Proposition de recherche doctorale

Résoudre semi-automatiquement les problèmes de couplage

Mots clés :

- **Directeur de thèse** : Mikal Ziane
- **Co-encadrant(s)** :
- **Unité de recherche** : Laboratoire d'informatique de Paris 6
- **Ecole doctorale** : École Doctorale Informatique, Télécommunications, Électronique de Paris
- **Domaine scientifique principal**: Divers

Résumé du projet de recherche (Langue 1)

Parmi les enjeux majeurs du développement logiciel, la prise en compte des changements est un des plus saillants : les logiciels sont trop rigides. En effet, un changement a priori local à un élément logiciel A peut se propager à tous les éléments logiciels qui dépendent de A et ainsi de suite en cascade. Le coût de ces changements peut alors devenir exorbitant. De nombreux patrons de conception [Gamma et al. 94] et plusieurs techniques de refactoring [Fowler et al 99] visent à découpler (isoler) les parties du logiciel qui doivent varier à des rythmes différents. Mais quand et comment appliquer ces patrons et ces transformations de refactoring et surtout est-il possible de le faire automatiquement ? En effet, l'application d'un patron et plus généralement de toute transformation de refactoring ne vise pas, par définition, à ajouter une fonctionnalité à un logiciel mais à en améliorer la qualité. Mais ceci doit être motivé par un objectif de qualité précis car découpler induit typiquement une ou plusieurs indirections qui augmentent la complexité du logiciel remanié. Or l'intention des patrons de conception est décrite de façon très informelle ce qui empêche tout traitement automatique. Heureusement, il est possible de définir rigoureusement une contrainte de couplage et en particulier une partie significative de l'intention d'un grand nombre de patrons de conception [Ziane 00][Ziane et al 03][Ammour et al. 05]. Les violations des contraintes dans un programme ciblent précisément les éléments logiciels sur lesquels appliquer les transformations de refactoring. Il s'agit alors de définir rigoureusement (de façon à pouvoir raisonner dessus) les transformations de refactoring susceptibles de résoudre les violations de contraintes de couplage. Il s'agit aussi définir une stratégie de contrôle de ces transformations pour automatiser autant que possible cette résolution.

Résumé du projet de recherche (Langue 2)

1. Un premier défi consiste à définir rigoureusement les transformations de refactoring susceptible de résoudre les violations de contrainte de couplage puis à les implémenter en garantissant qu'elles préservent le comportement du programme initial. On s'appuiera pour cela sur les résultats prometteurs de [Schäfer et al. 2012]. 2. Le second défi consiste à automatiser autant que possible [Tip et al 2011] la recherche et l'application d'une combinaison de ces transformations pour faire respecter les contraintes de couplage qui sont violées dans un logiciel donné.

Informations complémentaires (Langue 2)

BIBLIOGRAPHIE [Ammour et al 05] S. Ammour, M. Ziane, X. Blanc, S. Chantit "A UML precise specification of design patterns using decoupling constraints", 4th Workshop in Software Model Engineering, WiSME 05, Montego Bay, Jamaica, October 2005. [Fowler et al 99] Martin Fowler, Kent Beck, John Brant, William Opdyke, Don Roberts, Refactoring: Improving the Design of Existing Code, Addison-Wesley, 1999. [Gamma et al. 94] E. Gamma et al, "Design Patterns", Addison-Wesley professional computing series, 1994. [Schäfer et al. 2012] A Comprehensive Approach to Naming and Accessibility in Refactoring Java Programs, Max Schäfer, Andreas Thies, Friedrich Steimann, Frank Tip, IEEE Transactions on Software Engineering, Vol. 38 no. 6, Nov.-Dec. 2012, Pages 1233-1257 [Tip et al 2011] F. Tip, R. M. Fuhrer, A. Kie?zun, M. D. Ernst, I. Balaban, and B. De Sutter, "Refactoring using Type Constraints," ACM Trans. Program. Lang. Syst., vol. 33, no. 3, p. 9, 2011. [Ziane et al. 03] M. Ziane, G. Ardourel, M. Huchard, and S. Chantit, "Formalizing the Quality Constraints of Design Patterns" WEAR '03 Workshop on Encapsulation and Access Rights of the OOIS '03 conference, 2003. [Ziane 00] M. Ziane, "A Transformational Viewpoint on Design Patterns", IEEE International Conference on Automated Software Engineering (ASE), Grenoble, Sep. 2000 pp 273-276, published by IEEE Computer Society.