

# Action Concertée Incitative *Sécurité Informatique*

## Projet VERSYDIS

*Formalismes non séquentiels pour la vérification et la synthèse de systèmes distribués*

### Rapport final

30 octobre 2006

## 1 Équipes impliquées et participants

L'ACI VERSYDIS impliquait initialement deux laboratoires, le LIAFA, Université Paris 7, CNRS UMR 7089, et le LaBRI, Université Bordeaux 1, CNRS UMR 5800. La configuration au début de l'ACI était la suivante :

LIAFA	LaBRI
P. Gastin	J. Bernet
H. Gimbert	D. Janin
B. Genest	O. Ly
F. Horn	I. Walukiewicz
B. Lerman	P. Weil
A. Muscholl	
M. Zeitoun	
W. Zielonka	

Plusieurs changements d'affectation se sont produits durant le projet, qui implique à présent notamment le LSV, ENS Cachan, CNRS UMR 8643 (2 membres dont le porteur du projet). Les principaux changements sont les suivants :

- mutation de P. Gastin au LSV, ENS Cachan et nouvelle doctorante, N. Sznajder, sous sa direction ;
- mutation de A. Muscholl au LaBRI ;
- recrutement de B. Genest à l'IRISA, Rennes (CR CNRS) ;
- post-doc de H. Gimbert au LIX, École Polytechnique ;
- recrutement de M. Zeitoun au LaBRI.

La configuration actuelle est la suivante :

LIAFA	LaBRI	LSV	IRISA	LIX
F. Horn W. Zielonka	J. Bernet D. Janin O. Ly A. Muscholl I. Walukiewicz P. Weil M. Zeitoun	P. Gastin N. Sznajder	B. Genest	H. Gimbert

Cette mobilité des chercheurs, notamment du LIAFA vers le LaBRI, a été favorisée par les nombreuses visites et les collaborations fréquentes rendues possibles par l'ACI. Nous avons exposé nos idées au cours de plusieurs réunions impliquant l'ensemble des membres, donc une commune avec les ACI Cortos et Persée.

Deux thèses [2, 5] et deux habilitations à diriger des recherches [4, 6] ont été soutenues. Par ailleurs, la thèse de J. Bernet [1] sera soutenue le 8 novembre 2006. La thèse de H. Gimbert [3] est en cours d'évaluation.

## 2 La thématique

Le projet concerne la sécurité des systèmes informatiques, et tout particulièrement celle des systèmes distribués. Il propose de développer des concepts et des outils pour concevoir des systèmes sûrs et pour vérifier que le comportement de ces systèmes est conforme aux spécifications qui leur sont imposées. Plus précisément, ce projet s'inscrit dans le cadre de la vérification de modèle (model checking) et de la synthèse.

La méthode standard pour vérifier et synthétiser des systèmes est basée sur une représentation par automate fini, aussi bien du système que de la propriété. Ceci permet de réduire chacun de ces deux problèmes au test du vide pour les automates. Cette méthodologie peut aussi être formulée en termes de jeux à deux joueurs, ce qui a l'avantage d'offrir un cadre plus général et plus flexible. Une notion centrale pour vérifier les systèmes réactifs est celle de jeux de durée infinie, à deux joueurs, avec conditions de gain de parité.

Cette approche, bien que standard jusqu'à présent, a ses limites. La principale est que les systèmes réels ne peuvent que très rarement être représentés par des automates finis de taille raisonnable. En particulier, lorsqu'un système est composé de plusieurs modules travaillant de façon concurrente, une représentation par automate fini du système croît de façon exponentielle par rapport au nombre de composants. Pire, lorsque la communication entre composants est asynchrone, les canaux de communication du système ont une capacité non bornée, ce qui rend impossible la représentation fidèle du système par automate fini. Dans le cas de la synthèse, le problème est encore plus manifeste : synthétiser un système centralisé est certes un premier pas significatif, mais encore très éloigné de l'objectif de synthétiser un système distribué sur une architecture de communications fixée.

Le projet VERSYDIS proposait de travailler sur des extensions des méthodes basées sur les automates classiques, pour traiter et exploiter l'aspect distribué des systèmes. Nous avons travaillé sur les modèles ayant une sémantique distribuée, et sur des formalismes adaptés à la description de leur propriétés : nous avons proposé des algorithmes de vérification et nous avons étudié les questions sous-jacentes de décidabilité et de complexité. Nous avons étudié le problème de la synthèse distribuée, notamment via de nouveaux modèles de jeux. Ces lignes directrices principales nous ont également conduits à étudier des problèmes connexes, comme par exemple des extensions aux graphes de la notion de reconnaissabilité, des approches algébriques pour manipuler les langages d'arbres, des extensions des modèles de jeux avec des conditions de gain qualitatives, des modèles de jeux probabilistes, ou la vérification de schémas de programmes récursifs.

Nous présentons ci-dessous une liste plus détaillée des contributions du projet. Dans ce rapport synthétique, il n'est pas possible d'entrer en profondeur dans les résultats obtenus, qui sont présentés dans environ 60 publications en conférences et revues d'audience internationale. Pour un aperçu plus précis de certains aspects considérés dans ce projet, nous renvoyons le lecteur aux 2 mémoires d'habilitation et aux 4 mémoires de thèse de doctorat qui ont été écrits dans le cadre de ce projet.

## 2.1 Synthèse et vérification de systèmes communicants

La thèse [2] constitue un ouvrage assez complet concernant l’expressivité, la décidabilité et la complexité de la vérification des systèmes communicants spécifiés sous de multiples formalismes (MSC-graphes, langages réguliers, automates communicants ou logiques du second ordre MSO). Un exposé invité à été organisé sur le sujet pendant la conférence ACSD [42].

Dans [15], nous avons expliqué comment des propriétés pouvaient être décidées pour des systèmes concurrents communiquant par canaux non bornés, alors que seule la décidabilité des protocoles finis de communication était jusqu’alors connue. Pour cela, nous avons montré comment abstraire automatiquement cet ensemble infini par des exécutions représentatives bornées, décomposées en « atomes ». De plus, nous avons considéré des systèmes plus simples, à choix local, pour lesquels les algorithmes sont aussi performants que pour les automates. Enfin, nous avons montré que ces systèmes à choix locaux étaient toujours implantables sans blocage, alors que des protocoles plus complexes (localement coopératifs) sont toujours implantables, mais avec blocage.

Dans [39], nous avons défini un encodage très puissant par trace de Mazurkiewicz<sup>1</sup>. Il nous a permis d’étendre [15] à tous les protocoles ayant des représentants bornés (mais pas forcément décomposables en atomes). Nous avons surtout nettement étendu le théorème de Kleene en résolvant l’expressivité de ce type de systèmes concurrents en terme d’automates communicants et de logique du second ordre.

Dans [37], on a considéré une heuristique pour la synthèse en pratique des systèmes concurrents définis de manière globale. Plus précisément, un algorithme est donné pour tester si un protocole de communication est équivalent un système concurrent à choix local. Comme [15] montre que ces systèmes peuvent toujours être synthétisés sans blocage, nous obtenons la synthèse sans blocage des systèmes capturés par le test. Bien sûr, il se peut qu’un système puisse être synthétisé sans blocage, mais non capturé par notre test.

Comme on l’a dit précédemment, synthétiser automatiquement des systèmes concurrents définis de manière globale est une tâche ardue. Les méthodes connues de synthèse des systèmes concurrents utilisent souvent le théorème de Zielonka pour générer l’information supplémentaire à utiliser dans les messages, mais la méthode est malheureusement trop coûteuse en pratique. Une approche plus pragmatique est de permettre également d’ajouter dans certains canaux de communication des messages qui n’étaient pas spécifiés, ce qui facilite la synthèse et réduit la complexité [38]. De plus, l’étude minutieuse de la complexité de l’algorithme de Zielonka a permis d’améliorer sa complexité d’une exponentielle, rendant possible son utilisation en pratique [43].

Nous avons enfin considéré le problème du test de systèmes asynchrones [25]. Nous avons proposé deux notions d’équivalence par test. Nous les avons comparées avec celles existantes, et étudié leurs propriétés, établissant plusieurs résultats de décidabilité et d’indécidabilité.

## 2.2 Synthèse distribuée

Nous avons introduit le modèle des jeux distribués pour mieux comprendre le problème de la synthèse distribuée [53, 63]. Nous avons montré que ce modèle englobe les cas de décidabilité de la synthèse distribuée déjà connus (Rudie et Wonham, Pnueli et Rosner, Madhusudan et Thiagarajan). Nous avons proposé des opérations de simplification et de résolution de tels jeux. Plus récemment, on a travaillé sur l’utilisation de techniques de la théorie des automates sur les arbres qui peuvent être réutilisées dans le contexte de jeux distribués [23, 24]. Plusieurs de ces résultats font partie de ceux obtenus par Julien Bernet dans sa thèse [1].

---

<sup>1</sup>Ces traces représentent par des ordres partiels particuliers les exécutions des systèmes distribués.

De nombreux travaux antérieurs (Pnueli et Rosner, Madhusudan et Thiagarajan, Kupferman et Vardi) ont en effet montré que la synthèse de contrôleur distribué est indécidable si les contrôleurs locaux ne prennent leurs décisions que sur la base d’observations, sans communication additionnelle. Nous avons montré [34, 33] que donner aux agents la possibilité de surcharger leurs communications par une abstraction finie de leur vision du système permet de décider une large classe de systèmes distribués. Cette étude a été menée sur le modèle des automates asynchrones. Dans le cadre synchrone, nous avons récemment [36] introduit le concept d’architecture uniformément bien connectée, et montré que le problème de synthèse distribuée pour ces architectures est décidable pour les spécifications qui contraignent uniquement les entrées et les sorties du systèmes, et pas ses canaux internes. Nous avons enfin affiné notre compréhension des cas où la synthèse est décidable, dans le cadre des jeux distribués [23].

### 2.3 Au delà des jeux de parité

Comme précisé plus haut, les interactions entre un système réactif et son environnement sont souvent modélisées comme un jeu à deux joueurs. Dans la plupart de cas on utilise les jeux avec la condition de gain « parité » ou ses variantes. Pourtant, il y a d’autres conditions qui donnent des informations pertinentes sur le système et son comportement. Mais à chaque fois qu’on examine une nouvelle condition de gain se pose la question de savoir s’il existe des stratégies optimales et surtout, s’il existe des stratégies optimales *sans mémoire*, qui sont les seules que nous pouvons effectivement implanter. Le premier but de nos recherches était donc de caractériser les jeux sur les systèmes à nombre fini d’états qui admettent de telles stratégies. Ce problème a été complètement résolu [62, 47].

Nous avons aussi considéré le même problème pour des jeux sur des graphes infinis. Très souvent, on s’intéresse à ce problème sur les graphes d’exécution d’automates à pile. Dans ce contexte, de nouvelles conditions de gain deviennent pertinentes. Dans [44, 45], nous avons présenté des algorithmes optimaux pour résoudre les jeux avec conditions exprimant des propriétés sur la hauteur de la pile. Nous avons également considéré une version assez générale du problème où il n’y a pas de restrictions sur les graphes, et où la condition de gain est une condition « de Muller » sur un nombre infini de couleurs [16]. Nous avons montré que la version infinie de la condition de parité est la seule condition admettant des stratégies optimales sans mémoire.

Un autre sujet traité concerne la pertinence même des jeux infinis, en particulier du jeu de parité. En réalité chaque système a une durée de vie finie, alors les jeux infinis, modélisant les interactions non bornées en temps, ne sont qu’une idéalisation de la réalité. Nous avons montré [48] que les jeux de parité sont en fait une limite de jeux dont le paiement diminue avec le temps, ce qui est équivalent aux jeux probabilistes qui s’arrêtent avec une probabilité 1. Intuitivement, les jeux de parité peuvent ainsi être considérés comme une approximation des jeux de durée finie, mais inconnue (ce qui renforce l’intérêt de les étudier).

### 2.4 Formalismes de spécification : aspect distribué et complexité

Nous avons enfin résolu positivement un problème ouvert depuis fort longtemps sur les traces de Mazurkiewicz : la logique temporelle locale basée sur les deux modalités naturelles *exists-next* et *until* a le même pouvoir d’expression que la logique du premier ordre. Après un premier résultat restreint aux systèmes série-parallèle [8] nous avons généralisé cette équivalence à tous les systèmes distribués que l’on peut décrire par des traces de Mazurkiewicz [30, 9, 10].

Dans [41], nous avons défini l’équivalent de la logique temporelle linéaire pour spécifier des propriétés globales sur les protocoles de communication. Cette logique est facilement utilisable car visuelle, et nous avons montré la décidabilité de son model-checking, ainsi que la PSPACE-complétude d’un fragment assez expressif de cette logique.

Dans [40], nous avons considéré des systèmes concurrents décrits de manière séquentielle. Cependant, ils doivent suivre des spécifications décrites de façon concurrente. Ainsi, nous avons défini une extension de la logique temporelle linéaire LTL avec un opérateur parlant des exécutions équivalentes, du point de vue de la concurrence. Par exemple, nous pouvons expliquer comment une banque avec plusieurs branches peut s'assurer que si un client paraît avoir eu un solde négatif, ça n'est pas un artefact de la séquentialisation. Nous avons obtenu une complexité EXPSPACE pour cette logique, ainsi qu'une complexité PSPACE pour un fragment qui a aussi le même pouvoir d'expression que la logique du premier ordre. Enfin, nous avons obtenu une complexité polynomiale en temps pour le model-checking d'un mot dans de nombreux sous-cas.

Nous avons établi plusieurs résultats pour la complexité de formalismes de vérification. Dans [32], nous avons obtenu des théorèmes de complexité très génériques pour la vérification de propriétés exprimées en logique temporelle sur les traces. Ils permettent de retrouver la plupart des bornes de dans la littérature obtenues précédemment pour de telles logiques. Dans [17], nous sommes revenus à un fragment particulier de la logique temporelle linéaire LTL sur les mots. Nous avons démontré que le problème de satisfaisabilité pour ce fragment est dans NP [18], alors que pour la logique complète le problème est PSPACE-difficile. Ce résultat donne l'espoir de trouver une logique dite globale sur les traces de Mazurkiewicz ayant une complexité raisonnable pour son problème de satisfaisabilité.

Nous avons travaillé sur des problèmes ouverts de la théorie des langages d'arbres. Comme une stratégie se représente par un arbre, de telles questions ont naturellement émergé comme problèmes centraux dans le cadre de l'ACI. Le premier problème concerne la décidabilité de classes importantes de langages reconnaissables d'arbres finis et infinis. Une question ouverte depuis longtemps est celle de la décidabilité des langages définissables au premier ordre. Nous avons proposé une nouvelle structure algébrique pour la reconnaissance de langages d'arbres, et établi un théorème général de classification dans ce cadre [13]. Malheureusement, cette classification n'implique pas la décidabilité. Dans [27] on a montré la décidabilité de trois sous-classes de langages définissables au premier ordre. La deuxième question majeure dans le domaine est celle de la décidabilité de la hiérarchie des langages définis par alternances de points fixes. Nous avons réussi à résoudre le problème dans le cas particulier où le langage donné est déterministe [57].

Nous avons approfondi la notion algébrique de reconnaissabilité pour les langages de graphes [20], une notion qui dépend des opérations sur l'ensemble de graphes prises en considération. Nous avons comparé cette notion avec les propriétés de définissabilité logique. Les résultats principaux concernent les signatures dites HR et VR, et la signature modulaire. Nous avons étudié aussi le pouvoir d'expression des fragments de la logique du second ordre et des automates sur les graphes et nous avons obtenu une caractérisation dans des cas particuliers [29, 25].

## 2.5 Autres contributions

Nous avons étudié le problème de vérification pour les programmes récursifs d'ordre supérieur. Nous avons considéré l'Algol idéalisé, un langage théorique faisant la synthèse entre caractéristiques impératives et fonctionnelles. Un progrès récent dans la sémantique de ce langage permet de construire des modèles de programmes très précis. En les utilisant, nous avons complètement résolu le problème de décider l'équivalence de deux programmes pour ce langage [54, 55]. Nous avons aussi étudié les schémas de programmes récursifs, qui sont une forme abstraite de programmes récursifs. Dans les schémas d'ordre supérieurs, on peut avoir des fonctions prenant comme arguments des fonctions d'ordre supérieur. Nous avons montré la décidabilité de la vérification de propriétés du mu-calcul pour des schémas de programmes récursifs du second ordre [50].

Nous avons fait quelques contributions à la théorie de la vérification des systèmes temps-

réel. L'un des problèmes principaux dans ce domaine est l'absence d'un modèle d'automate tout à fait satisfaisant. Nous avons proposé le modèle d'automate alternant temporisé. Par définition, ces automates sont clos par opérations Booléennes. Nous avons montré que pour ce modèle, le test du vide est décidable uniquement pour les automates à une horloge. Le pouvoir expressif des automates alternants à une horloge n'est pas comparable à celui des automates classiques d'Alur and Dill. Dans [21, 22], nous avons étudié un autre modèle prometteur : celui des automates signal-événement. Nous avons montré un certain nombre de propriétés de clôture pour ces automates.

La méthode de dépliage de réseaux de Petri a été développée d'abord pour la vérification des réseaux dits sains, puis étendus à une classe un peu plus large de réseaux. Nous avons développé [64] une extension de la notion de dépliage aux réseaux généraux. Ce travail fait apparaître une notion de dépliage dit « fidèle », qui se confond avec le dépliage sain dans le cas des réseaux sains, mais qui a de meilleures propriétés que les dépliages sains dans le cas général. Les perspectives ouvertes concernent l'utilisation de cette notion de dépliage dans des algorithmes de vérification puis dans des outils informatiques.

Enfin, dans [35], nous avons montré comment améliorer le model-checker SPIN afin qu'il produise des contre-exemples les plus courts possible. Nous avons d'une part accéléré l'algorithme de recherche d'un contre-exemple, puis donné un algorithme fournissant une suite de contre-exemples de longueur décroissante, jusqu'à un contre-exemple de taille minimale. L'algorithme a été implanté dans SPIN et les tests ont produit des résultats très satisfaisants.

### 3 Publications

#### Thèses et habilitations

1. J. Bernet. *Jeux distribués*. Doctoral thesis, LaBRI, Université Bordeaux 1, Soutenance le 8 novembre 2006.
2. B. Genest. *The odyssey of MSC-graphs*. Doctoral thesis, LIAFA, University Paris 7, 2005.
3. H. Gimbert. *Jeux positionnels*. Doctoral thesis, LIAFA, University Paris 7, En cours d'évaluation.
4. D. Janin. *Contribution aux fondements des méthodes formelles: jeux, logique et automates*. Habilitation thesis, LaBRI, University Bordeaux 1, Nov. 2005.
5. B. Lerman. *Vérification et spécification des systèmes distribués*. Doctoral thesis, LIAFA, University Paris 7, Nov. 2005.
6. M. Zeitoun. *Concurrence et automates*. Habilitation thesis, LIAFA, University Paris 7, Dec. 2005.

#### Publications en journal (d'audience internationale, avec comité de sélection)

7. B. Courcelle and P. Weil. The recognizability of sets of graphs is a robust property. *Theoretical Comput. Sci.*, 342:173–228, 2005.
8. V. Diekert and P. Gastin. Local temporal logic is expressively complete for cograph dependence alphabets. *Inform. Comput.*, 195:30–52, 2004.
9. V. Diekert and P. Gastin. From local to global temporal logics over Mazurkiewicz traces.

*Theoretical Comput. Sci.*, 356(1-2):126–135, 2006. In honour of Professor Christian Choffrut on the occasion of his 60th birthday.

10. V. Diekert and P. Gastin. Pure future local temporal logics are expressively complete for Mazurkiewicz traces. *Inform. Comput.*, 204:1597–1619, 2006.
11. M. Droste and P. Gastin. On aperiodic and star-free formal power series in partially commuting variables. *ACM Trans. Comput. Syst.*, 2006. to appear. Also available as Research Report LSV-05-12, Laboratoire Spécification et Vérification, ENS Cachan, France, July 2005.
12. M. Droste and P. Gastin. Weighted automata and weighted logics. *Theoretical Comput. Sci.*, 2007. Special issue of ICALP’05. To appear. Also available as Research Report LSV-05-13, Laboratoire Spécification et Vérification, ENS Cachan, France, July 2005.
13. Z. Ésik and P. Weil. Algebraic recognizability of regular tree languages. *Theoretical Comput. Sci.*, 340:291–321, 2005.
14. B. Genest, D. Kuske, and A. Muscholl. A Kleene theorem and model checking for a class of communicating automata. *Inform. Comput.*, 204(6):920–956, 2006.
15. B. Genest, A. Muscholl, H. Seidl, and M. Zeitoun. Infinite-state High level MSCs: realizability and model-checking. *J. Comput. System Sci.*, 72(4):617–647, 2006 2006.
16. E. Graedel and I. Walukiewicz. Positional determinacy of games with infinitely many priorities. *Methods of Logic in Computer Science*, 2006. to appear.
17. D. Janin and G. Lenzi. On the relationship between monadic and weak monadic second order logic on arbitrary trees, with applications to the mu-calculus. *Fundamenta Informaticae*, 61(3,4):247–265, 2004.
18. A. Muscholl and I. Walukiewicz. An NP-complete fragment of LTL. *International Journal of Foundations of Computer Science*, 16(4):743–455, 2005.
19. D. Niwiński and I. Walukiewicz. Deciding nondeterministic hierarchy of deterministic tree automata. *Elec. Notes Theor. Comp. Sci.*, 123:195–208, 2005.
20. P. Weil. On the logical definability of certain graph poset languages. *J. Automata, Languages and Comput.*, 2005. To appear.

#### **Conférences (d’audience internationale, avec comité de sélection)**

21. B. Bérard, P. Gastin, and A. Petit. Intersection of regular signal-event (timed) languages. In E. Asarin and P. Bouyer, editors, *Proc. of 4th Int. Conf. on Formal Modelling and Analysis of Timed Systems (FORMATS’06)*, volume 4202 of *Lect. Notes Comp. Sci.*, pages 52–66. Springer, 2006.
22. B. Bérard, P. Gastin, and A. Petit. Refinements and abstractions of signal-event (timed) languages. In E. Asarin and P. Bouyer, editors, *Proc. of 4th Int. Conf. on Formal Modelling and Analysis of Timed Systems (FORMATS’06)*, volume 4202 of *Lect. Notes Comp. Sci.*, pages 67–81. Springer, 2006.
23. J. Bernet and D. Janin. Tree automata and discrete distributed games. In M. Liśkiewicz and R. Reischuk, editors, *Fundamentals of Computation Theory*, volume 3623 of *Lect. Notes Comp. Sci.*, pages 540–551. Springer, 2005.
24. J. Bernet and D. Janin. On distributed program specification and synthesis in architectures

- with cycles. In E. Najm, J.-F. Pradat-Peyre, and V. Vigié Donzeau-Gouge, editors, *26th IFIP WG 6.1 International Conference on Formal Methods for Networked and Distributed Systems (FORTE 2006)*, volume 4229 of *Lect. Notes Comp. Sci.*, pages 175–190. Springer, 2006.
25. D. Berwanger and D. Janin. Automata on directed graphs: Edge versus vertex marking. In *Third Int. Conf. on Graph Transformations (ICGT)*, volume 4128 of *Lect. Notes Comp. Sci.*, pages 46–60. Springer, 2006.
  26. P. Bhateja, P. Gastin, and M. Mukund. A fresh look at testing for asynchronous communication. In S. Graf and W. Zhang, editors, *Proc. of 4th Int. Symp. on Automated Tech. for Verification and Analysis (ATVA'06)*, volume 4218 of *Lect. Notes Comp. Sci.* Springer, 2006. To appear.
  27. M. Bojanczyk and I. Walukiewicz. Characterizing EF and EX tree logics. In *Proc. of 15th Int. Conf. on Concurrency Theory, CONCUR '04*, volume 3170 of *Lect. Notes Comp. Sci.*, pages 131–145. Springer, 2004.
  28. A. Bouquet, O. Serre, and I. Walukiewicz. Pushdown games with the unboundedness and regular conditions. In *Proc. of 23rd Foundations of Software Technology and Theoret. Comp. Sci., FSTTCS '03*, volume 2914 of *Lect. Notes Comp. Sci.*, pages 88–99, 2003.
  29. A. Dawar and D. Janin. On the bisimulation invariant fragment of monadic  $\Sigma_1$  in the finite. In *Proc. of 24th Foundations of Software Technology and Theoret. Comp. Sci., FSTTCS '04*, volume 3328, pages 224–236, 2004.
  30. V. Diekert and P. Gastin. Pure future local temporal logics are expressively complete for Mazurkiewicz traces. In *Proc. of 6th Symp. Latin American Theoret. Inform., LATIN '04*, volume 2976 of *Lect. Notes Comp. Sci.*, pages 232–241. Springer, 2004.
  31. M. Droste and P. Gastin. Weighted automata and weighted logics. In G. F. Italiano, editor, *Proc. of 32nd Int. Coll. on Automata, Languages and Programming, ICALP '05*, number 3580 in *Lect. Notes Comp. Sci.*, pages 513–525. Springer, 2005.
  32. P. Gastin and D. Kuske. Uniform satisfiability problem for local temporal logics over Mazurkiewicz traces. In L. de Alfaro, editor, *Proc. of 16th Int. Conf. on Concurrency Theory, CONCUR '05*, number 3653 in *Lect. Notes Comp. Sci.*, pages 533–547. Springer, 2005.
  33. P. Gastin, B. Lerman, and M. Zeitoun. Distributed games and distributed control for asynchronous systems. In M. Farach-Colton, editor, *Proc. of 6th Symp. Latin American Theoret. Inform., LATIN '04*, volume 2976 of *Lect. Notes Comp. Sci.*, pages 455–465. Springer, 2004.
  34. P. Gastin, B. Lerman, and M. Zeitoun. Distributed games with causal memory are decidable for series-parallel systems. In *Proc. of 24th Foundations of Software Technology and Theoret. Comp. Sci., FSTTCS '04*, volume 3328 of *Lect. Notes Comp. Sci.*, pages 275–286. Springer, 2004.
  35. P. Gastin, P. Moro, and M. Zeitoun. Minimization of counterexamples in SPIN. In *Model Checking Software: Proc. of 11th SPIN '04*, volume 2989 of *Lect. Notes Comp. Sci.*, pages 92–108. Springer, 2004.
  36. P. Gastin, N. Sznajder, and M. Zeitoun. Distributed synthesis for well-connected architectures. In N. Garg and S. Arun-Kumar, editors, *Proc. of 26th Foundations of Software Technology and Theoret. Comp. Sci., FSTTCS '06*, *Lect. Notes Comp. Sci.* Springer, 2006. To appear.

37. B. Genest. Compositional message sequence charts (CMSCs) are better to implement than MSCs. In *Proc. of Tools and Algorithms for the construction and analysis of systems, TACAS '05*, volume 3440 of *Lect. Notes Comp. Sci.*, pages 510–525. Springer, 2005.
38. B. Genest. On implementation of global concurrent systems with local asynchronous controllers. In *Proc. of 16th Int. Conf. on Concurrency Theory, CONCUR '05*, volume 3653 of *Lect. Notes Comp. Sci.*, pages 443–457, 2005.
39. B. Genest, D. Kuske, and A. Muscholl. A Kleene theorem and model checking for a class of communicating automata. In *Proc. of Developments in Language Theory, DLT '04*, volume 3340 of *Lect. Notes Comp. Sci.*, pages 30–48. Springer, 2004.
40. B. Genest, D. Kuske, A. Muscholl, and D. Peled. Snapshot verification. In *Proc. of Tools and Algorithms for the construction and analysis of systems, TACAS '05*, volume 3440 of *Lect. Notes Comp. Sci.* Springer, 2005.
41. B. Genest, M. Minea, A. Muscholl, and D. Peled. Specifying and verifying partial order properties using template MSCs. In *Proc. of Foundations of Software Science and Computation Structures, FOSSACS '04*, volume 2987 of *Lect. Notes Comp. Sci.*, pages 195–210. Springer, 2004.
42. B. Genest and A. Muscholl. Message sequence charts: A survey. In *5th Int. Conf. on Application of Concurrency to System Design, ACSD'05*, pages 2–4, 2005. Invited talk.
43. B. Genest and A. Muscholl. Constructing exponential-size deterministic Zielonka automata. In *Proc. of 33rd Int. Coll. on Automata, Languages and Programming, ICALP '06*, volume 4052 of *Lect. Notes Comp. Sci.*, pages 565–576, 2006.
44. H. Gimbert. Parity and exploration games on infinite graphs. In *Proc. of Computer Science Logic, CSL '04*, volume 3210 of *Lect. Notes Comp. Sci.*, pages 56–70. Springer, 2004.
45. H. Gimbert and W. Zielonka. Discounting infinite games but how and why? In L. de Alfaro, editor, *Proc. of Games in Design and Verification '04*, volume 119 of *Elec. Notes Theor. Comp. Sci.*, pages 3–9. Elsevier, 2004.
46. H. Gimbert and W. Zielonka. When can you play positionally? In *Proc. of Symp. on Math. Foundations of Comput. Sci., MFCS '04*, volume 3153 of *Lect. Notes Comp. Sci.*, pages 686–697. Springer, 2004.
47. H. Gimbert and W. Zielonka. Games where you can play optimally without any memory. In *Proc. of 16th Int. Conf. on Concurrency Theory, CONCUR '05*, volume 3653 of *Lect. Notes Comp. Sci.*, pages 428–442. Springer, 2005.
48. H. Gimbert and W. Zielonka. Deterministic priority mean-payoff games as limits of discounted games. In *Proc. of 33rd Int. Coll. on Automata, Languages and Programming, ICALP '06*, volume 4052, part II of *Lect. Notes Comp. Sci.*, pages 312–323. Springer, 2006.
49. L. Hélouët, M. Zeitoun, and A. Degorre. Scenarios and covert channels: another game... In L. de Alfaro, editor, *Proc. of Games in Design and Verification '04*, volume 119 of *Elec. Notes Theor. Comp. Sci.*, pages 93–116. Elsevier, 2004.
50. T. Knapik, D. Niwinski, P. Urzyczyn, and I. Walukiewicz. Unsafe grammars and panic automata. In *Proc. of 32nd Int. Coll. on Automata, Languages and Programming, ICALP '05*, volume 3580 of *Lect. Notes Comp. Sci.*, pages 1450–1461, 2005.
51. S. Lasota and I. Walukiewicz. Alternating timed automata. In *Proc. of Foundations of*

- Software Science and Computation Structures, FOSSACS '05*, number 3441 in *Lect. Notes Comp. Sci.*, pages 250–265, 2005. Journal version available from WWW.
52. O. Ly. Compositional verification: Decidability issues using graph substitutions. In *Proc. of Symp. on Math. Foundations of Comput. Sci., MFCS '04*, volume 3153 of *Lect. Notes Comp. Sci.*, pages 537–549. Springer, 2004.
  53. S. Mohalik and I. Walukiewicz. Distributed games. In *Proc. of 23rd Foundations of Software Technology and Theoret. Comp. Sci., FSTTCS '03*, volume 2914 of *Lect. Notes Comp. Sci.*, pages 338–351, 2003.
  54. A. Murawski, L. Ong, and I. Walukiewicz. Idealized Algol with ground recursion, and DPDA equivalence. In *Proc. of 32nd Int. Coll. on Automata, Languages and Programming, ICALP '05*, volume 3580 of *Lect. Notes Comp. Sci.*, pages 917–929, 2005.
  55. A. Murawski and I. Walukiewicz. Third-order idealized Algol with iteration is decidable. In *Proc. of Foundations of Software Science and Computation Structures, FOSSACS '05*, volume 3441 of *Lect. Notes Comp. Sci.*, pages 202–218, 2005.
  56. A. Muscholl and I. Walukiewicz. An NP-complete fragment of LTL. In *Proc. of Developments in Language Theory, DLT '04*, volume 3340 of *Lect. Notes Comp. Sci.*, pages 334–344. Springer, 2004.
  57. D. Niwiński and I. Walukiewicz. Deciding nondeterministic hierarchy of deterministic tree automata. In R. de Queiroz and P. Cégielski, editors, *Proceedings of WoLLIC'04*, volume 123 of *Elec. Notes Theor. Comp. Sci.*, pages 195–208. Elsevier, 2005.
  58. I. Walukiewicz. A landscape with games in the background. In *Proc. of IEEE LICS '04*, pages 356–366, 2004.
  59. I. Walukiewicz. From logic to games. In *Proc. of 25th Foundations of Software Technology and Theoret. Comp. Sci., FSTTCS '05*, volume 3821 of *Lect. Notes Comp. Sci.*, pages 78–91, 2005.
  60. P. Weil. Algebraic recognizability of languages. In *Proc. of Symp. on Math. Foundations of Comput. Sci., MFCS '04*, volume 3153 of *Lect. Notes Comp. Sci.*, pages 149–175. Springer, 2004.
  61. W. Zielonka. Perfect-information stochastic parity games. In *Proc. of Foundations of Software Science and Computation Structures, FOSSACS '04*, volume 2987 of *Lect. Notes Comp. Sci.*, pages 499–513. Springer, 2004.
  62. W. Zielonka. An invitation to play. In *Proc. of Symp. on Math. Foundations of Comput. Sci., MFCS '05*, volume 3618 of *Lect. Notes Comp. Sci.*, pages 58–70. Springer, 2005.

### Rapports de recherche

63. J. Bernet and D. Janin. Automata theory for distributed games. Technical report, LaBRI, 2004.
64. J.-M. Couvreur, D. Poitrenaud, and P. Weil. Unfoldings for general Petri nets. submitted.