# Application BASTRI
## Fiches Equipes

## LANDE (SR0126ZR)

Conception et validation de logiciels
LANDE  CELTIQUE (SR0327HR)

**Statut:** Terminée

**Responsable :** Thomas Jensen

**Mots-clés de "A - Thèmes de recherche en Sciences du numérique - 2023" :** *Aucun mot-clé.*

**Mots-clés de "B - Autres sciences et domaines d'application - 2023" :** *Aucun mot-clé.*

**Domaine :** Algorithmique, programmation, logiciels et architectures
**Thème :** Programmation, vérification et preuves

**Période :** 04/10/1994 -> 31/12/2008
**Dates d'évaluation :**

**Etablissement(s) de rattachement :** U. RENNES 1, ENS CACHAN, CNRS, INSA RENNES
**Laboratoire(s) partenaire(s) :** IRISA (UMR6074)

**CRI :** Centre Inria de l'Université de Rennes
**Localisation :** Centre Inria de l'Université de Rennes
**Code structure Inria :** 031017-0

**Numéro RNSR :** 199418361N
**N° de structure Inria:** SR0126ZR

### Présentation

Le thème de recherche central de l'équipe-projet LANDE est la conception d'outils d'aide au développement et à la validation de logiciels. Notre approche est fondée sur une collection de méthodes formelles permettant de spécifier ou d'extraire une *vue partielle* de l'architecture et du comportement d'un logiciel. Cette approche nous a amenés à étudier deux types de problèmes. D'une part, la spécification d'un logiciel en vues partielles nécessite une vérification de la *cohérence* entre plusieurs vues afin que celles-ci puissent être synthétisées. D'autre part, l'extraction d'une vue demande des techniques d'analyse statique et dynamique précises. Nous insistons sur la nécessité que les réponses apportées à ces problèmes reposent sur des bases formelles (sémantique du langage étudié, définitions de propriétés à vérifier dans des logiques formelles) afin d'obtenir une garantie sur les résultats produits. De plus, il est important que les outils construits soient le plus automatiques possible car les utilisateurs visés ne sont pas nécessairement des experts en méthodes formelles.

### Axes de recherche

La **description multi-vues** de l'architecture de gros logiciels a comme objectif de spécifier l'organisation globale de systèmes afin d'améliorer la maîtrise de leur développement (spécification, analyse, programmation, test, maintenance, etc.). Une ambition majeure dans ce domaine est le passage à l'échelle de techniques comme l'analyse, le raffinement ou la vérification de programmes. Nos travaux visent à assurer une forme de cohérence de ces descriptions hétérogènes. Le défi principal est de trouver comment mettre en relation des vues qui mettent en jeu des propriétés de nature très différentes ou qui se situent à des niveaux d'abstraction différents. Une fois les vues mises en relation, il devient possible d'utiliser des techniques standards d'analyse statique ou de vérification afin d'assurer des propriétés de cohérence.
La nouvelle technique de programmation appelée " **programmation par aspects** " consiste à décrire un logiciel comme un ensemble formé d'un composant principal et d'une collection de vues ou d'*aspects* décrivant des tâches comme la gestion mémoire, la synchronisation, les optimisations, etc. Un outil, appelé tisseur, est chargé de produire automatiquement un programme intégrant les différents aspects au composant principal. L'intérêt de cette approche est de localiser (dans les aspects) des choix de mise en oeuvre qui seraient sinon dispersés dans le code source. Après avoir proposé un aspect dédié à la sécurisation de code mobile, nous avons étudié les problèmes d'interactions qui se posent lorsque l'on doit tisser plusieurs aspects. En effet, comme pour les vues, les aspects ne sont pas obligatoirement orthogonaux et des conflits ou ambiguïtés peuvent apparaître lors du tissage. Nous travaillons

### Contact

- **Responsable :** Thomas Jensen
- **Tél :** 02.99.84.74.78
- **Secrétariat Tél :** 02.99.84.72.06

### En savoir plus

- Site de l'équipe
- Site sur inria.fr
- Derniers Rapports d'Activité :

### Documents sur la structure

- Intranet
- Privés

### Décisions

- 5789 (11/09/2007) : prolongation
- 6490 (06/02/2009) : fermeture

### Localisation

- **Adresse postale :** Centre Inria de l'Université de Rennes 263, avenue du Général Leclerc Campus universitaire de Beaulieu 35042 Rennes Cedex France
- **Coordonnées GPS :** 48.116, -1.64

également sur un langage d'aspect dédié à la composition de composants.

Pour faciliter la navigation et l'organisation de logiciels de taille importante nous cherchons à définir **un cadre logique pour les systèmes d'information** qui décrit uniformément la navigation, l'interrogation et l'analyse des données. Ce cadre est générique par rapport à la logique utilisée pour naviguer et interroger ; en particulier il peut être appliqué à plusieurs types de logiques de programme, comme les types ou des propriétés statiques. Ces travaux se basent sur une extension de la théorie de l'analyse de comcepts.

La validation d'un logiciel utilisent des méthodes d'**analyses et de test de programmes**. Nous nous intéressons à différents aspects de l'**analyse statique** de programmes, aussi bien sur le plan des fondements (spécification d'analyses à partir de règles d'inférence) que des applications (détection des pointeurs pendants pour l'aide à la mise au point de programmes C, analyse de flot de données et de contrôle dans des programmes Java et Java Card, analyse de protocoles cryptographiques) et à la mise en oeuvre d'analyses statiques par des techniques de résolution itérative de systèmes d'équations et de réécriture d'automates d'arbres.

Pour faciliter l'**analyse dynamique** de programmes, nous développons un outil d'analyse de traces d'exécution permettant à l'utilisateur d'exprimer des requêtes dans un langage de programmation logique. Ces requêtes peuvent être traitées à la volée, ce qui permet d'analyser des traces de grande taille. L'outil peut être utilisé pour le débogage de programmes séquentiels et nous étudions maintenant son application au problème de la détection d'intrusion.

En collaboration avec la société AQL nous poursuivons le développement de l'outil Casting dont le noyau utilise une méthode de **génération de suites de tests**. L'outil peut prendre des entrées dans des formats variés et produit des suites de tests selon des stratégies spécifiées par l'utilisateur. Nous adaptons actuellement Casting pour la génération de suites de test à partir de spécifications UML.

La **sécurité logicielle** constitue un domaine d'application privilégié pour l'équipe-projet. Nous élaborons un cadre pour la définition de propriétés de sécurité et une technique pour leur vérification automatique. Cette technique intègre des techniques d'analyses statiques et de vérification de modèle (" *model checking* "). Ce cadre a été appliqué à la formalisation et la vérification de politiques de sécurité d'applications programmées avec la nouvelle architecutre de sécurité de Java 2 et à la vérification de propriétés de sécurité des cartes à puce multi-applicatives programmées avec le langage Java Card.

## Relations industrielles et internationales