

**HABILITATION A DIRIGER DES RECHERCHES
UNIVERSITE DE LILLE 1 SCIENCES ET TECHNOLOGIES**

N° d'ordre : 42041

NOM/PRENOM DU CANDIDAT : Etien Anne

Ecole doctorale : ED SPI
Laboratoire/Etablissement : CRISTAL Université Lille 1
Discipline : Informatique

JURY :

- Garant de l'habilitation : Stéphane Ducasse
- Rapporteurs : Pr Serge Demeyer, Pr Marianne Huchard, Pr Jurgen Vinju
- Examineurs : Pr Xavier Blanc, Pr Laurence Duchien

SOUTENANCE : 28 juin 2016 à 14h dans l'amphithéâtre Inria (Bâtiment B)

TITRE DE L'HDR :

Apport de la méta-modélisation à l'évolution et au test

RESUME :

Pour beaucoup d'entreprises, leur richesse se trouvent dans leurs logiciels patrimoniaux qui existent souvent depuis plusieurs dizaines d'années et renferment une grande partie de la connaissance de l'entreprise, ses règles de gestions, son savoir faire...Au cours du temps, les besoins auxquels ces logiciels répondent ont évolué, de même que les technologies sur lesquels ils reposent, entraînant des modifications en leur sein même. Ces modifications ayant eu lieu après la livraison du logiciel sont considérées comme de la maintenance. Elles correspondent à plus de 80% du cycle de vie du logiciel et de son coût. Maintenir un logiciel est une activité complexe et nécessaire qui mérite d'être anticipé dès la phase de conception du logiciel. Des phases de remodularisation peut aussi être utiles pour réduire la complexité accumulée par des évolutions successives et fournir de nouvelles bases solides pour de futures évolutions.

Les différents travaux présentés dans ce document répondent à un unique objectif : Concevoir des systèmes de qualité, facilement maintenable et gérer leurs évolutions. La qualité des logiciels peut être assurée et mesurée de nombreuses façons. Dans ce document, nous nous intéressons à la qualité uniquement à travers les tests qui permettent par exemple aux développeurs d'identifier et de localiser des erreurs ou de vérifier après une évolution que les parties inchangées ne sont pas impactées. Enfin, les artefacts de logiciels n'évoluent pas de façon indépendantes. L'évolution de l'un d'entre eux peut avoir des conséquences sur l'un ou plusieurs autres.

Dans ce document, nous nous intéressons à deux types de logiciels, des chaînes de transformations de modèles et des programmes plus traditionnels. Ainsi les chaînes de transformations ne sont pas seulement vu ici comme un moyen de générer du code à partir de modèles via des transformations. Elles sont considérées comme des logiciels à part entière qu'il faudra ensuite maintenir et faire évoluer.

Les résultats présentés dans ce document peuvent se résumer de la façon suivante :

- Proposition d'un nouveau type de transformation, les transformations localisées, introduisant plus de réutilisation, de modularité et de flexibilité dans les chaînes de transformation. Des adaptations dans la définition et la construction des chaînes sont alors nécessaires.
- Identification, définition et automatisation de séquences d'opérateurs spécifiques à un logiciel et utilisées lors de remodularisation d'architecture.
- Adaptation pour les transformations de modèles de mécanismes de localisation d'erreur et de qualification de jeu de test, existant pour les logiciels traditionnels.
- Etude et proposition de solutions de co-évolution de différents artefacts (model-métamodèle, métamodèle-transformation, schéma de base de données-programmes).

Afin de proposer des solutions génériques, indépendantes du langage utilisé dans ces logiciels, ou du domaine d'application, les résultats proposés s'appuient suivant les cas sur la modélisation ou plus fréquemment sur la métamodélisation. La modélisation consiste à représenter sous forme abstraite, par exemple, un logiciel. La métamodélisation s'intéresse aux concepts nécessaires pour représenter ces modèles et aux relations qui peuvent exister entre ces concepts. Nos travaux proposent des solutions originales et ouvrent de nouvelles pistes de recherche comme l'introduction de la généralité dans les chaînes de transformation, transformation de programme d'un langage à un autre ou le débogage d'un code à partir du modèle initial ayant servi à sa génération.