Doctoral research project:
Parallel verification
# of a high-level Petri net algebra

Keywords: verification, modularity, high-level parallelism, Petri nets

I Host laboratory, funding and contacts

***Funding:* Paris regional doctoral scholarship, value of approximately 1570€/month for a duration of 3 years.**

Laboratory: **Laboratory for Algorithmics, Complexity and Logic** (LACL), **University Paris-12**

a. *Director*: Pr Gaétan Hains
b. *Email*: gaetan@hains.org
c. *Team*: "communicating systems" directed by Pr Elisabeth Pelz
d. *Postal address*: LACL, Université Paris XII--Val de Marne,
UFR de Sciences et Technologie, Bâtiment P2, 2ème étage
61 avenue du Général de Gaulle, 94010 Créteil
e. *Phone*: +33 (0)1 45 17 65 95
f. *Fax*: +33 (0)1 45 17 66 01
g. *Web site*: http://www.univ-paris12.fr/lacl/

Doctoral advisor: Pr Gaétan Hains
Co-advisors:

- Dr Frédéric Gava (**coordinator of the VEHICULAR project**)
  a. *Position*: Assistant Professor at LACL
  b. *Phone* : +33 (0)1 45 17 65 67
  c. *Email*: gava@univ-paris12.fr
  d. *Web page*: http://www.univ-paris12.fr/lacl/gava/
- Dr Franck Pommereau
  a. *Position*: Assistant Professor at LACL
  b. *Phone*: +33 (0)1 45 17 66 00
  c. *Email*: pommereau@univ-paris12.fr
  d. *Web page*: http://www.univ-paris12.fr/lacl/pommereau/

***Application*: Send a CV and cover letter to the one of the above-mentioned advisors (in PDF format preferably).**

II Doctoral research subject

The automatic verification of models (also called model-checking [13]) is a useful technique for automatic software verification, but it is generally expensive in terms of memory capacity and computing time (one speaks of its "state-space explosion problem"). Many works in this field have dealt with the acceleration of computation and reduction of the space exploration. Parallelization is one of the possible techniques to this end. However, it is in generally implemented with low-level models (both for its specification as model-checker and as parallel program [5] [8] [11]) and with a more or less naive data distribution (such as for example http://quasar.cnam.fr/).

The proposed work is first of all the design of a BSP [15] [16] parallel algorithm for the construction of the exploration of the state-space of a high-level coloured Petri net algebra, called M-Net [7] (or of a clearly definite subset [3] [17]) as well as a parallel checking of logical properties on this graph (some logics like LTL [13] or others are often very expressive but an expressive and effectively verifiable subset will be necessary). The use of such a strongly structured model will enable us to determine some structural characteristics opening the way of a high-level parallelism (more structured) and thus more efficient (and portable) which is the BSP model.

Then, a modular and polymorphic implementation (data type independent, therefore ideal for the model-cheking of symbolic system [2] and for a high-level algebra) will be carried out with a library for high-level parallel programming, called BSMLlib [9] BSMLlib is based on the OCaml[1] language [14] and wad developed jointly by LACL and LIFO (Computer Science laboratory of the University of Orleans). Optimizations to the algorithm will be made in particular with the extension of techniques of load-balancing based on BSP costs [1] or on the structure of the logical formulae with respect to the model [12] .

Finally, tests of our software applied to computer security problems (federating topic for LACL research) and to properties of high performance programs (C, Ada or FORTRAN with a parallel computation library such as MPI or with algorithmic skeletons[2]) will be carried out on the various PC's clusters at LACL and LIFO. Adaptations and additions will thereafter (or progressively) be added to model and check logical properties of computer science's or biological problems [18] which will be encountered in the literature.

More details (in French) on the VEHICULAR project can be found at: http://www.univ-paris12.fr/lacl/gava/vehiculaire/vehiculaire.pdf

III Bibliography

[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 ACSD, pages 99-108. IIIE, 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, ISPDC 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 Unravellling 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 paraitre) *Petri net modelling of biological networks.* Briefings in Bioinformatics.

[1] http://caml.inria.fr

[2] An *algorithmic skeleton* is a function which can be implemented in parallel: each kind of parallelism (divide-and-conquer, pipeline, task farming *etc.*) is realised by a skeleton; this makes possible the easy and safe parallelisation of programs, by using suitable skeletons that closely reflect the parallelism intrinsic in the computational problem [4] .