

Ecole Doctorale
Mathématiques et Sciences et Technologies de l'Information et de la Communication

*Laboratoire des Sciences et Techniques de l'Information, de la Communication
et de la Connaissance*

AVIS DE SOUTENANCE DE THESE

Le mercredi 28 novembre 2018 à 10h

au Pôle Numérique Brest Bouguen, amphithéâtre numérique, 6 rue du bouguen, Brest

Monsieur COSTIOU STEVEN

soutiendra une thèse de doctorat sur le sujet suivant :

**Adaptation non-anticipée de comportement : application au déverminage de programmes en
cours d'exécution**

Le jury sera ainsi composé :

- **MME BORNE ISABELLE, Professeur des universités**
Université de Bretagne Sud - VANNES
- **M. DONY CHRISTOPHE, Professeur des universités**
Université Montpellier II - MONTPELLIER
- **M. DUCASSE STEPHANE, Directeur de Recherche**
INRIA Lille - Nord Europe - VILLENEUVE-D'ASCQ
- **M. FABRESSE LUC, Professeur**
IMT Lille Douai - DOUAI
- **M. KERBOEUF MICKAEL, Maître de conférences**
Univ. de Bretagne Occidentale - BREST
- **M. PLANTEC ALAIN, Professeur des universités**
Univ. de Bretagne Occidentale - BREST

A BREST, le 12 novembre 2018
Le Président de l'Université
de Bretagne Occidentale,



M. GALLOU

Titre : Adaptation non-anticipée de comportement : application au déverminage de programmes en cours d'exécution

Mots clés : Déverminage non-anticipé, adaptation de comportement, adaptation dynamique centrée objet

Résumé : Certains programmes doivent fonctionner en continu et ne peuvent pas être interrompu en cas de dysfonctionnement. C'est par exemple le cas de drones en mission, de satellites et de certains objets connectés. Pour de telles applications, le défi est d'identifier les problèmes et de les corriger pendant l'exécution du programme. De plus, dans le contexte des systèmes à objets, il peut être nécessaire d'observer et d'instrumenter individuellement le comportement de certains objets particuliers.

Dans cette thèse, nous proposons une solution d'adaptation dynamique de comportement permettant de déverminer individuellement les objets d'un programme en cours d'exécution. Cette solution est présentée sous la forme d'un patron applicable aux langages objets à typage dynamique. Ce patron permet d'implanter de façon minimale et générique des capacités additionnelles d'adaptation dynamique à granularité objet. Une mise en œuvre de ce patron pour un langage de programmation particulier permet d'instrumenter dynamiquement un programme pour collecter des objets spécifiques et d'adapter leur comportement pendant l'exécution. Nous expérimentons notre patron par des mises en œuvre en Pharo et en Python. Des dévermineurs dédiés à la mise au point de programmes en cours d'exécution sont mis en œuvre pour ces deux langages objet. Ces outils sont évalués pour des cas de déverminage concrets : pour une simulation de drones, pour des applications connectées déployées sur des systèmes cyber-physiques distants, pour un serveur de discussion en ligne ainsi que sur un défaut en production d'un logiciel de génération de documents.

Title: Unanticipated behavior adaptation: application to the debugging of running programs

Keywords: Unanticipated debugging, behavior adaptation, dynamic object-centric adaptation

Abstract: Some programs must run continuously and cannot be interrupted in the event of a malfunction. This is, for example, the case of drones, satellites and some *internet-of-things* applications. For such applications, the challenge is to identify and fix problems while the program is still running. Moreover, in the context of object-oriented systems, it may be necessary to observe and instrument the behavior of very specific objects.

In this thesis, we propose a method to adapt object behavior in a running program. This solution is presented as a pattern applicable to dynamically typed object-oriented languages. This pattern makes it possible to implement, in a minimal and generic way, additional debugging capabilities at the level of objects. An implementation of this pattern for a particular programming language makes it possible to dynamically instrument a program, by collecting specific objects and adapting their behavior during run-time. We experiment this pattern in Pharo and Python implementations with dedicated debuggers for each language. These tools are evaluated on typical debugging case studies: a simulation of drones, connected applications deployed on remote cyber-physical systems, an online discussion server and a debugging session to fix a defect in a production software.