CONSEILLER SCIENTIFIQUE

LIONEL SEINTURIER

Professeur des universités
Université de Lille

Je suis Professeur des Universités en informatique à l'Université de Lille depuis 2006. J'ai été auparavant Maître de conférences à Sorbonne Université (UPMC, Paris 6) et j'ai soutenu ma thèse de Doctorat en 1997 au CNAM Paris. De 2011 à 2016, j'ai été membre junior de l'institut Universitaire de France (IUF). Je dirige depuis 2013 une équipe-projet commune à l'Institut National de la Recherche en Informatique et en Automatique (Inria) et l'Université de Lille. De 2019 à 2023, j'ai été président de la section Informatique (27) du Conseil national des universités.

Dans le domaine de l'informatique, mes recherches portent sur les systèmes répartis et le génie logiciel. Mes contributions scientifiques sont dans le domaine des composants et des architectures logicielles pour les systèmes répartis. Dans une première partie de ma carrière, j'ai travaillé sur la notion de programmation par aspects. Je suis notamment le co-auteur d'un framework de référence pour la notion de programmation par aspects dynamique. Depuis une dizaine d'années, je travaille sur les systèmes logiciels auto-adaptables. Il s'agit de proposer de nouvelles solutions pour faire en sorte que ces systèmes puissent évoluer autant que possible en autonomie dans des environnements d'exécution ouverts. Je m'intéresse plus particulièrement à deux propriétés : l'auto-optimisation et l'auto-protection. Avec l'auto-optimisation, il s'agit de partager, collecter et analyser les comportements et les données distribués pour adapter, optimiser et maintenir en permanence les systèmes logiciels. Avec l'autoprotection, il s'agit automatiser autant que possible la sécurité des systèmes logiciels face aux comportements malveillants et aux vulnérabilités.

Autres responsabilités exercées

- Membre du Conseil scientifique de l'Université de Lille
- Responsable de l'équipe-projet Spirals commune à Inria et à l'Université de Lille

Principales publications

- S. Brisset, R. Rouvoy, L. Seinturier, R. Pawlak. *STFM: Fast Matching of Web Pages using Similarity-based Flexible Tree Matching*. Information Systems (**IS**). Volume 112. February 2023. Elsevier.
- S. Brisset, R. Rouvoy, L. Seinturier, R. Pawlak. *ERRATUM: Leveraging Flexible Tree Matching to Repair Broken Locators in Web Automation Scripts*. Information and Software Technology (**IST**). Volume 144. April 2022. Elsevier.
- G. Fieni, R. Rouvoy, L. Seinturier. *SelfWatts: On-the-fly Selection of Performance Events to Optimize Software-defined Power Meters*. 21th IEEE/ACM International Symposium on Cluster, Cloud, and Internet Computing (**CCGrid'21**). Melbourne, Australia. May 2021.
- Z. Yu, C. Bai, L. Seinturier, M. Monperrus. *Characterizing the Usage, Evolution and Impact of Java Annotations in Practice*. IEEE Transactions on Software Engineering (**TSE**). 47(5):969-986. May 2021.
- M. Colmant, R. Rouvoy, M. Kurpicz, A. Sobe, P. Felber, L. Seinturier. *The Next 700 CPU Power Models*. Journal of Systems and Software (**JSS**). 144:382-396. October 2018. Elsevier.
- F. Alvares, E. Rutten, L. Seinturier. *A Domain-specific Language for The Control of Self-adaptive Component-based Architecture*. Journal of Systems and Software (**JSS**). 130:94-112. August 2017. Elsevier.
- B. Cornu, E. Barr, M. Monperrus, L. Seinturier. *Casper: Automatic Tracking of Null Dereferences to Inception with Causality Traces*. Journal of Systems and Software (**JSS**). 122:52-62. December 2016. Elsevier.

- J. Xuan, B. Cornu, M. Martinez, B. Baudry, L. Seinturier, M. Monperrus. *B-Refactoring: Automatic Test Code Refactoring to Improve Dynamic Analysis*. Information and Software Technology (**IST**). 76:65-80. August 2016.
- F. Paraiso, P. Merle, L. Seinturier. *soCloud: A Service-Oriented Component-Based PaaS for Managing Portability, Provisioning, Elasticity, and High Availability Across Multiple Clouds*. **Computing** Journal. 98(5):539-565. May 2016.
- R. Pawlak, M. Monperrus, N. Petitprez, C. Noguera, L. Seinturier. *Spoon: A Library for Implementing Analyses and Transformations of Java Source Code*. Software Practice and Experience (**SPE**). 46(9):1155-1179. September 2016. Wiley.