

Alexandre DURET-LUTZ

4-6 rue du 8 mai 1945,  
94110 Arcueil  
adl@gnu.org  
(+33) 9 50 38 87 38

31 ans  
nationalité française

LRDE / EPITA  
14-16 rue Voltaire,  
94270 Le Kremlin-Bicêtre  
adl@lrde.epita.fr  
(+33) 1 53 15 59 17  
<http://www.lrde.epita.fr/~adl/>

---

## Situation actuelle

---

Docteur en informatique de l'université Paris 6, diplômé de l'EPITA, actuellement enseignant-chercheur au Laboratoire de Recherche de l'EPITA.

---

## Cursus

---

- 2003–2007
- Doctorat au sein de l'équipe *Modélisation et vérification* du *Laboratoire d'Informatique de Paris 6* (LIP6), à l'université P. & M. Curie.  
  
Sujet : *Contributions à l'approche automate pour la vérification de propriétés de systèmes concurrents*. Thèse dirigée par Fabrice KORDON ([fabrice.kordon@lip6.fr](mailto:fabrice.kordon@lip6.fr)) et encadrée par Denis POITRENAUD ([denis.poitrenaud@lip6.fr](mailto:denis.poitrenaud@lip6.fr)).
- 2002–2003
- DEA Systèmes Informatiques Répartis (Edite de Paris). Rang : 1/24. Mention très bien.
- 2001–2002
- Ingénieur de recherche au Laboratoire de Recherche et Développement de l'EPITA (LRDE).  
Domaine d'activité : génie logiciel (patrons de conception, généricité, méta-programmation) appliqué au traitement d'images.
- 1998–2001
- Cycle d'ingénierie à l'EPITA (École Pour l'Informatique et les Techniques Avancées). Spécialisation Sciences Cognitives et Informatique Avancée (SCIA). Mention bien.
  - En parallèle : formation par la recherche au Laboratoire de Recherche et Développement de l'EPITA (LRDE).
- 1996–1998
- Mathématiques Spéciales MP au lycée Clémenceau à Reims.
  - Mathématiques Supérieures MPSI au lycée Roosevelt à Reims.

## Enseignements

### *Récapitulatif des enseignements*

Niveau	TP	TD	Cours	Années	
Licence : 1 <sup>re</sup> année	Prog. impérative et éléments d'algorithmique 28h (mi-TD mi-TP)			2003	
Licence : 2 <sup>e</sup> année	Programmation Système et Internet 95h		41h	2003–2006	
	21	Initiation au C 62h		2004–2006	
Licence : 3 <sup>e</sup> année	Environnement de développement 81h		81h	2004–2006	
Master : 2 <sup>e</sup> année	Model checking		2h	2003	
	Administration et architecture des systèmes		12h	2006	
Info Spé : (Classe prépa, Bac+2)	Théorie des langages rationnels 36h		60h	2007–	
Ingénieur : 1 <sup>re</sup> année (Bac+3)	Complexité des algorithmes		6h	136h	2007–
	Algorithmique distribuée			14h	2009–
	OCaml et son système de modules			6h	2002
	Utilisation et implémentation des exceptions			3h	2002
Ingénieur : 2 <sup>e</sup> année (Bac+4)	Initiation au model checking		16h	2008–	
Apprentissage : 1 <sup>re</sup> année (Bac+3)	Complexité des algorithmes		4h	83h	2007–
	Modélisation par objets			28h	2007
En entreprise	Objets, C++, STL, et prog. générique 80h			1999–2002	

### *Détail des enseignements*

Depuis 2007

- Enseignant-chercheur à l'EPITA.
  - Info Spé (deuxième année de classes préparatoires) : Rédaction et encadrement de TD et TP sur les automates et la théorie des langages rationnels.
  - Première année d'ingénierie : Cours de tronc commun sur la complexité des algorithmes (calculs de complexité, structures de données, algorithmes diviser pour régner, programmation dynamique), cours optionnel sur l'algorithmique distribuée (problèmes fondamentaux tels que l'exclusion mutuelle, le consensus ou l'élection).
  - Second année d'ingénierie : cours d'introduction au model checking (logiques temporelles, automates sur mots infinis, algorithmes associés).
  - En première année d'apprentissage : complexité des algorithmes (même cours que pour le cycle d'ingénierie), cours de modélisation par objets (pour décharger le prof. habituel).

---

## Enseignements (suite)

- 2003–2006
- Moniteur puis ATER à l’université P. & M. Curie (Paris 6)
    - Licence 1<sup>re</sup> année : Programmation impérative et éléments d’algorithmique (LI102). Introduction à la programmation impérative en Visual Basic sous Excel (principalement boucles, récursion, tris, et tableaux). Séances de tutorat pour les étudiants en difficulté du module, 28h eq.TD.
    - Licence 2<sup>e</sup> année : Programmation Système et Internet (LI232). Programmation de scripts shell Unix, notions de programmation événementielle et concurrente, programmation scripts CGI. Doublures de TP pour 3 groupes et chargé de TD pour 2 groupes, 115h eq.TD.
    - Licence 2<sup>e</sup> année : Initiation au C (LI205). Introduction à la programmation impérative en C sous Unix (principalement boucles, récursion, tris, et tableaux.) Chargé de TD pour un groupe, doublures de TP pour deux groupes, 69h eq.TD.
    - Licence 3<sup>e</sup> année : Environnement de développement (LI365). Outils de développement sous Unix : shell, make, emacs, gdb, cvs, ... Chargé de TD pour un groupe : 162h eq.TD. Implication dans la rédaction des TD, TP, et examens.
    - Master 2<sup>e</sup> année. Un cours de 2h sur *le model checking par l’approche automate* pour un module du DEA Systèmes Informatiques Répartis (maintenant “Master Systèmes et Applications Répartis”).
    - Master 2<sup>e</sup> année. TD du module *architecture et administration des systèmes* du “Master Systèmes et Applications Répartis”. 12h.
- 2002
- Vacataire à l’EPITA. Deux cours de trois heures sur OCaml et son système de modules, aux élèves de première année d’ingénierie à l’EPITA.
- 2002
- Vacataire à l’EPITA. Présentation de trois heures sur les exceptions. Utilisation (syntaxe, *exception safety*, ...) dans différents langages (C++, Ada, Java, Eiffel), et implémentations (sjlj et 0-cost).
- 1999–2002
- Cryo Interactive, Paris. (4 formations d’une semaine chacune.) Préparation du support de formation (paradigme des objets, C++ et STL, programmation générique) et encadrement des TPs pour une série de formations dispensées par l’EPITA pour Cryo. Responsables : Philippe AUBESSARD (Cryo), Laurent TRÉBULLE (EPITA). 80h de présentiel.

---

## Activités de recherche

- Mots clefs
- Méthodes formelles de vérification de modèles, model checking, logiques temporelles, automates de Büchi, automates de Streett, hypothèses d’équité, ordre partiel, tests de vacuité.

## Activités de recherche (suite)

### Problématique

- Mes recherches s'inscrivent dans le cadre des méthodes formelles de vérification de systèmes concurrents finis. Pour un tel système dont on possède d'une part une modélisation, et d'autre part un ensemble de propriétés comportementales qu'il doit vérifier, on peut distinguer plusieurs méthodes de vérification allant de la preuve manuelle à l'exploration exhaustive de l'espace des états du système.

Les méthodes basées sur l'exploration de l'espace d'états, désignées aussi sous le nom de *model checking*, constituent la branche automatisée des méthodes formelles : on cherche à déterminer algorithmiquement si un modèle donné vérifie une propriété donnée.

Je m'intéresse plus particulièrement à l'approche par automates du *model checking*. Elle procède en plusieurs étapes :

- L'ensemble des états du système modélisé est développé sous la forme d'un automate  $A$  dont les nœuds représentent des configurations du système et les transitions des changements de configuration possibles. Le langage reconnu par  $A$ , c'est-à-dire l'ensemble des chemins existants dans  $A$ , représente l'ensemble des exécutions possibles du système modélisé.
- La propriété à vérifier sur le système, généralement exprimée avec une logique temporelle, est elle-même traduite sous la forme d'un automate  $B$  dont le langage représente l'ensemble des comportements qui ne vérifient pas la propriété.
- La synchronisation de ces deux automates produit un nouvel automate  $C = A \otimes B$  dont le langage est l'intersection des deux précédents : l'ensemble des exécutions du système qui ne vérifient pas la propriété.
- Ce dernier automate est ensuite exploré par un algorithme d'*emptiness check* (test de vacuité) pour déterminer si son langage est vide (la propriété est vérifiée) ou non (il existe un contre-exemple).

Comme les systèmes que nous modélisons ont des comportements infinis, les automates manipulés ici travaillent sur des mots de longueur infinie. Les automates généralement utilisés à cette fin sont des automates Büchi, et c'est ce qui nécessite l'emploi d'un algorithme d'*emptiness check* spécifique dans la dernière étape.

Cette approche souffre d'un problème majeur : l'explosion de l'espace d'états. En effet, les systèmes que l'on cherche à vérifier peuvent être gros au point que développer l'ensemble de leurs états (l'automate  $A$ ) en mémoire n'est plus possible. D'autres contraintes, comme l'ajout d'hypothèses d'équité (p.ex. équité entre processus dans un système concurrent), participent à l'explosion des automates manipulés.

L'objectif de ma recherche est de contribuer à la réduction de cette explosion, en apportant des optimisations à ces différentes étapes du *model checking*.

### Réalisations

- Les différents algorithmes étudiés et leur benchmarks sont tous implémentés au sein de Spot, une bibliothèque de *model-checking* commencée lorsque j'étais au LIP6. Cette bibliothèque d'environ 60000 lignes de C++ est disponible en open source sur le site <http://spot.lip6.fr>, et permet d'expérimenter différentes configurations d'algorithmes.

### Encadrement

- J'ai encadré 4 étudiants de l'Epita et un stagiaire de DEA sur le projet Spot (mentionné ci-dessus), ainsi que 8 étudiants de l'Epita sur le projet Vaucanson (une bibliothèque générique pour la manipulation d'automates finis pondérés).

## Publications

### *Revue nationale*

- 2004 • **Nouvelles techniques de model-checking pour la vérification de systèmes complexes.** Y. THIERRY-MIEG, S. BAARIR, A. DURET-LUTZ, et F. KORDON. *Génie Logiciel*, no 69, Juin 2004, pages 17–23.

### *Conférences internationales avec comité de sélection*

- 2009 • **On-the-fly Emptiness Check of Transition-based Streett Automata.** A. DURET-LUTZ, D. POITRENAUD and J.-M. COUVREUR. Proceedings of the 7th International Symposium on Automated Technology for Verification and Analysis (ATVA'09), volume 5799 of Lecture Notes in Computer Science, pages 213–227. Springer-Verlag. October 2009.
- 2008 • **An XML format proposal for the description of weighted automata, transducers, and regular expressions.** A. DEMAILLE, A. DURET-LUTZ, F. LESAIN, S. LOMBARDY, J. SAKAROVITCH, and F. TERRONES. In the proceedings of the seventh international workshop on Finite-State Methods and Natural Language Processing (FSMNLP'08). Sep, 2008.
- 2007 • **Emptiness Check of Powerset Büchi Automata using Inclusion Tests.** S. BAARIR, A. DURET-LUTZ. In the Proceedings of the 7th International Conference on Application of Concurrency to System Design (ACSD'07), July 2007. IEEE Computer Society. Pages 41–50.
- 2005 • **On-the-fly Emptiness Checks for Generalized Büchi Automata.** J.-M. COUVREUR, A. DURET-LUTZ, and D. POITRENAUD. In the proceedings of the 12th International SPIN Workshop on Model Checking of Software, volume 3639 of Lecture Notes in Computer Science, pages 143–158. Springer-Verlag. August 2005.
- 2004 • **SPOT : an Extensible Model Checking Library using Transition-based Generalized Büchi Automata.** A. Duret-Lutz and D. Poitrenaud. In the proceedings of the 12th IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'04), pages 76–83, Volendam, The Netherlands, October 2004. IEEE Computer Society Press.
- **Improving Reachability Analysis for Partially Symmetric High Level Petri Nets.** S. BAARIR, J.-M. ILIÉ, and A. DURET-LUTZ. In the proceedings of the poster session of the 12th IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'04), pages 5–8, Volendam, The Netherlands, October 2004.
- 2003 • **A Static C++ Object-Oriented Programming (SCOOP) Paradigm Mixing Benefits of Traditional OOP and Generic Programming.** N. BURRUS, A. DURET-LUTZ, T. GÉRAUD, D. LESAGE, and R. POSS. In the proceedings of the workshop on Multiparadigm Programming with OO languages (MPOOL 2003), Anaheim, California, 2003.
- **Multi-band Segmentation using Morphological Clustering and Fusion — Application to Color Image Segmentation.** H. XUE, T. GÉRAUD, and A. DURET-LUTZ. In the proceedings of the IEEE International Conference on Image Processing (ICIP 2003), Barcelona, September 2003.
- 2002 • **Generic Implementation of Morphological Image Operators.** J. DARBON, T. GÉRAUD, and A. DURET-LUTZ. In the proceedings of the International Symposium on Mathematical Morphology VI (ISMM'2002), Sydney, New South Wales, Australia, April 2002.
- 2001 • **Expression Templates in Ada.** A. DURET-LUTZ. (**Best paper award.**) In the Proceedings of the 6th International Conference on Reliable Software Technologies - Ada-Europe 2001, Leuven, Belgium, May 14–18, 2001. Volume 2043 of Lectures Notes in Computer Science, Springer-Verlag, 2001, pages 191–202.

## Publications (suite)

- **Applying Generic Programming to Image Processing.** T. GÉRAUD, Y. FABRE, and A. DURET-LUTZ. In the proceedings of the IASTED International Conference on Applied Informatics (AI'2001) - Symposium Advances in Computer Applications, ACTA Press, pages 577–581, Innsbruck, Austria, February 2001.
- **Design Patterns for Generic Programming in C++.** A. DURET-LUTZ, T. GÉRAUD, and A. DEMAILLE. In the proceedings of the 6th USENIX Conference on Object-Oriented Technologies and Systems (COOTS'2001), pages 189–202, San Antonio, Texas, USA, January-February 2001.
- 2000 • **Olena : a Component-Based Platform for Image Processing, mixing Generic, Generative and OO Programming.** A. DURET-LUTZ. In the Proceedings of the 2nd International Symposium on Generative and Component-Based Software Engineering (GCSE 2000), Young Researchers Workshop (published in "Net.ObjectDays2000").
- **Obtaining Genericity for Image Processing and Pattern Recognition Algorithms.** T. GÉRAUD, Y. FABRE, A. DURET-LUTZ, D. PAPADOPOULOS-ORFANOS, and J.-F. MANGIN. In the proceedings of the 15th International Conference on Pattern Recognition (ICPR'2000), IEEE Computer Society, vol. 4, pages 816-819, Barcelona, Spain, September 2000.
- **Generic Programming Redesign Pattern.** T. GÉRAUD and A. DURET-LUTZ. In the proceedings of the 5th European Conference on Pattern Languages of Programs (EuroPLOP'2000), Irsee, Germany, July 2000.

### *Conférence nationale avec comité de sélection*

- 2007 • **Test de vacuité pour automates de Büchi ensemblistes avec tests d'inclusion.** S. BAARIR, A. DURET-LUTZ. Actes du 6<sup>e</sup> Colloque Francophone sur la Modélisation des Systèmes Réactifs (MSR'07), Octobre 2007. **À paraître.**

### *Rapports techniques*

- 2006 • **Emptiness Check of Powerset Büchi Automata using Inclusion Tests** S. BAARIR, A. DURET-LUTZ. Technical Report 2006/003, Université Pierre et Marie Curie, LIP6-CNRS, Paris, France, October 2006.
- 2003 • **SPOT : une bibliothèque de vérification de propriétés de logique temporelle à temps linéaire.** Mémoire de DEA, DEA Systèmes Informatiques Répartis, Université de Paris 6, septembre 2003.
- 2000 • **Improving Object-Oriented Generic Programming.** A. DURET-LUTZ and T. GÉRAUD. Technical Report 0001, EPITA-LRDE, April 2000.
- 1999 • **Une bibliothèque et un environnement de programmation d'opérateurs de traitement d'images.** R. CLOUARD, A. ELMOATAZ, F. ANGOT, O. LEZORAY et A. DURET-LUTZ. Rapport interne - No. 99008 - GREYC-ISMRA - novembre 1999.
- **Patrons de conception statiques pour la programmation générique en C++.** A. DURET-LUTZ et T. GÉRAUD. Rapport technique 9903, EPITA-LRDE, septembre 1999.
- **Composants génériques de calcul scientifique.** T. GÉRAUD et A. DURET-LUTZ. Rapport technique 9901, EPITA-LRDE, mars 1999.