Matrice de dépendances enrichie

Jannik Laval, Alexandre Bergel, Stéphane Ducasse, Romain Piers

RMoD Team, INRIA - Lille Nord Europe - USTL - CNRS UMR 8022, Lille, France
firstname.lastname@inria.fr

Résumé. Les matrices de dépendance (DSM - Dependency Structure Matrix), développées dans le cadre de l'optimisation de processus, ont fait leurs preuves pour identifier les dépendances logicielles entre des packages ou des sous-systèmes. Il existe plusieurs algorithmes pour structurer une matrice de façon à ce qu'elle reflète l'architecture des éléments analysés et mette en évidence des cycles entre les sous-systèmes. Cependant, les implémentations de matrices de dépendance existantes manquent d'informations importantes pour apporter une réelle aide au travail de réingénierie. Par exemple, le poids des relations qui posent problème ainsi que leur type ne sont pas clairement présentés. Ou encore, des cycles indépendants sont fusionnés. Il est également difficile d'obtenir une visualisation centrée sur un package. Dans ce papier, nous améliorons les matrices de dépendance en ajoutant des informations sur (i) le type de références, (ii) le nombre d'entités référençantes, (iii) le nombre d'entités référencées. Nous distinguons également les cycles indépendants. Ce travail a été implémenté dans l'environnement de réingénierie open-source *Moose*. Il a été appliqué à des études de cas complexes comme le framework *Morphic UI* contenu dans les environnements Smalltalk open-source *Squeak* et *Pharo*. Les résultats obtenus ont été appliqués dans l'environnement de programmation *Pharo* et ont mené à des améliorations.

1 Introduction

Comprendre la structure de grosses applications est un défi mais aussi une tâche importante pour la maintenance (Demeyer et al., 2002). Plusieurs approches présentent des informations à propos des packages et de leurs relations, par la visualisation d'artefacts logiciels, de leur structure, de leur évolution ou encore de métriques. Les métriques logicielles peuvent être difficiles à comprendre puisqu'elles dépendent fortement de l'application analysée. Distribution Map (Ducasse et al., 2006) résoud ce problème en montrant comment les propriétés se répartissent dans une application. Langelier et al. (2005) caractérisent l'évolution des packages et de leurs métriques. Package Surface Blueprint (Ducasse et al., 2007) révèle la structure interne d'un package et les relations avec les autres packages — le concept de surface représente les relations entre le package analysé et les packages auxquels il accède. Dong et Godfrey (2007) ont travaillé sur la présentation de systèmes par des graphes de dépendances objets de haut niveau pour aider à la compréhension des systèmes au niveau de leurs structures de packages.