆] Luc Segoufin. Automata and logics for words and trees over an infinite alphabet. *In* Zoltán Ésik, editor, *Proceedings of the 15th Annual EACSL Conference on Computer Science Logic (CSL'06), Szeged, Hungary, September 2006*, volume 4207 of *Lecture Notes in Computer Science*, pages 41–57. Springer.

[CI-224⋆] Mikołaj Bojańczyk, Anca Muscholl, Thomas Schwentick, Luc Segoufin, and Claire David. Two-variable logic on words with data. *In Proceedings of the 21st Annual IEEE Symposium on Logic in Computer Science (LICS'06), Seattle, Washington, USA, August 2006*, pages 7–16. IEEE Computer Society Press.

[CI-225⋆] Ahmed Bouajjani, Marius Bozga, Peter Habermehl, Radu Iosif, Pierre Moro, and Tomáš Vojnar. Programs with lists are counter automata. *In* Thomas Ball and Robert B. Jones, editors, *Proceedings of the 18th International Conference on Computer Aided Verification (CAV'06), Seattle, Washington, USA, August 2006*, volume 4144 of *Lecture Notes in Computer Science*, pages 517–531. Springer.

[CI-226⋆] Ahmed Bouajjani, Peter Habermehl, Adam Rogalewicz, and Tomáš Vojnar. Abstract regular tree model checking of complex dynamic data structures. *In* Kwangkeun Yi, editor, *Proceedings of the 13th International Symposium Static Analysis (SAS'06), Seoul, Korea, August 2006*, volume 4134 of *Lecture Notes in Computer Science*, pages 52–70. Springer.

[CI-227⋆] Luciano Caroprese, Sergio Greco, Cristina Sirangelo, and Ester Zumpano. Declarative semantics of production rules for integrity maintenance. *In* Sandro Etalle and Mirosław Truszczyński, editors, *Proceedings of the 22nd International Conference on Logic Programming (ICLP'06), Seattle, WA, USA, August 2006*, volume 4079 of *Lecture Notes in Computer Science*, pages 26–40. Springer.

[CI-228⋆] José Fortes Gálvez, Sylvain Schmitz, and Jacques Farré. Shift-resolve parsing: Simple, linear time, unbounded lookahead. *In* Oscar H. Ibarra and Hsu-Chun Yen, editors, *Proceedings of the 11th International Conference on Implementation and Application of Automata (CIAA'06), Taipei, Taiwan, ROC, August 2006*, volume 4094 of *Lecture Notes in Computer Science*, pages 253–264. Springer-Verlag.

[CI-229⋆] Maribel Fernández, Ian Mackie, and François-Régis Sinot. Interaction nets vs. the ρ-calculus: Introducing bigraphical nets. *In* Jos Baeten and Iain Phillips, editors, *Proceedings of the 12th International Workshop on Expressiveness in Concurrency (EXPRESS'05), San Francisco, CA, USA, August 2005*, volume 154(3) of *Electronic Notes in Theoretical Computer Science*, pages 19–32. Elsevier Science Publishers, July 2006.

[CI-230⋆] Mikołaj Bojańczyk, Claire David, Anca Muscholl, Thomas Schwentick, and Luc Segoufin. Two-variable logic on data trees and XML reasoning. *In* Stijn Vansummeren, editor, *Proceedings of the 25th Annual ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'06), Chicago, Illinois, USA, June 2006*, pages 10–19. ACM Press.

[CI-231⋆] Thomas Chatain and Claude Jard. Complete finite prefixes of symbolic unfoldings of safe time Petri nets. *In* Susanna Donatelli and P. S. Thiagarajan, editors, *Proceedings of the 27th International Conference on Applications and Theory of Petri Nets (ICATPN'06), Turku, Finland, June 2006*, volume 4024 of *Lecture Notes in Computer Science*, pages 125–145. Springer.

[CI-232⋆] Serge Haddad, Lynda Mokdad, and Patrice Moreaux. A new approach to the evaluation of non Markovian stochastic Petri nets. *In* Susanna Donatelli and P. S. Thiagarajan, editors, *Proceedings of the 27th International Conference on Applications and Theory of Petri Nets (ICATPN'06), Turku, Finland, June 2006*, volume 4024 of *Lecture Notes in Computer Science*, pages 221–240. Springer.

[CI-233⋆] Sylvain Schmitz. Noncanonical LALR(1) parsing. *In* Zhe Dang and Oscar H. Ibarra, editors, *Proceedings of the 10th International Conference on Developments in Language Theory (DLT'06), Santa Barbara, CA, USA, June 2006*, volume 4036 of *Lecture Notes in Computer Science*, pages 95–107. Springer.

[CI-234⋆] Marc Aiguier, Diane Bahrami, and Delphine Longuet. An abstract way to define rewriting logic. *In* Farhad Arbab and Marjan Sirjani, editors, *Proceedings of the 1st IPM International Workshop on Foundations of Software Engineering (FSEN'05), Tehran, Iran, October 2005*, volume 159 of *Electronic Notes in Theoretical Computer Science*, pages 205–226. Elsevier Science Publishers, May 2006.

[CI-235⋆] Adam Antonik and Michael Huth. Efficient patterns for model checking partial state spaces in CTL∩LTL. *In* Steve Brookes and Michael Mislove, editors, *Proceedings of the 22nd Conference on Mathematical Foundations of Programming Semantics (MFPS'06), Genova, Italy, May 2006*, volume 158 of *Electronic Notes in Theoretical Computer Science*, pages 41–57. Elsevier Science Publishers.

[CI-236⋆] Emmanuel Beffara. A concurrent model for linear logic. *In* Martín Escardó, Achim Jung, and Michael Mislove, editors, *Proceedings of the 21st Conference on Mathematical Foundations of Programming Semantics (MFPS'05), Birmingham, U.K., May 2005*, volume 155 of *Electronic Notes in Theoretical Computer Science*, pages 147–168. Elsevier Science Publishers, May 2006.

[CI-237⋆] Serge Haddad, Patrice Moreaux, and Sylvain Rampacek. Client synthesis for web services by way of a timed semantics. *In* Yannis Manolopoulos, Joaquim Filipe, Panos Constantopoulos, and José Cordeiro, editors, *Proceedings of the 8th International Conference on Enterprise Information Systems (ICEIS'06), volume 4, Paphos, Cyprus, May 2006*, pages 19–26.

[CI-238⋆] Filippo Furfaro, Giuseppe M. Mazzeo, and Cristina Sirangelo. Exploiting cluster analysis for constructing multi-dimensional histograms on both static and evolving data. *In* Yannis E. Ioannidis, Marc H. Scholl, Florian Schmidt, Joachim W. Matthes, Michael Hatzopoulos, Klemens Böhm, Alfons Kemper, Torsten Grust, and Christian Böhm, editors, *Advances in Database Technology — Proceedings of the 10th International Conference on Extending Database Technology (EDBT'06), Munich, Germany, March 2006*, volume 3896 of *Lecture Notes in Computer Science*, pages 442–459. Springer.

[CI-239⋆] Peter Habermehl, Radu Iosif, and Tomáš Vojnar. Automata-based verification of programs with tree updates. *In* Holger Hermanns and Jens Palsberg, editors, *Proceedings of the 12th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'06), Vienna, Austria, March 2006*, volume 3920 of *Lecture Notes in Computer Science*, pages 350–364. Springer.

[CI-240⋆] François-Régis Sinot. Token-passing nets: Call-by-need for free. *In* Maribel Fernández and Ian Mackie, editors, *Proceedings of the 1st International Workshop on Developments in Computational Models (DCM'05), Lisbon, Portugal, July 2005*, pages 129–139. Elsevier Science Publishers, March 2006.

[CI-241⋆] Dietmar Berwanger, Anuj Dawar, Paul Hunter, and Stephan Kreutzer. DAG-width and parity games. *In* Bruno Durand and Wolfgang Thomas, editors, *Proceedings of the 23rd Annual Symposium on Theoretical Aspects of Computer Science (STACS'06), Marseilles, France, February 2006*, volume 3884 of *Lecture Notes in Computer Science*, pages 524–536. Springer.

[CI-242⋆] Ahmed Bouajjani, Peter Habermehl, Adam Rogalewicz, and Tomáš Vojnar. Abstract regular tree model checking. *In* Jiří Srba and Scott A. Smolka, editors, *Proceedings of the 7th International Workshop on Verification of Infinite State Systems (INFINITY'05), San Francisco, CA, USA, August 2005*, volume 149(1) of *Electronic Notes in Theoretical Computer Science*, pages 37–48. Elsevier Science Publishers, February 2006.

[CI-243⋆] Laurent Mazaré. Decidability of opacity with non-atomic keys. *In* Theo Dimitrakos and Fabio Martinelli, editors, *Proceedings of the 2nd IFIP TC1 WG1.7 Workshop on Formal Aspects in Security and Trust (FAST'04), Toulouse, France, August 2004*, volume 173 of *International Federation for Information Processing*, pages 71–84. Springer, 2005.

[CI-244⋆] Marc Aiguier, Christophe Gaston, Pascale Le Gall, Delphine Longuet, and Assia Touil. A temporal logic for input output symbolic transition systems. *In Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC'05), Taipei, Taiwan, ROC, December 2005*, pages 43–50. IEEE Computer Society Press.

[CI-245⋆] Béatrice Bérard, Franck Cassez, Serge Haddad, Didier Lime, and Olivier H. Roux. When are timed automata weakly timed bisimilar to time Petri nets? *In* R. Ramanujam and Sandeep Sen, editors, *Proceedings of the 25th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'05), Hyderabad, India, December 2005*, volume 3821 of *Lecture Notes in Computer Science*, pages 273–284. Springer.

[CI-246⋆] Peter Habermehl and Tomáš Vojnar. Regular model checking using inference of regular languages. *In* Julian Bradfield and Faron Moller, editors, *Proceedings of the 6th International Workshop on Verification of Infinite State Systems (INFINITY'04), London, UK, September 2004*, volume 138(3) of *Electronic Notes in Theoretical Computer Science*, pages 21–36. Elsevier Science Publishers, December 2005.

[CI-247⋆] Béatrice Bérard, Franck Cassez, Serge Haddad, Didier Lime, and Olivier H. Roux. Comparison of the expressiveness of timed automata and time Petri nets. *In* Paul Pettersson and Wang Yi, editors, *Proceedings of the 3rd International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'05), Uppsala, Sweden, September-October 2005*, volume 3829 of *Lecture Notes in Computer Science*, pages 211–225. Springer, November 2005.

[CI-248⋆] Thomas Brihaye, Véronique Bruyère, and Jean-François Raskin. On optimal timed strategies. *In* Paul Pettersson and Wang Yi, editors, *Proceedings of the 3rd International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'05), Uppsala, Sweden, September-October 2005*, volume 3829 of *Lecture Notes in Computer Science*, pages 49–64. Springer, November 2005.

[CI-249⋆] Thomas Chatain and Claude Jard. Time supervision of concurrent systems using symbolic unfoldings of time Petri nets. *In* Paul Pettersson and Wang Yi, editors, *Proceedings of the 3rd International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'05), Uppsala, Sweden, September-October 2005*, volume 3829 of *Lecture Notes in Computer Science*, pages 196–210. Springer, November 2005.

[CI-250⋆] Béatrice Bérard, Franck Cassez, Serge Haddad, Didier Lime, and Olivier H. Roux. Comparison of different semantics for time Petri nets. *In* Doron A. Peled and Yih-Kuen Tsay, editors, *Proceedings of the 3rd International Symposium on Automated Technology for Verification and Analysis (ATVA'05), Taipei, Taiwan, ROC, October 2005*, volume 3707 of *Lecture Notes in Computer Science*, pages 293–307. Springer.

[CI-251⋆] Benedikt Bollig and Martin Leucker. A hierarchy of implementable MSC languages. *In* Farn Wang, editor, *Proceedings of 25th IFIP WG6.1 International Conference on Formal Techniques for Networked and Distributed Systems (FORTE'05), Taipei, Taiwan, ROC, October 2005*, volume 3731 of *Lecture Notes in Computer Science*, pages 53–67. Springer.

[CI-252⋆] Thomas Chatain, Loïc Hélouët, and Claude Jard. From automata networks to HMSCs: A reverse model engineering perspective. *In* Farn Wang, editor, *Proceedings of 25th IFIP WG6.1 International Conference on Formal Techniques for Networked and Distributed Systems (FORTE'05), Taipei, Taiwan, ROC, October 2005*, volume 3731 of *Lecture Notes in Computer Science*, pages 489–502. Springer.

[CI-253⋆] Sami Evangelista, Serge Haddad, and Jean-François Pradat-Peyre. Syntactical colored Petri nets reductions. *In* Doron A. Peled and Yih-Kuen Tsay, editors, *Proceedings of the*

3rd International Symposium on Automated Technology for Verification and Analysis (ATVA'05), Taipei, Taiwan, ROC, October 2005, volume 3707 of *Lecture Notes in Computer Science*, pages 202–216. Springer.

[CI-254⋆] Kais Klai, Serge Haddad, and Jean-Michel Ilié. Modular verification of Petri nets properties: A structure-based approach. *In* Farn Wang, editor, *Proceedings of 25th IFIP WG6.1 International Conference on Formal Techniques for Networked and Distributed Systems (FORTE'05), Taipei, Taiwan, ROC, October 2005*, volume 3731 of *Lecture Notes in Computer Science*, pages 189–203. Springer.

[CI-255⋆] Souheib Baarir, Claude Dutheillet, Serge Haddad, and Jean-Michel Ilié. On the use of exact lumpability in partially symmetrical well-formed nets. *In Proceedings of the 2nd International Conference on Quantitative Evaluation of Systems (QEST'05), Torino, Italy, September 2005*, pages 23–32. IEEE Computer Society Press.

[CI-256⋆] Michael Benedikt and Luc Segoufin. Towards a characterization of order-invariant queries over tame structures. *In* Luke Ong, editor, *Proceedings the 19th International Workshop on Computer Science Logic (CSL'05), Oxford, UK, August 2005*, volume 3634 of *Lecture Notes in Computer Science*, pages 276–291. Springer.

[CI-257⋆] Benedikt Bollig. On the expressiveness of asynchronous cellular automata. *In* Maciej Liskiewicz and Rüdiger Reischuk, editors, *Proceedings of the 15th International Symposium on Fundamentals of Computation Theory (FCT'05), Lübeck, Germany, August 2005*, volume 3623 of *Lecture Notes in Computer Science*, pages 528–539. Springer.

[CI-258⋆] Filippo Furfaro, Giuseppe M. Mazzeo, and Cristina Sirangelo. Clustering-based histograms for multi-dimensional data. *In* A. Min Tjoa and Juan Trujillo, editors, *Proceedings of the 7th International Conference Data Warehousing and Knowledge Discovery (DaWaK'05), Copenhagen, Denmark, August 2005*, volume 3589 of *Lecture Notes in Computer Science*, pages 478–487. Springer.

[CI-259⋆] Alessandro Armando, David A. Basin, Yohan Boichut, Yannick Chevalier, Luca Compagna, Jorge Cuéllar, Paul Hankes Drielsma, Pierre-Cyrille Héam, Olga Kouchnarenko, Jacopo Mantovani, Sebastian Mödersheim, David von Oheimb, Michaël Rusinowitch, Judson Santiago, Mathieu Turuani, Luca Viganò, and Laurent Vigneron. The AVISPA tool for the automated validation of internet security protocols and applications. *In* Kousha Etessami and Sriram Rajamani, editors, *Proceedings of the 17th International Conference on Computer Aided Verification (CAV'05), Edinburgh, Scotland, UK, July 2005*, volume 3576 of *Lecture Notes in Computer Science*, pages 281–285. Springer.

[CI-260⋆] Thomas Chatain and Claude Jard. Models for the supervision of web services orchestration with dynamic changes. *In Telecommunications 2005: Advanced Industrial Conference on Telecommunications – Service Assurance with Partial and Intermittent Resources Conference – E-Learning on Telecommunications Workshop (AICT/SAPIR/ELETE 2005), Lisbon, Portugal, July 2005*, pages 446–451. IEEE Computer Society Press.

[CI-261⋆] Graham Steel. Deduction with XOR constraints in security API modelling. *In* Robert Nieuwenhuis, editor, *Proceedings of the 20th International Conference on Automated Deduction (CADE'05), Tallinn, Estonia, July 2005*, volume 3632 of *Lecture Notes in Artificial Intelligence*, pages 322–336. Springer.

[CI-262⋆] Luciano Caroprese, Sergio Greco, Cristina Sirangelo, and Ester Zumpano. A logic based approach to P2P databases. *In* Andrea Calí, Diego Calvanese, Enrico Franconi, Maurizio Lenzerini, and Letizia Tanca, editors, *Proceedings of the 13th Italian Symposium on Advanced Database Systems (SEDB'05), Bressanone, Italy, June 2005*, pages 67–74.

[CI-263⋆] Luc Segoufin and Victor Vianu. Views and queries: determinacy and rewriting. *In* Chen Li, editor, *Proceedings of the 24th Annual ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'05), Baltimore, Maryland, USA, June 2005*, pages 49–60. ACM Press.

[CI-264⋆] François-Régis Sinot and Ian Mackie. Macros for interaction nets: A conservative extension of interaction nets. *In* Maribel Fernández, editor, *Proceedings of the 2nd International Workshop on Term Graph Rewriting (TERMGRAPH'04), Rome, Italy, October 2004*, volume 127(5) of *Electronic Notes in Theoretical Computer Science*, pages 153–169. Elsevier Science Publishers, May 2005.

[CI-265⋆] Emmanuel Beffara and François Maurel. Concurrent nets: a study of prefixing in process calculi. *In* Jos Baeten and Flavio Corradini, editors, *Proceedings of the 11th International Workshop on Expressiveness in Concurrency (EXPRESS'04), London, UK, August 2004*, volume 128(2) of *Electronic Notes in Theoretical Computer Science*, pages 67–86. Elsevier Science Publishers, April 2005.

[CI-266⋆] Ahmed Bouajjani, Peter Habermehl, Pierre Moro, and Tomáš Vojnar. Verifying programs with dynamic 1-selector-linked structures in regular model checking. *In* Nicolas Halbwachs and Lenore D. Zuck, editors, *Proceedings of the 11th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'05), Edinburgh, UK, April 2005*, volume 3440 of *Lecture Notes in Computer Science*, pages 13–29. Springer.

[CI-267⋆] Romain Janvier, Yassine Lakhnech, and Laurent Mazaré. Completing the picture: Soundness of formal encryption in the presence of active adversaries. *In* Mooly Sagiv, editor, *Programming Languages and Systems — Proceedings of the 14th European Symposium on Programming (ESOP'05), Edinburgh, U.K., April 2005*, volume 3444 of *Lecture Notes in Computer Science*, pages 172–185. Springer.

[CI-268⋆] François-Régis Sinot. Call-by-name and call-by-value as token-passing interaction nets. *In* Paweł Urzyczyn, editor, *Proceedings of the 7th International Conference on Typed Lambda Calculi and Applications (TLCA'05), Nara, Japan, April 2005*, volume 3461 of *Lecture Notes in Computer Science*, pages 386–400. Springer.

[CI-269⋆] Dietmar Berwanger and Erich Grädel. Entanglement – A measure for the complexity of directed graphs with applications to logic and games. *In* Franz Baader and Andrei Voronkov, editors, *Proceedings of the 11th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'04), Montevideo, Uruguay, March 2005*, volume 3452 of *Lecture Notes in Artificial Intelligence*, pages 209–223. Springer.

[CI-270⋆] Filippo Furfaro, Giuseppe M. Mazzeo, Domenico Saccà, and Cristina Sirangelo. Hierarchical binary histograms for summarizing multi-dimensional data. *In* Hisham Haddad, Lorie M. Liebrock, and Roger L. Omicini, Andrea andWainwright, editors, *Proceedings of the 20th ACM Symposium on Applied Computing (SAC'05), Santa Fe, New Mexico, USA, March 2005*, pages 598–603. ACM Press.

[CI-271⋆] Graham Steel and Alan Bundy. Attacking group multicast key management protocols using coral. *In* Alessandro Armando and Luca Viganò, editors, *Proceedings of the Workshop on Automated Reasoning for Security Protocol Analysis (ARSPA'04), Cork, Ireland, July 2004*, volume 125(1) of *Electronic Notes in Theoretical Computer Science*, pages 125–144. Elsevier Science Publishers, March 2005.

[CI-272⋆] Michael Benedikt and Luc Segoufin. Regular tree languages definable in FO. *In* Volker Diekert and Bruno Durand, editors, *Proceedings of the 22nd Annual Symposium on Theoretical Aspects of Computer Science (STACS'05), Stuttgart, Germany, February 2005*, volume 3404 of *Lecture Notes in Computer Science*, pages 327–339. Springer.

[CI-273⋆] Dietmar Berwanger and Giacomo Lenzi. The variable hierarchy of the μ-calculus is strict. *In* Volker Diekert and Bruno Durand, editors, *Proceedings of the 22nd Annual Symposium on Theoretical Aspects of Computer Science (STACS'05), Stuttgart, Germany, February 2005*, volume 3404 of *Lecture Notes in Computer Science*, pages 97–109. Springer.

[CI-274⋆] Shahrokh Saeednia, Steve Kremer, and Olivier Markowitch. An efficient strong designated verifier signature scheme. *In* Jong In Lim and Dong Hoon Lee, editors, *Revised Papers of the 6th International Conference on Information Security and Cryptology (ICISC'03), Seoul, Korea, November 2003*, volume 2971 of *Lecture Notes in Computer Science*, pages 40–54. Springer, 2004.

[CI-275⋆] Serge Haddad, Jean-Michel Ilié, and Kais Klai. Design and evaluation of a symbolic and abstraction-based model checker. *In* Farn Wang, editor, *Proceedings of the 2nd International Symposium on Automated Technology for Verification and Analysis (ATVA'04), Taipei, Taiwan, ROC, October-November 2004*, volume 3299 of *Lecture Notes in Computer Science*, pages 196–210. Springer.

[CI-276⋆] Serge Haddad and Patrice Moreaux. Approximate analysis of non-Markovian stochastic systems with multiple time scale delays. *In Proceedings of the 12th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'04), Volendam, The Netherlands, October 2004*, pages 23–30. IEEE Computer Society Press.

[CI-277⋆] Souheib Baarir, Serge Haddad, and Jean-Michel Ilié. Exploiting partial symmetries in well-formed nets for the reachability and the linear time model checking problems. *In Proceedings of the 7th Workshop on Discrete Event Systems (WODES'04), Reims, France, September 2004*, pages 223–228.

[CI-278⋆] Thomas Brihaye, Véronique Bruyère, and Jean-François Raskin. Model-checking for weighted timed automata. *In* Yassine Lakhnech and Sergio Yovine, editors, *Proceedings of the Joint Conferences Formal Modelling and Analysis of Timed Systems (FORMATS'04) and Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT'04), Grenoble, France, September 2004*, volume 3253 of *Lecture Notes in Computer Science*, pages 277–292. Springer.

[CI-279⋆] Thomas Chatain and Claude Jard. Symbolic diagnosis of partially observable concurrent systems. *In* David de Frutos-Escrig and Manuel NÚñez, editors, *Proceedings of 24th IFIP WG6.1 International Conference on Formal Techniques for Networked and Distributed Systems (FORTE'04), Madrid Spain, September 2004*, volume 3235 of *Lecture Notes in Computer Science*, pages 326–342. Springer.

[CI-280⋆] Joyce El Haddad and Serge Haddad. A fault-contained spanning tree protocol for arbitrary networks. *In* David A. Bader and Ashfaq A. Khokhar, editors, *Proceedings of the ISCA 17th International Conference on Parallel and Distributed Computing Systems (PDCS'04), San Francisco, CA, USA, September 2004*, pages 410–415. International Society for Computers and their Applications.

[CI-281⋆] Sami Evangelista, Serge Haddad, and Jean-François Pradat-Peyre. New coloured reductionsfor software validation. *In Proceedings of the 7th Workshop on Discrete Event Systems (WODES'04), Reims, France, September 2004*, pages 355–360.

[CI-282⋆] Serge Haddad, Lynda Mokdad, and Patrice Moreaux. Performance evaluation of non-Markovian stochastic discrete event systems – a new approach. *In Proceedings of the 7th Workshop on Discrete Event Systems (WODES'04), Reims, France, September 2004*.

[CI-283⋆] Christel Baier, Marcus Größer, Martin Leucker, Benedikt Bollig, and Franck Ciesinski. Probabilistic controller synthesis. *In* Jean-Jacques Lévy, Ernst W. Mayr, and John C. Mitchell, editors, *Proceedings of the 3rd IFIP International Conference on Theoretical Computer Science (IFIP TCS'04), Toulouse, France, August 2004*, IFIP Conference Proceedings, pages 493–506. Kluwer Academic Publishers.

[CI-284⋆] Benedikt Bollig and Martin Leucker. Message-passing automata are expressively equivalent to EMSO logic. *In* Philippa Gardner and Nobuko Yoshida, editors, *Proceedings of the 15th International Conference on Concurrency Theory (CONCUR'04), London, UK, August 2004*, volume 3170 of *Lecture Notes in Computer Science*, pages 146–160. Springer.

[CI-285⋆] Luis Caires and Étienne Lozes. Elimination of quantifiers and undecidability in spatial logics for concurrency. *In* Philippa Gardner and Nobuko Yoshida, editors, *Proceedings of the 15th International Conference on Concurrency Theory (CONCUR'04), London, UK, August 2004*, volume 3170 of *Lecture Notes in Computer Science*, pages 240–257. Springer.

[CI-286⋆] Sergio Greco, Cristina Sirangelo, Irina Trubitsyna, and Ester Zumpano. Feasibility conditions and preference criteria in querying and repairing inconsistent databases. *In* Fernando Galindo, Makoto Takizawa, and Roland Traunmüller, editors, *Proceedings of the 15th International Conference on Database and Expert Systems Applications (DEXA'04), Zaragoza, Spain, August-September 2004*, volume 3180 of *Lecture Notes in Computer Science*, pages 44–55. Springer.

[CI-287⋆] Graham Steel, Alan Bundy, and Monika Maidl. Attacking a protocol for group key agreement by refuting incorrect inductive conjectures. *In* David A. Basin and Michaël Rusinowitch, editors, *Proceedings of the 2nd International Joint Conference on Automated Reasoning (IJCAR'04), Cork, Ireland, August 2004*, volume 3097 of *Lecture Notes in Artificial Intelligence*, pages 137–151. Springer-Verlag.

[CI-288⋆] Ahmed Bouajjani, Peter Habermehl, and Tomáš Vojnar. Abstract regular model checking. *In* Rajeev Alur and Doron A. Peled, editors, *Proceedings of the 16th International Conference on Computer Aided Verification (CAV'04), Boston, Massachusetts, USA, July 2004*, volume 3114 of *Lecture Notes in Computer Science*, pages 372–386. Springer.

[CI-289⋆] Helmut Seidl, Thomas Schwentick, Anca Muscholl, and Peter Habermehl. Counting in trees for free. *In* Josep Díaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannella, editors, *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP'04), Turku, Finland, July 2004*, volume 3142 of *Lecture Notes in Computer Science*, pages 1136–1149. Springer.

[CI-290⋆] Rohit Chadha, Steve Kremer, and Andre Scedrov. Formal analysis of multi-party contract signing. *In Proceedings of the 17th IEEE Computer Security Foundations Workshop (CSFW'04), Asilomar, Pacific Grove, California, USA, June 2004*, pages 266–279. IEEE Computer Society Press.

[CI-291⋆] Filippo Furfaro, Giuseppe M. Mazzeo, Domenico Saccà, and Cristina Sirangelo. A new histogram-based technique for compressing multi-dimensional data. *In* Maristella Agosti, Nicoletta Dess, and Fabio A. Schreiber, editors, *Proceedings of the 12th Italian Symposium on Advanced Database Systems (SEDB'04), S. Margherita di Pula, Cagliari, Italy, June 2004*, pages 18–29.

[CI-292⋆] Lars M. Kristensen and Laure Petrucci. An approach to distributed state space exploration for coloured Petri nets. *In* Jordi Cortadella and Wolfgang Reisig, editors, *Proceedings of the 25th International Conference on Applications and Theory of Petri Nets (ICATPN'04), Bologna, Italy, June 2004*, volume 3099 of *Lecture Notes in Computer Science*, pages 474–483. Springer.

[CI-293⋆] Charles Lakos and Laure Petrucci. Modular analysis of systems composed of semi-autonomous subsystems. *In* Mike Kishinevsky and Philippe Darondeau, editors, *Proceedings of the 4th International Conference on Application of Concurrency to System Design (ACSD'04), Hamilton, Ontario, Canada, June 2004*, pages 185–194. IEEE Computer Society Press.

[CI-294⋆] Étienne Lozes. Adjuncts elimination in the static ambient logic. *In* Flavio Corradini and Uwe Nestmann, editors, *Proceedings of the 10th International Workshop on Expressiveness in Concurrency (EXPRESS'03), Marseilles, France, September 2003*, volume 96 of *Electronic Notes in Theoretical Computer Science*, pages 51–72. Elsevier Science Publishers, June 2004.

[CI-295⋆] Jean Cardinal, Steve Kremer, and Stefan Langerman. Juggling with pattern matching. *In* Paolo Ferragina and Roberto Grossi, editors, *Proceedings of the 3rd International Conference on Fun with Algorithms (FUN'04), Isola d'Elba, Italy, May 2004*, pages 147–158. Edizioni Plus, Università di Pisa.

[CI-296⋆] Volker Diekert and Paul Gastin. Pure future local temporal logics are expressively complete for Mazurkiewicz traces. *In* Martin Farach-Colton, editor, *Proceedings of the 6th Latin American Symposium on Theoretical Informatics (LATIN'04), Buenos Aires, Argentina, April 2004*, volume 2976 of *Lecture Notes in Computer Science*, pages 232–241. Springer.

[CI-297⋆] Paul Gastin, Benjamin Lerman, and Marc Zeitoun. Distributed games and distributed control for asynchronous systems. *In* Martin Farach-Colton, editor, *Proceedings of the 6th Latin American Symposium on Theoretical Informatics (LATIN'04), Buenos Aires, Argentina, April 2004*, volume 2976 of *Lecture Notes in Computer Science*, pages 455–465. Springer.

[CI-298⋆] Paul Gastin, Pierre Moro, and Marc Zeitoun. Minimization of counterexamples in SPIN. *In* Susanne Graf and Laurent Mounier, editors, *Proceedings of the 11th International SPIN Workshop on Model Checking Software (SPIN'04), Barcelona, Spain, April 2004*, volume 2989 of *Lecture Notes in Computer Science*, pages 92–108. Springer.

[CI-299⋆] Thomas Brihaye, Christian Michaux, Cédric Rivière, and Christophe Troestler. On o-minimal hybrid systems. *In* Rajeev Alur and George J. Pappas, editors, *Proceedings of the 7th International Conference on Hybrid Systems: Computation and Control (HSCC'04), Philadelphia, Pennsylvania, USA, March 2004*, volume 2993 of *Lecture Notes in Computer Science*, pages 219–233. Springer.

[CI-300⋆] Anca Muscholl, Thomas Schwentick, and Luc Segoufin. Active context-free games. *In* Volker Diekert and Michel Habib, editors, *Proceedings of the 21st Annual Symposium on Theoretical Aspects of Computer Science (STACS'04), Montpellier, France, February 2004*, volume 2996 of *Lecture Notes in Computer Science*, pages 452–464. Springer.

Autres communications dans des conférences

[CO-1] Elie Bursztein. NetQi: A model checker for anticipation game. *In* Moonzoo Kim and Mahesh Viswanathan, editors, *Proceedings of the 6th International Symposium on Automated Technology for Verification and Analysis (ATVA'08), Seoul, Korea, October 2008*, Lecture Notes in Computer Science. Springer. To appear.

[CO-2] Benedikt Bollig, Joost-Pieter Katoen, Carsten Kern, and Martin Leucker. *Smyle*: A tool for synthesizing distributed models from scenarios by learning. *In* Franck van Breugel and Marsha Chechik, editors, *Proceedings of the 19th International Conference on Concurrency Theory (CONCUR'08), Toronto, Canada, August 2008*, volume 5201 of *Lecture Notes in Computer Science*, pages 162–166. Springer.

[CO-3] Sławomir Lasota, David Nowak, and Zhang Yu. On completeness of logical relations for monadic types. *In* Mitsu Okada and Ichiro Satoh, editors, *Revised Selected Papers of the 11th Asian Computing Science Conference (ASIAN'06), Tokyo, Japan, December 2006*, volume 4435 of *Lecture Notes in Computer Science*, pages 223–230. Springer, January 2008.

[CO-4] Alain Finkel, Étienne Lozes, and Arnaud Sangnier. Towards model checking pointer systems. *In* Benedikt Löwe, editor, *Proceedings of the International Conference on Infinity in Logic & Computation (ILC'07), Cape Town, South Africa, November 2007*.

[CO-5] Elie Bursztein. Time has something to tell us about network address translation. *In* Úlfar Erlingsson and Andrei Sabelfeld, editors, *Proceedings of the 12th Nordic Workshop on Secure IT Systems (NordSec'07), Reykjavik, Iceland, October 2007*.

[CO-6] Elie Bursztein. Network incidents anticipation. *In* Christopher Kruegel, Richard Lippmann, and Andrew Clark, editors, *Proceedings of the 10th International Symposium on Recent Advances in Intrusion Detection (RAID'07), Gold Coast, Australia, September 2007*, volume 4637 of *Lecture Notes in Computer Science*. Springer. Poster presentation.

[CO-7] Cas Cremers and Pascal Lafourcade. Comparing state spaces in automatic security protocol verification. *In* Michael Goldsmith and Bill Roscoe, editors, *Proceedings of the 7th International Workshop on Automated Verification of Critical Systems (AVoCS'07), Oxford, UK, September 2007*, Electronic Notes in Theoretical Computer Science, pages 49–63. Elsevier Science Publishers.

[CO-8] Stéphanie Delaune, Steve Kremer, and Mark D. Ryan. Symbolic bisimulation for the applied pi calculus. *In* Daniele Goria and Catuscia Palamidessi, editors, *Preliminary Proceedings of the 5th International Workshop on Security Issues in Concurrency (SecCo'07), Lisbon, Portugal, September 2007*.

[CO-9] Adel Bouhoula and Florent Jacquemard. Verifying regular trace properties of security protocols with explicit destructors and implicit induction. *In* Pierpaolo Degano, Ralf Küsters, Luca Viganò, and Steve Zdancewic, editors, *Proceedings of the Joint Workshop on Foundations of Computer Security and Automated Reasoning for Security Protocol Analysis (FCS-ARSPA'07), Wrocław, Poland, July 2007*, pages 27–44.

[CO-10] Véronique Cortier and Stéphanie Delaune. Deciding knowledge in security protocols for monoidal equational theories. *In* Pierpaolo Degano, Ralf Küsters, Luca Viganò, and Steve Zdancewic, editors, *Proceedings of the Joint Workshop on Foundations of Computer Security and Automated Reasoning for Security Protocol Analysis (FCS-ARSPA'07), Wrocław, Poland, July 2007*, pages 63–80.

[CO-11] Stéphanie Delaune, Hai Lin, and Christopher Lynch. Protocol verification via rigid/ flexible resolution. *In* Silvio Ghilardi, Ulrike Sattler, Viorica Sofronie-Stokkermans, and Ashish Tiwari, editors, *Proceedings of the Workshop on Automated Deduction: Decidability, Complexity, Tractability (ADDCT'07), Bremen, Germany, July 2007*, pages 2–16.

[CO-12] Steve Kremer and Laurent Mazaré. Adaptive soundness of static equivalence. *In* Michael Backes and Yassine Lakhnech, editors, *Proceedings of the 3rd Workshop on Formal and Computational Cryptography (FCC'07), Venice, Italy, July 2007*.

[CO-13] Benedikt Bollig and Dietrich Kuske. Muller message-passing automata and logics. *In* Zoltán Ésik, Carlos Martín-Vide, and Victor Mitrana, editors, *Preliminary Proceedings of the 1st International Conference on Language and Automata Theory and Applications (LATA'07), Tarragona, Spain, March-April 2007*.

[CO-14] Rémi Brochenin, Stéphane Demri, and Étienne Lozes. Reasoning about sequences of memory states. *In* Josh Berdine and Mooly Sagiv, editors, *Proceedings of the 1st Workshop on Heap Analysis and Verification (HAV'07), Braga, Portugal, March 2007*.

[CO-15] Laurent Mazaré. Computationally sound analysis of protocols using bilinear pairings. *In* Riccardo Focardi, editor, *Preliminary Proceedings of the 7th International Workshop on Issues in the Theory of Security (WITS'07), Braga, Portugal, March 2007*, pages 6–21.

[CO-16] Adel Bouhoula and Florent Jacquemard. Automating sufficient completeness check for conditional and constrained TRS. *In* Jordi Levy, editor, *Proceedings of the 20th International Workshop on Unification (UNIF'06), Seattle, Washington, USA, August 2006*.

[CO-17] Pascal Lafourcade, Denis Lugiez, and Ralf Treinen. ACUNh: Unification and disunification using automata theory. *In* Jordi Levy, editor, *Proceedings of the 20th International Workshop on Unification (UNIF'06), Seattle, Washington, USA, August 2006*, pages 6–20.

[CO-18] Adel Bouhoula and Florent Jacquemard. Security protocols verification with implicit induction and explicit destructors. *In* Maribel Fernández and Claude Kirchner, editors, *Preliminary Proceedings of the 1st International Workshop on Security and Rewriting Techniques (SecReT'06), Venice, Italy, July 2006*, pages 37–44.

[CO-19] Stéphanie Delaune, Steve Kremer, and Mark D. Ryan. Verifying properties of electronic voting protocols. *In Proceedings of the IAVoSS Workshop On Trustworthy Elections (WOTE'06), Cambridge, UK, June 2006*, pages 45–52.

[CO-20] Sébastien Bardin, Alain Finkel, Étienne Lozes, and Arnaud Sangnier. From pointer systems to counter systems using shape analysis. *In* Ramesh Bharadwaj, editor, *Proceedings of the 5th International Workshop on Automated Verification of Infinite-State Systems (AVIS'06), Vienna, Austria, April 2006*.

[CO-21] Régis Gascon. Verifying qualitative and quantitative properties with LTL over concrete domains. *In* Holger Schlingloff, editor, *Proceedings of the 4th Workshop on Methods for Modalities (M4M-4), Berlin, Germany, December 2005*, volume 194 of *Informatik Bericht*, pages 54–61. Humboldt Universität zu Berlin.

[CO-22] Stéphanie Delaune, Steve Kremer, and Mark D. Ryan. Receipt-freeness: Formal definition and fault attacks (extended abstract). *In Proceedings of the Workshop Frontiers in Electronic Elections (FEE 2005), Milan, Italy, September 2005*.

[CO-23] Sławomir Lasota, David Nowak, and Yu Zhang. On completeness of logical relations for monadic types. *In* Martin Hofmann and Hans-Wolfgang Loidl, editors, *Proceedings of the 3rd APPSEM II Workshop (APPSEM'05), Frauenchiemsee, Germany, September 2005*.

[CO-24] Laurent Fribourg and Stéphane Messika. Brief announcement: Coupling for Markov decision processes — Application to self-stabilization with arbitrary schedulers. *In* Marcos Kawazoe Aguilera and James Aspnes, editors, *Proceedings of the Twenty-Fourth Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC'05), Las Vegas, Nevada, USA, July 2005*, page 322. ACM Press.

[CO-25] Manuel Baclet and Rémy Chevallier. Timed verification of the SPSMALL memory. *In Proceedings of the 1st International Conference on Memory Technology and Design (ICMTD'05), Giens, France, May 2005*, pages 89–92.

[CO-26] Jean-Michel Couvreur. A BDD-like implementation of an automata package. *In* Michael Domaratzki, Alexander Okhotin, Kai Salomaa, and Sheng Yu, editors, *Revised Selected Papers of the 9th International Conference on Implementation and Application of Automata (CIAA'04), Kingston, Ontario, Canada, July 2004*, volume 3317 of *Lecture Notes in Computer Science*, pages 310–311. Springer, January 2005.

[CO-27] Stéphanie Delaune and Francis Klay. Vérification automatique appliquée à un protocole de commerce électronique. *In Actes des 6èmes Journées Doctorales Informatique et Réseau (JDIR'04), Lannion, France, November 2004*, pages 260–269.

[CO-28] Nicolas Markey and Philippe Schnoebelen. TSMV: A symbolic model checker for quantitative analysis of systems. *In Proceedings of the 1st International Conference on Quantitative Evaluation of Systems (QEST'04), Enschede, The Netherlands, September 2004*, pages 330–331. IEEE Computer Society Press.

[CO-29] Adel Bouhoula and Florent Jacquemard. Constrained tree grammars to pilot automated proof by induction. *In* Maria Paola Bonacina and Thierry Boy de la Tour, editors, *Proceedings of the 5th Workshop on Strategies in Automated Deduction (STRATEGIES'04), Cork, Ireland, July 2004*, pages 64–78.

[CO-30] Stéphanie Delaune and Florent Jacquemard. Narrowing-based constraint solving for the verification of security protocols. *In* Michael Kohlhase, editor, *Proceedings of the 18th International Workshop on Unification (UNIF'04), Cork, Ireland, July 2004.*

[CO-31] Sébastien Bardin, Christophe Darlot, and Alain Finkel. FAST: un model-checker pour systèmes à compteurs. *In* Jacques Julliand, editor, *Actes du 6ème Atelier sur les Approches Formelles dans l'Assistance au Développement de Logiciels (AFADL'04), Besançon, France, June 2004*, pages 377–380.

[CO-32] Sébastien Bardin and Laure Petrucci. COAST: des réseaux de Petri à la planification assistée. *In* Jacques Julliand, editor, *Actes du 6ème Atelier sur les Approches Formelles dans l'Assistance au Développement de Logiciels (AFADL'04), Besançon, France, June 2004*, pages 285–298.

[CO-33] Sébastien Bardin and Laure Petrucci. From PNML to counter systems for accelerating Petri nets with FAST. *In* Ekkart Kindler, editor, *Proceedings of the Workshop on Interchange Format for Petri Nets, Bologna, Italy, June 2004*, pages 26–40.

[CO-34] Mathieu Baudet. Random polynomial-time attacks and Dolev-Yao models. *In* Siva Anantharaman, editor, *Proceedings of the Workshop on Security of Systems: Formalism and Tools (SASYFT'04), Orléans, France, June 2004.*

[CO-35] Mongi Ben Gaid, Béatrice Bérard, and Olivier De Smet. Modélisation et vérification d'un évaporateur en Uppaal. *In* Jacques Julliand, editor, *Actes du 6ème Atelier sur les Approches Formelles dans l'Assistance au Développement de Logiciels (AFADL'04), Besançon, France, June 2004*, pages 223–238.

[CO-36] Nicolas Markey. TSMV: model-checking symbolique de systèmes simplement temporisés. *In* Jacques Julliand, editor, *Actes du 6ème Atelier sur les Approches Formelles dans l'Assistance au Développement de Logiciels (AFADL'04), Besançon, France, June 2004*, pages 349–352.

[CO-37] Olivier Michel and Florent Jacquemard. An analysis of the Needham-Schroeder public-key protocol with MGS. *In* Gheorghe Paun, editor, *Proceedings of the 5th Workshop on Membrane Computing (WMC'04), Milano, Italy, June 2004*, pages 295–315.

[CO-38] Sébastien Bardin, Alain Finkel, and David Nowak. Toward symbolic verification of programs handling pointers. *In* Ramesh Bharadwaj, editor, *Proceedings of the 3rd International Workshop on Automated Verification of Infinite-State Systems (AVIS'04), Barcelona, Spain, April 2004.*

[CO-39] Nathalie Bertrand and Philippe Schnoebelen. Verifying nondeterministic channel systems with probabilistic message losses. *In* Ramesh Bharadwaj, editor, *Proceedings of the 3rd International Workshop on Automated Verification of Infinite-State Systems (AVIS'04), Barcelona, Spain, April 2004.*

[CO-40⋆] Sylvain Schmitz and Joseph Le Roux. Feature unification in TAG derivation trees. *In* Claire Gardent and Anoop Sarkar, editors, *Proceedings of the 9th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+'08), Tübingen, Germany, June 2008*, pages 141–148.

[CO-41⋆] Marco Beccuti, Daniele Codetta-Raiteri, Giuliana Franceschinis, and Serge Haddad. A framework to design and solve Markov decision well-formed net models. *In Proceedings*

of the 4th International Conference on Quantitative Evaluation of Systems (QEST'07), Edinburgh, Scotland, September 2007, pages 165–166. IEEE Computer Society Press.

[CO-42⋆] Mehdi Ben Hmida, Céline Boutrous-Saab, Serge Haddad, Valérie Monfort, and Ricardo Ferraz Tomaz. Towards the dynamic adaptability of SOA. *In* Jorge Cardoso, José Cordeiro, and Joaquim Filipe, editors, *Proceedings of the 9th International Conference on Enterprise Information Systems (ICEIS'07), volume EIS, Funchal, Madeira, Portugal, June 2007*, pages 474–479.

[CO-43⋆] Pierre-Cyrille Héam, Olga Kouchnarenko, and Jérôme Voinot. How to handle QoS aspects in web services substitutivity verification. *In Proceedings of the IEEE International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE'07), Paris, France, June 2007*, pages 333–338. IEEE Computer Society Press.

[CO-44⋆] Mehdi Ben Hmida and Serge Haddad. Vers l'adaptabilité dynamique des architectures orientées services. *In Actes des 3èmes Journées Francophones sur le Développement de Logiciels Par Aspects (JFDLPA'07), Toulouse, France, March 2007*, pages 73–88.

[CO-45⋆] Mehdi Ben Hmida, Céline Boutrous-Saab, Serge Haddad, Valérie Monfort, and Ricardo Ferraz Tomaz. Dynamically adapting clients to web services changing. *In* Cesare Pautasso and Christoph Bussler, editors, *Proceedings of the Workshop on Emerging Web Services Technology (WEWST'06), Zurich, Switzerland, December 2006*, pages 91–96.

[CO-46⋆] Christine Choppy, Serge Haddad, Hanna Klaudel, Fabrice Kordon, Laure Petrucci, and Yann Thierry-Mieg. Tutorial on formal methods for distributed and cooperative systems. *In* Kamel Barkaoui, Ana Cavalcanti, and Antonio Cerone, editors, *Proceedings of the 3rd International Colloquium on Theoretical Aspects of Computing (ICTAC'06), Tunis, Tunisia, November 2006*, volume 4281 of *Lecture Notes in Computer Science*, pages 362–365. Springer.

[CO-47⋆] Adam Antonik, Nathaniel Charlton, and Michael Huth. Computing safe winning regions of parity games in polynomial time. *In Proceedings of the 4th Irish Conference on the Mathematical Foundations of Computer Science and Information Technology'06 (MFCSIT'06), Cork, Ireland, August 2006*.

[CO-48⋆] Romain Janvier, Yassine Lakhnech, and Laurent Mazaré. Relating the symbolic and computational models of security protocols using hashes. *In* Pierpaolo Degano, Ralf Küsters, Luca Viganò, and Steve Zdancewic, editors, *Proceedings of the Joint Workshop on Foundations of Computer Security and Automated Reasoning for Security Protocol Analysis (FCS-ARSPA'06), Seattle, Washington, USA, August 2006*.

[CO-49⋆] Yassine Lakhnech, Laurent Mazaré, and Bogdan Warinschi. Soundness of symbolic equivalence for modular exponentiation. *In* Véronique Cortier and Steve Kremer, editors, *Proceedings of the 2nd Workshop on Formal and Computational Cryptography (FCC'06), Venice, Italy, July 2006*, pages 19–23.

[CO-50⋆] Mbaye Sene, Patrice Moreaux, and Serge Haddad. Performance evaluation of distributed database – a banking system case study. *In* Alexandre Dolgui, Gérard Morel, and Carlos E. Pereira, editors, *Proceedings of the 12th IFAC Symposium on Information Control Problems in Manufacturing (INCOM'06), Saint-Étienne, France, May 2006*, pages 351–356.

[CO-51⋆] Sergio Greco, Cristina Sirangelo, Irina Trubitsyna, and Ester Zumpano. Preferred repairs for inconsistent databases. *In Encyclopedia of Database Technologies and Applications*, pages 480–485. Idea Group, 2005.

[CO-52⋆] Claude Jard, Thomas Chatain, and Pierre Bourhis. Diagnostic temporel dans les systèmes répartis à l'aide de dépliages de réseaux de Petri temporels. *In* Hassane Alla and Éric Rutten, editors, *Actes du 5ème Colloque sur la Modélisation des Systèmes Réactifs (MSR'05), Autrans, France, October 2005*, pages 351–365. Hermès.

[CO-53⋆] Tarek Melliti, Serge Haddad, Alexandru Suna, and Amal El Fallah-Seghrouchni. Web-MASI: Multi-agent systems interoperability using web services based approach. *In* Andrzej Skowron, Jean-Paul A. Barth?s, Lakhmi C. Jain, Ron Sun, Pierre Morizet-Mahoudeaux, Jiming Liu, and Ning Zhong, editors, *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'05), Compiègne, France, September 2005*, pages 739–742. IEEE Computer Society Press.

[CO-54⋆] François-Régis Sinot. Explicit rewriting. *In Proceedings of the 2nd Workshop on the Rho-Calculus, Palaiseau, France, May 2005*.

[CO-55⋆] Thomas Chatain. Diagnostic pour les systèmes distribués dynamiques partiellement observables. *In* Richard Castanet, editor, *Actes du 11ème Colloque Francophone sur l'Ingénierie des Protocoles, Bordeaux, France, April 2005*. Hermès.

[CO-56⋆] Liana Bozga, Cristian Ene, Romain Janvier, Yassine Lakhnech, Laurent Mazaré, and Michaël Périn. Automatic verification of security properties based on abstractions. *In* Edmund M. Clarke, Marius Minea, and Ferucio Laurentiu Tiplea, editors, *Proceedings of the NATO Advanced Research Workshop on Verification of Infinite State Systems with Applications to Security (VISSAS'05), Timisoara, Romania, March 2005*, volume 1 of *NATO Security through Science Series D: Information and Communication Security*, pages 23–53. IOS Press.

[CO-57⋆] Laurent Mazaré. Satisfiability of Dolev-Yao constraints. *In* Alessandro Armando and Luca Viganò, editors, *Proceedings of the Workshop on Automated Reasoning for Security Protocol Analysis (ARSPA'04), Cork, Ireland, July 2004*, volume 125(1) of *Electronic Notes in Theoretical Computer Science*, pages 109–124. Elsevier Science Publishers, March 2005.

[CO-58⋆] Amal El Fallah-Seghrouchni, Serge Haddad, Tarek Melliti, and Alexandru Suna. Interopérabilité des systèmes multi-agents à l'aide des services web. *In* Boissier Olivier and Zahia Guessoum, editors, *Actes des 12èmes Journées Francophones des Systèmes Multi-Agents (JFSMA'04), Paris, France, November 2004*, pages 91–104. Hermès.

[CO-59⋆] Myrto Arapinis and Anatol Slissenko. A pattern based language for programming of heuristics of proof search. *In* Dana Petcu, Viorel Negru, Daniela Zaharie, and Tudor Jebelean, editors, *Proceedings of the 6th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'04), Timisoara, Romania, September 2004*, pages 400–411. Mirton Publisher.

[CO-60⋆] Rohit Chadha, Steve Kremer, and Andre Scedrov. Formal analysis of multi-party contract signing. *In* Peter Ryan, editor, *Preliminary Proceedings of the 4th IFIP WG1.7 Workshop on Issues in the Theory of Security (WITS'04), Barcelona, Spain, April 2004*, pages 153–163.

[CO-61⋆] Serge Haddad, Tarek Melliti, Patrice Moreaux, and Sylvain Rampacek. A dense time semantics for web services specification languages. *In Proceedings of the IEEE 1st International Conference on Information and Communication Technologies: From Theory to Applications (ICCTA'04), Damascus, Syria, April 2004*, pages 647–648.

[CO-62⋆] Serge Haddad, Tarek Melliti, Patrice Moreaux, and Sylvain Rampacek. Modeling web services interoperability. *In Proceedings of the 8th International Conference on Enterprise Information Systems (ICEIS'06), Porto, Portugal, April 2004*, pages 287–295.

[CO-63⋆] Laurent Mazaré. Using unification for opacity properties. *In* Peter Ryan, editor, *Preliminary Proceedings of the 4th IFIP WG1.7 Workshop on Issues in the Theory of Security (WITS'04), Barcelona, Spain, April 2004.*

[CO-64⋆] Étienne Lozes. Separation logic preserves the expressive power of classical logic. *In Proceedings of the 2nd Workshop on Semantics, Program Analysis, and Computing Environments for Memory Management (SPACE'04), Venice, Italy, January 2004.*

Thèses

[TH-1] Régis Gascon. Spécification et vérification de propriétés quantitatives sur des automates à contraintes. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, November 2007.

[TH-2] Fabrice Chevalier. Logiques pour les systèmes temporisés : contrôle et expressivité. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, June 2007.

[TH-3] Stéphane Demri. Logiques pour la spécification et vérification. Mémoire d'habilitation, Université Paris 7, Paris, France, June 2007.

[TH-4] Emmanuelle Encrenaz-Tiphène. Contributions pour la conception et la vérification de systèmes matériels embarqués. Mémoire d'habilitation, Université Paris 6, Paris, France, June 2007.

[TH-5] Pierre-Alain Reynier. Vérification de systèmes temporisés et distribués : modèles, algorithmes et implémentabilité. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, June 2007.

[TH-6] Mathieu Baudet. Sécurité des protocoles cryptographiques : aspects logiques et calculatoires. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, January 2007.

[TH-7] Nathalie Bertrand. Modèles stochastiques pour les pertes de messages dans les protocoles asynchrones et techniques de vérification automatique. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, October 2006.

[TH-8] Houda Bel mokadem. Vérification des propriétés temporisées des automates programmables industriels. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, September 2006.

[TH-9] Pascal Lafourcade. Vérification des protocoles cryptographiques en présence de théories équationnelles. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, September 2006. 209 pages.

[TH-10] Vincent Bernat. Théories de l'intrus pour la vérification des protocoles cryptographiques. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, June 2006.

[TH-11] Stéphanie Delaune. Vérification des protocoles cryptographiques et propriétés algébriques. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, June 2006.

[TH-12] Manuel Baclet. Applications du model-checking à des problèmes de vérification de systèmes sur puce. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, December 2005.

[TH-13] François Laroussinie. Model checking temporisé — algorithmes efficaces et complexité. Mémoire d'habilitation, Université Paris 7, Paris, France, December 2005.

[TH-14] Ralf Treinen. Résolution symbolique de contraintes. Mémoire d'habilitation, Université Paris-Sud 11, Orsay, France, November 2005.

[TH-15] Sébastien Bardin. Vers un model checking avec accélération plate de systèmes hétérogènes. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, October 2005.

[TH-16] Yu Zhang. Cryptographic logical relations — what is the contextual equivalence for cryptographic protocols and how to prove it? Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, October 2005.

[TH-17] Stéphane Messika. Méthodes probabilistes pour la vérification des systèmes distribués. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, December 2004.

[TH-18] Jean-Michel Couvreur. Contribution à l'algorithmique de la vérification. Mémoire d'habilitation, Université de Bordeaux I, Bordeaux, France, July 2004.

[TH-19⋆] Delphine Longuet. Test à partir de spécifications axiomatiques. Thèse de doctorat, Laboratoire IBISC, Université d'Évry-Val d'Essonne, France, October 2007.

[TH-20⋆] Sylvain Schmitz. Approximating context-free grammars for parsing and verification. Thèse de doctorat, Laboratoire I3S, Université de Nice-Sophia Antipolis, France, September 2007.

[TH-21⋆] Thomas Chatain. Dépliages symboliques de réseaux de Petri de haut niveau et application à la supervision des systèmes répartis. Thèse de doctorat, Université Rennes 1, Rennes, France, November 2006.

[TH-22⋆] Laurent Mazaré. Computational soundness of symbolic models for cryptographic protocols. Thèse de doctorat, Institut National Polytechnique de Grenoble, France, October 2006.

[TH-23⋆] François-Régis Sinot. Stratégies efficaces et modèles d'implantation pour les langages fonctionnels. Thèse de doctorat, École Polytechnique, Palaiseau, France, September 2006.

[TH-24⋆] Thomas Brihaye. Verification and control of o-minimal hybrid systems and weighted timed automata. Thèse de doctorat, Université Mons-Hainault, Mons, Belgium, May 2006.

[TH-25⋆] Dietmar Berwanger. Games and logical expressiveness. Ph.D. Thesis, Department of Computer Science, RWTH Aachen, Germany, 2005.

[TH-26⋆] Emmanuel Beffara. Logique, réalisabilité et concurrence. Thèse de doctorat, Université Paris 7, Paris, France, December 2005.

[TH-27⋆] Cristina Sirangelo. Approximate query answering on multi-dimensional data. Ph.D. Thesis, University of Calabria, Cosenza, Italy, November 2005.

[TH-28⋆] Benedikt Bollig. Automata and logics for message sequence charts. Thèse de doctorat, Department of Computer Science, RWTH Aachen, Germany, May 2005.

[TH-29⋆] Étienne Lozes. Expressivité des logiques spatiales. Thèse de doctorat, Laboratoire de l'Informatique du Parallélisme, ENS Lyon, France, November 2004.

[TH-30⋆] Graham Steel. Discovering attacks on security protocols by refuting incorrect inductive conjectures. Ph.D. Thesis, School of Informatics, University of Edinburgh, UK, May 2004.

Notes de cours

[Nc-1] Hubert Comon-Lundh. Soundness of abstract cryptography, 2007. Course notes (part 1), Symposium on Cryptography and Information Security (SCIS2008), Tokai, Japan.

[Nc-2] Jean-Loup Carré. Réécriture, confluence. Course notes, Préparation à l'agrégation, ENS Cachan, France, December 2007.

[Nc-3] Paul Gastin. Algorithmique. Course notes, Magistère STIC, ENS Cachan, France, November 2007.

[Nc-4] Laurent Fribourg. Probabilistic self-stabilizing algorithms, Markov chains and Markov decision processes. Course notes, Master Parisien de Recherche en Informatique, Paris, France, October-November 2007.

[Nc-5] Paul Gastin. Langages formels. Course notes, Magistère STIC, ENS Cachan, France, May 2007.

[Nc-6] Jean Goubault-Larrecq. Cours de complexité 2. Course notes, Magistère STIC, ENS Cachan, France, 2006.

[Nc-7] Nicolas Markey. Temporal logics. Course notes, Master Parisien de Recherche en Informatique, Paris, France, 2006.

[Nc-8] Stéphane Demri. Temporal logics, 2005. Course notes, Master Parisien de Recherche en Informatique, Paris, France.

[Nc-9] Stéphane Demri. Complexité algorithmique de variantes de LTL pour la vérification, 2004. Course notes, DEA Algorithmique, Paris, France.

[Nc-10] Ralf Treinen. Cours de complexité, second semestre, première année, magistère STIC, 2004. Course notes, Magistère STIC, ENS Cachan, France.

Rapports internes non publiés par ailleurs

[Ra-1] Ştefan Ciobâcă. *Verification of anonymity properties in e-voting protocols*. Rapport de Master, Master Parisien de Recherche en Informatique, Paris, France, September 2008.

[Ra-2] Stéphane Demri and Denis Lugiez. Complexity of modal logics with Presburger constraints. Research Report LSV-08-25, Laboratoire Spécification et Vérification, ENS Cachan, France, September 2008.

[Ra-3] Guillem Godoy and Florent Jacquemard. Unique normalization for shallow TRS. Research Report LSV-08-21, Laboratoire Spécification et Vérification, ENS Cachan, France, June 2008. 17 pages.

[Ra-4] Jean Goubault-Larrecq. A cone-theoretic Krein-Milman theorem. Research Report LSV-08-18, Laboratoire Spécification et Vérification, ENS Cachan, France, June 2008. 8 pages.

[Ra-5] Thomas Brihaye, Arnaud Da Costa, François Laroussinie, and Nicolas Markey. ATL with strategy contexts and bounded memory. Research Report LSV-08-14, Laboratoire Spécification et Vérification, ENS Cachan, France, April 2008. 22 pages.

[Ra-6] Véronique Cortier and Stéphanie Delaune. Safely composing security protocols. Research Report LSV-08-06, Laboratoire Spécification et Vérification, ENS Cachan, France, March 2008. 39 pages.

[Ra-7] Alain Finkel and Jérôme Leroux. Presburger functions are piecewise linear. Research Report LSV-08-08, Laboratoire Spécification et Vérification, ENS Cachan, France, March 2008. 9 pages.

[Ra-8] Elie Bursztein. Network administrator and intruder strategies. Research Report LSV-08-02, Laboratoire Spécification et Vérification, ENS Cachan, France, February 2008. 23 pages.

[Ra-9] Florent Jacquemard and Michaël Rusinowitch. Rewrite closure of Hedge-automata languages. Research Report LSV-07-31, Laboratoire Spécification et Vérification, ENS Cachan, France, October 2007. 22 pages.

[Ra-10] Benedikt Bollig. On the expressive power of 2-stack visibly pushdown automata. Research Report LSV-07-27, Laboratoire Spécification et Vérification, ENS Cachan, France, September 2007. 32 pages.

[Ra-11] Pierre Chambart. *Canaux fiables et non-fiables : frontières de la décidabilité*. Rapport de Master, Master Parisien de Recherche en Informatique, Paris, France, September 2007.

[Ra-12] Arnaud Da Costa. *Propriétés de jeux multi-agents*. Rapport de Master, Master de Logique Mathématique et Fondements de l'Informatique, Paris, France, September 2007.

[Ra-13] Camille Vacher. *Accessibilité inverse dans les automates d'arbres à mémoire d'ordre supérieur*. Rapport de Master, Master Parisien de Recherche en Informatique, Paris, France, September 2007.

[Ra-14] Jules Villard. *Logique spatiale pour le pi-calcul appliqué*. Rapport de Master, Master Parisien de Recherche en Informatique, Paris, France, September 2007.

[Ra-15] Riccardo Bresciani. The ZRTP protocol — security considerations. Research Report LSV-07-20, Laboratoire Spécification et Vérification, ENS Cachan, France, May 2007. 23 pages.

[Ra-16] Najla Chamseddine, Marie Duflot, Laurent Fribourg, and Claudine Picaronny. Determinate probabilistic timed automata as Markov chains with parametric costs. Research Report LSV-07-21, Laboratoire Spécification et Vérification, ENS Cachan, France, May 2007. 17 pages.

[Ra-17] Adel Bouhoula and Florent Jacquemard. Tree automata, implicit induction and explicit destructors for security protocol verification. Research Report LSV-07-10, Laboratoire Spécification et Vérification, ENS Cachan, France, February 2007. 21 pages.

[Ra-18] Patricia Bouyer, Serge Haddad, and Pierre-Alain Reynier. Undecidability results for timed automata with silent transitions. Research Report LSV-07-12, Laboratoire Spécification et Vérification, ENS Cachan, France, February 2007. 22 pages.

[Ra-19] Bogdan Księżopolski and Pascal Lafourcade. Attack and revison of an electronic auction protocol using OFMC. Technical Report 549, Department of Computer Science, ETH Zurich, Switzerland, February 2007. 13 pages.

[Ra-20] Jean Goubault-Larrecq. Believe it or not, GOI is a model of classical linear logic. Research Report LSV-07-03, Laboratoire Spécification et Vérification, ENS Cachan, France, January 2007. 18 pages.

[Ra-21] Pierre-Alain Reynier. Diagonal constraints handled efficiently in UPPAAL. Research Report LSV-07-02, Laboratoire Spécification et Vérification, ENS Cachan, France, January 2007. 4 pages.

[Ra-22] S. Akshay. *Formal Specification and Verification of Timed Communicating Systems*. Rapport de Master, Master Parisien de Recherche en Informatique, Paris, France, September 2006.

[Ra-23] Sergiu Bursuc. *Contraintes de déductibilité modulo Associativité-Commutativité*. Rapport de Master, Master Parisien de Recherche en Informatique, Paris, France, September 2006.

[Ra-24] Guyslain Naves. *Accessibilité dans les automates temporisés à deux horloges*. Rapport de Master, Master Parisien de Recherche en Informatique, Paris, France, September 2006.

[Ra-25] Rémi Brochenin. *Techniques d'automates pour raisonner sur la mémoire*. Rapport de Master, Master Recherche Informatique de Lyon — Informatique Fondamentale, Lyon, France, June 2006.

[Ra-26] Julien Olivain and Jean Goubault-Larrecq. Detecting subverted cryptographic protocols by entropy checking. Research Report LSV-06-13, Laboratoire Spécification et Vérification, ENS Cachan, France, June 2006. 19 pages.

[Ra-27] Arnaud Sangnier. Étude bibliographique sur la vérification de systèmes allouant dynamiquement de la mémoire. Technical report, EDF R&D, June 2006.

[Ra-28] Benedikt Bollig and Dietrich Kuske. Distributed Muller automata and logics. Research Report LSV-06-11, Laboratoire Spécification et Vérification, ENS Cachan, France, May 2006. 23 pages.

[Ra-29] Jean Goubault-Larrecq, Sławomir Lasota, and David Nowak. Logical relations for monadic types. Research Report cs.LO/0511006, Computing Research Repository, November 2005. 81 pages.

[Ra-30] Pascal Lafourcade, Denis Lugiez, and Ralf Treinen. Intruder deduction for the equational theory of exclusive-or with distributive encryption. Research Report LSV-05-19, Laboratoire Spécification et Vérification, ENS Cachan, France, October 2005. 39 pages.

[Ra-31] Simon Pinot. *Analyse de stabilité d'algorithme distribués probabilistes*. Rapport de Master, Master de Logique Mathématique et Fondements de l'Informatique, Paris, France, September 2005.

[Ra-32] Nathalie Sznajder. *Synthèse de contrôleur pour les systèmes distribués synchrones*. Rapport de Master, Master Parisien de Recherche en Informatique, Paris, France, September 2005.

[Ra-33] Adel Bouhoula and Florent Jacquemard. Automatic verification of sufficient completeness for specifications of complex data structures. Research Report LSV-05-17, Laboratoire Spécification et Vérification, ENS Cachan, France, August 2005. 14 pages.

[Ra-34] Adel Bouhoula and Florent Jacquemard. Automated induction for complex data structures. Research Report LSV-05-11, Laboratoire Spécification et Vérification, ENS Cachan, France, July 2005. 24 pages.

[Ra-35] Serge Haddad and Jean-François Pradat-Peyre. Efficient reductions for LTL formulae verification. Research Report 634, Centre De Recherche en Informatique du CNAM, Paris, France, 2004.

[Ra-36] Hubert Comon-Lundh and Stéphanie Delaune. The finite variant property: How to get rid of some algebraic properties. Research Report LSV-04-17, Laboratoire Spécification et Vérification, ENS Cachan, France, December 2004. 21 pages.

[Ra-37] Pascal Lafourcade, Denis Lugiez, and Ralf Treinen. Intruder deduction for AC-like equational theories with homomorphisms. Research Report LSV-04-16, Laboratoire Spécification et Vérification, ENS Cachan, France, November 2004. 69 pages.

[Ra-38] Serge Haddad and Patrice Moreaux. Sub-stochastic matric analysis and performance bounds. Research Report RAP-CReSTIC-1, Centre de Recherche en Sciences et Technologies de l'Information et de la Communication, Reims, France, October 2004.

[Ra-39] Fabrice Chevalier. *Détection d'erreurs dans les systèmes temporisés*. Rapport de DEA, DEA Algorithmique, Paris, France, September 2004. 59 pages.

[Ra-40] Véronique Cortier, Stéphanie Delaune, and Pascal Lafourcade. A survey of algebraic properties used in cryptographic protocols. Technical Report LSV-04-15, Laboratoire Spécification et Vérification, ENS Cachan, France, September 2004. 36 pages.

[Ra-41] Benjamin Ratti. *Automates d'arbre d'ordre deux*. Rapport de DEA, DEA Programmation, Paris, France, September 2004. 45 pages.

[Ra-42] Pierre-Alain Reynier. *Analyse en avant des automates temporisés*. Rapport de DEA, DEA Algorithmique, Paris, France, September 2004. 68 pages.

[Ra-43] Agnès Robin. *Aux frontières de la décidabilité...* Rapport de DEA, DEA Algorithmique, Paris, France, July 2004. 33 pages.

[Ra-44] Laurent Fribourg, Stéphane Messika, and Claudine Picaronny. Mixing time of the asymmetric simple exclusion problem on a ring with two particles. Research Report LSV-04-12, Laboratoire Spécification et Vérification, ENS Cachan, France, June 2004. 15 pages.

[Ra-45] Manuel Baclet and Rémy Chevallier. Using UPPAAL to verify an on-chip memory. Research Report LSV-04-11, Laboratoire Spécification et Vérification, ENS Cachan, France, May 2004. 12 pages.

[Ra-46] Manuel Baclet, Renaud Pacalet, and Antoine Petit. Register transfer level simulation. Research Report LSV-04-10, Laboratoire Spécification et Vérification, ENS Cachan, France, May 2004. 15 pages.

[Ra-47] Stéphanie Delaune and Florent Jacquemard. A theory of guessing attacks and its complexity. Research Report LSV-04-1, Laboratoire Spécification et Vérification, ENS Cachan, France, January 2004. 25 pages.

[Ra-48] Kumar N. Verma and Jean Goubault-Larrecq. Karp-Miller trees for a branching extension of VASS. Research Report LSV-04-3, Laboratoire Spécification et Vérification, ENS Cachan, France, January 2004. 21 pages.

[Ra-49⋆] Dietmar Berwanger and Laurent Doyen. On the power of imperfect information. Technical Report MTC-2008-006, Lab. Models and Theory of Computation, EPFL, Lausanne, Switzerland, July 2008.

[Ra-50⋆] Étienne André. *Handling Theories in Logic Functors for Recomposing Description Logics*. Rapport de Master, Master de Recherche en Informatique, Rennes, France, June 2007.

[Ra-51⋆] Diego Figueira. *Bisimulations on Neighbourhood Semantics*. Rapport de Master, Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Argentina, December 2006.

[Ra-52⋆] Sylvain Schmitz. Modular syntax demands verification. Technical Report I3S/RR-2006-32-FR, Laboratoire I3S, Université de Nice-Sophia Antipolis, France, October 2006.

[Ra-53⋆] Antoine Mercier. *La logique linéaire multiplicative cyclique est NP-complète (d'apr?s Mati Pentus)*. Rapport de Master, Master de Logique Mathématique et Fondements de l'Informatique, Paris, France, July 2006.

[Ra-54⋆] Florent Bouchy. *Bibliothèque de méthodes pour la classification*. Rapport de Master, Master Recherche Informatique, Tours, France, September 2005.

[Ra-55⋆] Arnaud Sangnier. *Vers la vérification de réseaux de Petri colorés temporels*. Rapport de Master, Master Systèmes et Applications Répartis, Paris, France, September 2005.

[Ra-56⋆] Delphine Longuet. *Une théorie du raffinement orientée propriétés pour les automates communicants*. Rapport de DEA, lami, June 2005.

[Ra-57⋆] Rohit Chadha, Steve Kremer, and Andre Scedrov. Analysis of multi-party contract signing. Technical Report 516, Université Libre de Bruxelles, Belgium, 2004.

[Ra-58⋆] Myrto Arapinis. *Développement d'un langage pour la représentation des tactiques de recherche de preuves fondées sur les motifs*. Rapport de DEA, DEA Programmation, Paris, France, September 2004.

Rapports de contrats

[Rc-1] Benedikt Bollig, Patricia Bouyer, Franck Cassez, Thomas Chatain, Paul Gastin, Serge Haddad, and Claude Jard. Model for distributed timed systems. Deliverable 3.1, projet DOTS (ANR-06-SETI-003), September 2008.

[Rc-2] Franck Cassez, François Laroussinie, Didier Lime, and Nicolas Markey. Quantitative objectives in timed games. Deliverable 1.1, projet DOTS (ANR-06-SETI-003), September 2008.

[Rc-3] François Laroussinie *et al.* Projet dots (anr-06-seti-003) : Rapport à 18 mois, September 2008. 5 pages.

[Rc-4] François Laroussinie *et al.* Projet dots (anr-06-seti-003) : Rapport à 12 mois, March 2008. 6 pages.

[Rc-5] Pascal Lafourcade. Rapport final d'activité à 11 mois, contrat CNRS/DGA référence : 06 60 019 00 470 75 01 « Utilisation et exploitation des théories équationnelles dans l'analyse des protocoles cryptographiques ». Contract report, DGA, October 2007. 6 pages.

[Rc-6] ARC ProNoBis. Pronobis: Probability and nondeterminism, bisimulations and security – Rapport final, October 2007.

[Rc-7] LIAFA, CRIL Technology, EDF R&D, LSV, and Verimag. Projet RNTL Averiles – fourniture f1.1 : Modèles, September 2007. 6 pages.

[Rc-8] LIAFA, CRIL Technology, EDF R&D, LSV, and Verimag. Projet RNTL Averiles – fourniture f1.2 : Extraction de modèles, September 2007. 19 pages.

[Rc-9] LIAFA, CRIL Technology, EDF R&D, LSV, and Verimag. Projet RNTL Averiles – fourniture f1.3 : Algorithmes de vérification, September 2007. 19 pages.

[Rc-10] LIAFA, CRIL Technology, EDF R&D, LSV, and Verimag. Rapport à mi-parcours du projet RNTL Averiles (analyse et vérification de logiciels embarqués avec structures de mémoire dynamique, September 2007. 4 pages.

[Rc-11] LIAFA, LSV, and Verimag. Projet RNTL Averiles – fourniture f1.4 : Prototypes d'outil, September 2007. 3 pages.

[Rc-12] Alain Ourghanlian, Marius Bozga, Adam Roglewicz, and Arnaud Sangnier. Projet RNTL Averiles – fourniture f1.6 : Expérimentation, September 2007. 16 pages.

[Rc-13] François Laroussinie *et al.* Projet dots (anr-06-seti-003) : Rapport à 6 mois, August 2007. 7 pages.

[Rc-14] Stéphanie Delaune and Francis Klay. Synthèse des expérimentations. Technical Report 10, projet RNTL PROUVÉ, May 2007. 10 pages.

[Rc-15] Pascal Lafourcade. Rapport d'activités à 6 mois, contrat CNRS/DGA référence : 06 60 019 00 470 75 01 « Utilisation et exploitation des théories équationnelles dans l'analyse des protocoles cryptographiques ». Contract report, DGA, April 2007. 5 pages.

[Rc-16] Pascal Lafourcade. Rapport d'activités à 3 mois, contrat CNRS/DGA référence : 06 60 019 00 470 75 01 « Utilisation et exploitation des théories équationnelles dans l'analyse des protocoles cryptographiques ». Contract report, DGA, January 2007. 3 pages.

[Rc-17] Patricia Bouyer *et al.* ACI Sécurité Informatique CORTOS — rapport final, November 2006. 17 pages.

[Rc-18] Francis Klay, Liana Bozga, Yassine Lakhnech, Laurent Mazaré, Stéphanie Delaune, and Steve Kremer. Retour d'expérience sur la validation du vote électronique. Technical Report 7, projet RNTL PROUVÉ, November 2006. 47 pages.

[Rc-19] Philippe Schnoebelen, Ahmed Bouajjani, and Grégoire Sutre. ACI Sécurité Informatique PERSÉE — rapport final, November 2006. 12 pages.

[Rc-20] Paul Gastin *et al.* ACI Sécurité Informatique VERSYDIS — rapport final, October 2006. 10 pages.

[Rc-21] Laurent Fribourg, Claudine Picaronny, and Simon Pinot. Manipulation de backoff dans CSMA/CA. Contract Report (Lot 4.2 fourniture 6), projet RNTL Averroes, March 2006. 20 pages.

[Rc-22] Arnaud Sangnier. Fourniture au CEA d'exemples d'extraction de modèles dans le cadre de la collaboration EDF R&D–LSV. Contract report, contrat 4300038040 LSV/ EDF R&D, March 2006. 11 pages.

[Rc-23] Steve Kremer, Yassine Lakhnech, and Ralf Treinen. The Prouvé manual: Specifications, semantics, and logics. Technical Report 7, projet RNTL PROUVÉ, December 2005. 49 pages.

[Rc-24] Stéphanie Delaune, Francis Klay, and Steve Kremer. Spécification du protocole de vote électronique. Technical Report 6, projet RNTL PROUVÉ, November 2005. 19 pages.

[Rc-25] Philippe Schnoebelen *et al.* ACI Sécurité Informatique PERSÉE — rapport à mi-parcours, November 2005. 8 pages.

[Rc-26] Sébastien Bardin, Frédéric Herbreteau, Mihaela Sighireanu, Grégoire Sutre, and Aymeric Vincent. Intégration des outils PERSÉE (proposition d'architecture). Délivrable 3.1 — Partie 1 du Projet PERSÉE de l'ACI Sécurité Informatique, June 2005. 35 pages.

[Rc-27] Patricia Bouyer *et al.* ACI Sécurité Informatique CORTOS « Control and Observation of Real-Time Open Systems » — rapport à mi-parcours, April 2005. 6 pages.

[Rc-28] Liana Bozga, Stéphanie Delaune, Francis Klay, and Laurent Vigneron. Retour d'expérience sur la validation du porte-monnaie électronique. Technical Report 5, projet RNTL PROUVÉ, March 2005. 29 pages.

[Rc-29] Véronique Cortier, Francis Klay, Yassine Lakhnech, Bertrand Tavernier, and Ralf Treinen. Projet RNTL PROUVÉ — fiche d'étape 2004. Technical report, projet RNTL PROUVÉ, March 2005. 6 pages.

[Rc-30] Vincent Bernat, Hubert Comon-Lundh, Véronique Cortier, Stéphanie Delaune, Florent Jacquemard, Pascal Lafourcade, Yassine Lakhnech, and Laurent Mazaré. Sufficient conditions on properties for an automated verification: theoretical report on the verification of protocols for an extended model of the intruder. Technical Report 4, projet RNTL PROUVÉ, December 2004. 33 pages.

[Rc-31] Ralf Treinen. The PROUVÉ specification language. Technical Report 3, Projet RNTL PROUVÉ, August 2004. 10 pages.

[Rc-32] Liana Bozga, Stéphanie Delaune, Francis Klay, and Ralf Treinen. Spécification du protocole de porte-monnaie électronique. Technical Report 1, projet RNTL PROUVÉ, June 2004. 12 pages.

[Rc-33] Véronique Cortier, Stéphanie Delaune, and Pascal Lafourcade. A survey of algebraic properties used in cryptographic protocols. Technical Report 2, projet RNTL PROUVÉ, June 2004. 19 pages.

[Rc-34] Marie Duflot, Laurent Fribourg, Thomas Hérault, Richard Lassaigne, Frédéric Magniette, Stéphane Messika, Sylvain Peyronnet, and Claudine Picaronny. Probabilistic model checking of the CSMA/CD protocol using PRISM and APMC. Contract Report (Lot 4.2 fourniture 2), projet RNTL Averroes, June 2004. 22 pages.

[Rc-35] Laurent Fribourg, Stéphane Messika, and Claudine Picaronny. Parametric and probabilistic verification of Nicollin-Sifakis-Yovine's model of the CSMA/CD protocol. Contract Report (Lot 4.2 fourniture 3), projet RNTL Averroes, June 2004. 17 pages.

[Rc-36] Manuel Baclet and Rémy Chevallier. Using UPPAAL to verify an on-chip memory. Contract Report (Work Package 3.2 Fourniture 1), projet T126 MEDEA+ Blueberries, May 2004. 12 pages.

[Rc-37] Marie Duflot, Stéphane Messika, and Claudine Picaronny. Vérification du protocole CSMA/Cd sous PRISM. Contract Report (Lot 4.2 fourniture 1), projet RNTL Averroes, January 2004. 12 pages.

Réalisations informatiques

[Lo-1] Elie Bursztein. Netanalyzer v0.7.5, January 2008. Written in C and Perl (about 25000 lines).

[Lo-2] Elie Bursztein. Netqi v1rc1. Available at `http://www.netqi.org/`, December 2007. Written in C and Java (about 10000 lines).

[Lo-3] Paul Gastin and Denis Oddoux. Ltl2ba v1.1, August 2007. Written in C++ (about 4000 lines).

[Lo-4] Paul Gastin. GasTEX: Graphs and automata simplified in TEX (v2.8), November 2006. Written in TEX(about 2000 lines).

[Lo-5] Nathalie Bertrand. Sumo – reachability analysis for lossy channels, February 2006. See [CI-105] for a description. Written in OCaml (3000 lines).

[Lo-6] Ralf Treinen. RTALOOP: The RTA list of open problems. Web site at `http://www.lsv.ens-cachan.fr/rtaloop/`, started 1997, 2004. Size as of July 2004: 100 problems, 90 pages, 432 references.

[Lo-7] Julien Olivain. Net-entropy v1.0: An entropy checker for ciphered network connections, September 2004.

[Lo-8] Thomas Hugel. SSP: Stochastic shortest paths, July 2004. Written in Caml (about 500 lines).

[Lo-9] Julien Olivain. EVTGEN v1.0: A programmable generic generator of event sequences, July 2004. Written in C (about 5000 lines).

[Lo-10] Sébastien Bardin, Christophe Darlot, Alain Finkel, Jérôme Leroux, and Laurent Van Begin. FAST v1.5: Fast Acceleration of Symbolic Transition systems. Available at `http://www.lsv.ens-cachan.fr/fast/`, June 2004.

Liste des acronymes utilisés

ABR ..... Alternating Bit Rate (un protocole télécom)
AC ..... Allocation Couplée (i.e. incluant un monitorat)
ACI ..... Action Concertée Incitative
ACMO ..... Agent Chargé de la Mise en Oeuvre (de la politique d'hygiène et de sécurité)
AERES ..... Agence d'Évaluation de la Recherche et de l'Enseignement Supérieur
AFCRST ..... Association Franco-Chinoise pour la Recherche Scientifique et Technique
ARA ..... Action de Recherche Amont (de l'ANR)
ARC ..... Action de Recherche Collaborative (de l'INRIA)
AS ..... Action Spécifique (du CNRS)
ASTI ..... Fédération des Associations Françaises des Sciences et Technologies de l'Information
ATL ..... Alternating-time Temporal Logic
BDD ..... Boolean Decision Diagram (une structure de données)
BTL ..... Branching-time Temporal Logic
BVASS ..... Branching Vector Addition System with States
CEGAR ..... CounterExample-Guided Abstraction-Refinement
CIFRE ..... Conventions Industrielles de Formation par la Recherche
CLTL ..... Constraint LTL
CPGE ..... Classes Préparatoires aux Grandes Écoles
CQDD ..... Constrained Queue-content Decision Diagram
CSMA/CD ..... Carrier Sense Multiple Access with Collision Detection
CSMA/CA ..... Carrier Sense Multiple Access with Collision Avoidance
CTL ..... Computation Tree Logic
DBM ..... Difference-Bounds Matrix
DCSSI ..... Direction Centrale de la Sécurité des Systèmes d'Information
DGA ..... Direction Générale pour l'Armement
DLP ..... Dynamic Logic of Permission
DPTA ..... Determinate Probabilistic Timed Automaton
ECTL ..... Extended Computation Tree Logic
EDSP ..... École Doctorale Sciences Pratiques à Cachan
EPI ..... Équipe-Projet INRIA
ETL ..... Extended Temporal Logic
FIFO ..... First In, First Out
GIE ..... Groupement d'Intérêt Économique
GPL ..... GNU General Public License
HDR ..... Habilitation à Diriger des Recherches
IEEE ..... Institute of Electrical and Electronics Engineers
IFIP ..... International Federation for Information Processing
IP ..... Internet Protocol

LCEN ..... Loi du 21 juin 2004 pour la Confiance dans l'Économie Numérique
LCS ..... Lossy Channel Systems
LOLF ..... Loi Organique du 1er août 2001 relative aux Lois de Finances
LTL ..... Linear-time Temporal Logic
LURPA ..... Laboratoire Universitaire de Recherche en Production Automatisée
MITL ..... Metric Interval Temporal Logic
MPRI ..... Master Parisien de Recherche en Informatique
MSO ..... Monadic Second-Order Logic
MTL ..... Metric Temporal Logic
NAT ..... Network Address Translation
NDD ..... Number Decision Diagram
PCRD ..... Programme-Cadre de Recherche et de Développement (de l'Union Européenne)
MSO ..... Monadic Second-Order Logic
PDL ..... Propositional Dynamic Logic
PDM ..... Processus de Décision Markovien
PGM ..... Pragmatic General Multicast (un protocole de communication)
PID ..... à action Proportionnelle – Intégrale – Dérivée
PI ..... Peripheral Interconnect
PRAG ..... Professeur agrégé
REX ..... Réseau d'Excellence
RNRT ..... Réseau National de Recherche en Télécommunications
RNTL ..... Réseau National des Technologies Logicielles
RSA ..... Rivest Shamir Adleman (les inventeurs d'un algorithme de chiffrement)
SATIE .. Laboratoire Systèmes et Applications des Technologies de l'Information et de l'Énergie
SOC ..... System On Chip
SPIN ..... Simple Promela INterpreter (un model-checker)
TATL ..... Timed Alternating-time Temporal Logic
TCTL ..... Timed Computation Tree Logic
TPTL ..... Timed Propositional Temporal Logic
UML ..... Unified Modeling Language
VASS ..... Vector Addition System with States
VCI ..... Virtual Component Interface
VHDL ..... Very High Speed Integrated Circuit (VHSIC) Hardware Description Language
VSMT ..... Vérification de Systèmes Multi-tâches Temps réel
WCTL ..... Weighted Computation Tree Logic
WMTL ..... Weighted Metric Temporal Logic
WSTS ..... Well-Structured Transition System
XML ..... Extensible Markup Language
XSLT ..... Extensible Stylesheet Language Transformations

Organigramme de l'unité

Le schéma ci-dessous représente l'organigramme du laboratoire au 1er janvier 2006.