

# Sujet de thèse : Vérification parallèle de haut-niveau d'une algèbre de réseaux de Petri de haut-niveau

**Mots-clés** : vérification, modularité, parallélisme de haut-niveau, réseaux de Petri

## I Laboratoire d'accueil, financement et contacts

**Financement** : Allocation de la région Île-de-France sur la base de 2600 euros brut chargé par mois (soit environ 1570 euros net par mois) pour une durée de 36 mois (3 ans), non cumulable avec un autre financement hors vacations universitaires.

**Laboratoire** : Laboratoire d'Algorithmique, Complexité et Logique (LACL), EA 4213

- a. *Directeur* : Pr Gaétan Hains
- b. *Email* : [gaetan@hains.org](mailto:gaetan@hains.org)
- c. *Équipe* : « systèmes communiquant » dirigée par Pr Elisabeth Pelz
- d. *Adresse complète* : LACL, Université Paris XII--Val de Marne,  
UFR de Sciences et Technologie, Bâtiment P2, 2<sup>ème</sup> étage  
61 avenue du Général de Gaulle, 94010 Créteil
- e. *Téléphone* : 01 45 17 65 95
- f. *Télécopie* : 01 45 17 66 01
- g. *Site Internet* : <http://www.univ-paris12.fr/lacl/>

*Directeur de thèse* : Pr Gaétan Hains

*Co-encadrants* :

- Dr Frédéric Gava (**responsable du projet VEHICULAIRE**)
  - a. *Fonction* : MCF au LACL
  - b. *Téléphone* : 01 45 17 65 67
  - c. *Email* : [gava@univ-paris12.fr](mailto:gava@univ-paris12.fr)
  - d. *Page perso* : <http://www.univ-paris12.fr/lacl/gava/>
- Dr Franck Pommereau
  - a. *Fonction* : MCF au LACL
  - b. *Téléphone* : 01 45 17 66 00
  - c. *Email* : [pommereau@univ-paris12.fr](mailto:pommereau@univ-paris12.fr)
  - d. *Page perso* : <http://www.univ-paris12.fr/lacl/pommereau/>

**Contact** : Envoyer un CV ainsi qu'une lettre de recommandation à l'une des personnes susmentionnées (format PDF de préférence).

## II Sujet de thèse

La vérification automatique de modèles (appelée *model checking* [13]) est une approche séduisante mais généralement coûteuse en termes de temps de calcul et d'espace mémoire nécessaires (on parle de l'explosion de l'espace des états ou « *state-space explosion problem* »). Beaucoup de travaux dans le domaine sont dédiés à l'accélération des calculs et à la réduction de l'espace d'exploration. La parallélisation de ce calcul est l'une des techniques envisageables. Cependant, elle est en général employée avec des modèles de bas-niveau (tant au niveau du modèle vérifié que du langage et du parallélisme utilisés pour implanter le *model checker* [5] [8] [11]) et avec une distribution plus ou moins naïve des données (comme par exemple <http://quasar.cnam.fr/>).

Le travail de cette thèse est tout d'abord la conception d'un algorithme parallèle BSP [15] [16] de construction de l'espace d'exploration d'une algèbre de réseaux de Petri colorés de haut-niveau, appelé M-Net [7] (ou d'un sous-ensemble clairement défini pour nos types d'applications

[3] [17] ) ainsi que de la vérification parallèle de propriétés logiques sur ce graphe (certaines logiques comme LTL [13] ou autres sont souvent très expressives mais un sous-ensemble expressif et vérifiable efficacement sera nécessaire). L'utilisation d'un tel modèle fortement structuré nous permettra d'en déterminer des caractéristiques structurelles ouvrant la voie à un parallélisme de haut-niveau (plus structuré) et donc plus efficace (et portable) qu'est le modèle BSP.

Ensuite, une implantation modulaire et polymorphe (indépendante des types de données, donc idéale pour le *model checking* symbolique [2] et pour une algèbre de haut-niveau) sera effectuée avec une bibliothèque de programmation parallèle de haut-niveau, appelée BSMLlib [9] (basé sur le langage OCaml<sup>1</sup> [14] ), développée conjointement au LACL et au LIFO (laboratoire d'informatique d'Orléans). Des optimisations seront apportées à cet algorithme tout au long de cette thèse notamment avec l'extension de techniques d'équilibrage de charge basés sur les coûts BSP [1] ou sur la structure de la formule logique vis-à-vis du modèle [12] .

Enfin, des tests appliqués de notre logiciel à des problèmes de sécurité informatique (thème fédérateur du LACL) ainsi que sur des propriétés de programmes hautes performances (C, Ada ou Fortran avec une bibliothèque de calculs parallèles telle que MPI ou avec squelettes algorithmiques<sup>2</sup> ) seront effectués sur les différentes grappes de PCs du LACL et du LIFO. Des adaptations et des ajouts seront par la suite (ou au fur et à mesure) ajoutés pour modéliser et vérifier des propriétés logiques de problèmes informatique ou biologiques [18] qui seront rencontrés dans la littérature.

Vous trouverez plus de détails sur le projet VEHICULAIRE auquel appartient cette thèse à l'adresse suivante : <http://www.univ-paris12.fr/lacl/gava/vehiculaire/vehiculaire.pdf>

### III Références

- [1] M. Bamha and G. Hains. An Efficient equi-semi-join Algorithm for Distributed Architectures. . In V. S. Sunderam, G. Dick van Albada, P. M. A. Sloot, and J. Dongarra, editors, International Conference on Computational Science (ICCS 2005), Part II, number 3515 in LNCS, Springer-Verlag, 2005.
- [2] B. Boigelot. Symbolic Methods for Exploring Infinite State Spaces. PhD Thesis, volume 189, Collection des Publications de la Faculté des Sciences Appliquées de l'Université de Liège, 1999.
- [3] R. Bouroulet, H. Klaudel, and E. Pelz. A semantics of security protocol language using a class of composable high-level petri nets. In Application of Concurrency to System Design, volume 4th ACS D, pages 99-108. IIEE, 2004.
- [4] Jan Duennweber, Sergei Gorlatch, Anne Benoit and Murray Cole Integrating MPI-Skeletons with Web Service in Proceedings of ParCo 2005.
- [5] H. Garavel, R. Mateescu, C. Joubert and al. DISTRIBUTOR and BCG\_MERGE: Tools for Distributed Explicit State Space Generation, March 2006.
- [6] G. Gardey, O. H. Roux and O. F. Roux. State space computation and analysis of time Petri nets. Theory and Practice of Logic Programming (TPLP). Special Issue on Specification Analysis and Verification of Reactive Systems, 2005.
- [7] Hanna Klaudel and Franck Pommereau. M-nets, a survey. Acta Informatica, À paraître.
- [8] L. Kristensen and L. Petrucci. An approach to distributed state space exploration for coloured Petri nets. ICATPN'2004, LNCS 3099. Springer, 2004
- [9] F. Loulergue, F. Gava, and D. Billiet. Bulk Synchronous Parallel ML: Modular Implementation and Performance Prediction. In, International Conference on Computational Science (ICCS 2005), Part II, number 3515 in LNCS, pages 1046-1054. Springer, 2005.
- [10] M. Mäkelä. Modular reachability analyzer for high-level Petri nets. 5th Workshop on Discrete Event Systems. Kluwer Academic Publishers, 2000.
- [11] D.Petcu, Parallel explicit-state reachability analysis and state space construction, Proceedings of Second International Symposium on Parallel and Distributed Computing, ISPD 2003, 13-14 October 2003, Ljubljana, IEEE Computer Society Press, Los Alamitos
- [12] J.-M. Couvreur and D. Poitrenaud. Dépliage pour la vérification de propriétés temporelles. In Vérification et mise en œuvre des réseaux de Petri - Tome 2, chapter 3, pages 127-161. Hermès, 2003.
- [13] Model Checking, Edmund M. Clarke, Jr., Orna Grumberg and Doron A. Peled, MIT Press, 1999
- [14] D. Rémy. Using, Understanding, and Unravelling the OCaml Language. In G. Barthe, P. Dyjber, L. Pinto, and J. Saraiva, editors, Applied Semantics, number 2395 in LNCS, pages 413-536. Springer, 2002.
- [15] R. Bisseling, Parallel Scientific Computation. A Structured approach using BSP and MPI, Oxford University Presse, 2004.
- [16] D. B. Skillicorn, J. M. D. Hill, and W. F. McColl. Questions and answers about BSP. Scientific Programming, 6(3): 249-274
- [17] Franck Pommereau Versatile Boxes: a Multi-Purpose Algebra of High-Level Petri net Proc. of DASD'07, SCS/ACM, 2007
- [18] C. Chaouiya (à paraître) *Petri net modelling of biological networks*. Briefings in Bioinformatics.

---

<sup>1</sup> <http://caml.inria.fr>

<sup>2</sup> Un squelette algorithmique est une « fonction » qui peut être implantée en parallèle : chaque type de parallélisme (diviser-pour-régner, *pipeline* etc.) se voit doté d'un squelette ; cela permet de paralléliser facilement et de manière plus sûre des programmes, en utilisant des squelettes appropriés et qui « représentent » bien le parallélisme intrinsèque du problème [4] .