

# Utilisation de patches sémantiques dans un cadre distribué à large échelle

Encadrant : Gilles MULLER, projet OBASCO EMN-INRIA, LINA  
Co-encadrants : Esther Pacitti (ATLAS), Yoann Padioleau (OBASCO)  
courriel : `Gilles.Muller@emn.fr`  
tél. : 02 51 85 82 04

## Contexte scientifique

Dans un système d'exploitation moderne, les pilotes de périphériques peuvent représenter jusqu'à 70% du code source. Les pilotes dépendent fortement du reste du système en ce qui concerne de nombreuses fonctions et structures de données définies dans des bibliothèques support et le noyau du système (qui peut être assimilé à une bibliothèque). Ces deux faits induisent un problème significatif relatif aux évolutions ; tout changement dans une interface d'une bibliothèque entraîne de nombreuses répercussions dans les pilotes utilisant ces bibliothèques. Ces répercussions, que nous nommons *évolutions collatérales* peuvent être complexes, favoriser l'apparition de bogues et s'étaler sur de longues périodes.

Nous avons étudié le problème des évolutions collatérales dans le contexte des pilotes de Linux [1, 2]. En particulier, dans le cas d'une des dernières versions de linux (noyau 2.6.13), il existe plus de 150 bibliothèques support dédiées à des familles de périphériques ou des bus, et plus de 2000 fichiers dédiés à des instances de périphériques. Le nombre d'évolutions collatérales qui affectent ces fichiers est impressionnant. En utilisant un outil que nous avons développé, nous avons identifié plus de 1200 évolutions probables dans les interfaces depuis la sortie de Linux 2.6, alors que dans Linux 2.2, ce nombre était inférieur à 300. Ces évolutions ont pu induire des évolutions collatérales dans plus de 400 fichiers, pour plus de 1000 points au sein du code.

Cette étude montre le besoin d'une approche automatique afin d'alléger la tâche de mise à jour des pilotes et d'éviter d'introduire des erreurs. Dans le cadre du projet ANR non thématique Coccinelle, nous travaillons sur une approche reposant sur un langage dédié qui permet à un développeur modifiant une bibliothèque de décrire de manière précise les évolutions collatérales induites par les changements d'interface. Par comparaison avec les patchs qui représentent la solution actuelle pour transmettre les changements dans Linux, nos spécifications sont génériques et peuvent s'appliquer à tous les fichiers impliqués par une évolution collatérale. De fait, une spécification repose sur les sémantiques de l'ancien et du nouveau code, d'où le nom *patch sémantique* pour décrire une évolution collatérale.

Plus précisément, nous développons un cadriciel offrant à la fois un langage dédié, SmPL [3],<sup>1</sup> et un outil de transformation de programmes permettant d'appliquer les patch sémantiques sur les pilotes de périphériques. Dans l'état actuel d'avancement de nos travaux, SmPL nous a permis d'écrire des patches sémantiques pour deux tiers des évolutions collatérales que nous avons étudiées en détail. Nous disposons également d'un premier prototype qui nous permet d'évaluer "en grandeur" l'applicabilité de notre approche. Nos premiers résultats montrent que la taille d'un patch sémantique peut être jusqu'à 343 plus petite que celle des patchs classiques équivalents.

---

<sup>1</sup>SmPL est l'acronyme de "Semantic Patch Language"

## Objectif du stage

Ce stage a pour but d'étudier les problèmes qui se posent lors de l'utilisation systématique de patchs sémantiques dans un cadre distribué à très large échelle comme celui de Linux. On peut citer en particulier, l'accès transparents aux patchs sémantiques dans un cadre pairs à pairs, la détection et gestion des conflits entre patchs sémantiques, la vérification de la compatibilité d'un pilote vis-à-vis de patch sémantiques. Il s'agit d'un sujet volontairement exploratoire qui s'adresse à un étudiant motivé par une thématique à la croisée du domaine des systèmes distribués et des langages de programmation.

## Références

- [1] Yoann Padioleau, Julia Lawall, and Gilles Muller. Understanding collateral evolution in Linux device drivers. In *The first ACM SIGOPS EuroSys conference (EuroSys)*, pages 59–71, Leuven, Belgium, April 2006. previous version also available as INRIA Research Report RR-5769.
- [2] Julia Lawall, Gilles Muller, and Richard Urunuela. Tarantula : Killing driver bugs before they hatch. In *The 4th AOSD Workshop on Aspects, Components, and Patterns for Infrastructure Software (ACP4IS)*, pages 13–18, Chicago, IL, March 2005.
- [3] Yoann Padioleau, Julia Lawall, and Gilles Muller. SmPL : A domain-specific language for specifying collateral evolutions in Linux device drivers. In *International ERCIM Workshop on Software Evolution*, Lille, France, April 2006.