

Maintenance logicielle par observation

Encadrement

- [Xavier Blanc](#), Professeur à l'Université de Bordeaux
- [Jean-Rémy Falleri](#), Maître de Conférences à l'Enseirb-Matmeca

Contexte

La maintenance est la phase la plus coûteuse dans le cycle de vie d'un logiciel. Une partie significative de cette phase, la maintenance évolutive, consiste à maintenir un logiciel à jour avec les changements de son environnement. En effet, les logiciels dépendent de nombreux composants externes (bibliothèques, services web, services REST). Comme ces composants externes évoluent de manière non contrôlée, il faut être capable de réagir rapidement à leurs changements, afin de disposer de logiciels qui tirent le mieux parti de leur environnement.

Sujet

De nombreuses approches ont été introduites pour faciliter la maintenance évolutive d'un logiciel vis à vis de ses dépendances avec des composants externes. Ces approches prennent en entrée deux versions des composants externes (l'actuelle dont dépend le logiciel, et la nouvelle qui vient d'être mise à jour) et produisent en sortie des correspondances entre les fonctionnalités de l'ancienne et de la nouvelle version. Toutes ces approches sont basées sur des techniques qui mesurent les similarités entre les noms et la structure des fonctionnalités. Cependant, il a été montré que ces techniques sont loin de fonctionner dans tous les cas et notamment sur des données réelles issues des logiciels open-source.

Dans cette thèse, nous nous intéressons à une approche orthogonale. Nous souhaitons évaluer si il est faisable de faciliter la maintenance évolutive d'un logiciel en observant la façon dont d'autres logiciels ont réagis aux évolutions des composants dont ils dépendent. Cette méthodologie induit deux sous questions qu'il faudra traiter:

1. *Où trouver des exemples de maintenance évolutive de dépendances vers des composants externes?*
2. *Comment extraire de ces exemples les changements précis qu'il est nécessaire d'effectuer?*

Travail demandé

Le travail du doctorant consistera à répondre à ces deux questions. Comme cas d'étude, nous utiliserons le problème de la mise à jour de librairies logicielles du domaine du Web, en JavaScript. Comme source de données nous utiliserons la plate-forme d'hébergement de code source GitHub. Nous choisissons ce contexte car:

- Le langage JavaScript est dynamiquement typé. Ainsi il est difficile d'utiliser les méthodes traditionnelles de similarité syntaxique et structurelle.
- GitHub est une plate-forme contenant un nombre gigantesque de projets (plusieurs dizaines de milliers de nouveaux projets sont créés chaque semaine). Il faudra donc être en mesure de pouvoir trouver une manière efficace de traiter cette masse de données.

Le doctorant devra mettre une méthodologie qui permet de trouver des exemples de maintenance évolutive de librairies Web en JavaScript, et d'en extraire les changements précis. Pour cela il devra tout d'abord déterminer un corpus de librairies et de mises à jour. Ensuite il devra élaborer une approche permettant de trouver des exemples correspondant à ce corpus sur GitHub. Enfin il devra proposer une stratégie qui permet d'extraire les changements à effectuer à un fin niveau de granularité. Bien entendu, le doctorant devra valider sa méthodologie sur le corpus des librairies et mises à jour, en terme de qualité des résultats (nombre de faux positifs, nombre de faux négatifs).