

Optimizing Large Class Smell by Applying Class Normalization Rules

By

Marwan Ahmed Saeed Lardhi

Supervisor

Dr. Saeed Mohammed Baneamoon

ABSTRACT

Software needs to be updated and modified after delivery to correct faults and enhance system quality. Classes of system undergoes continuous modifications and making that source code complex and difficult to maintain. As a result, the class becomes large and called in this case large class smell. This study proposes an effective method for optimizing extraction large class smell using class normalization rules in order to ease maintenance and improve the quality of software by creating new classes with strongly and similarity attributes and shared behavior. The proposed method introduced a technique to extract a class with many responsibilities and that is chosen by the developer or automatically, where is produced an access-set table of attributes, and then is calculated the Jaccard similarity measure to create a similarity matrix for attributes. After that is designed the structural similarity matrix of each extracted class, to calculate the cohesion of each class. Experimental results show that applying the proposed method for dividing the large class into many cohesive classes provide better performance in software development compared with existing methods.

Keywords: Extract Class Refactoring, Large Class smell, Class normalization, Cohesion

تحسين الفئة الكبيرة بتطبيق قواعد تطبيع الفئة

اعداد

مروان احمد سعيد لرضي

اشراف

د. سعيد محمد بانعيمون

الملخص

تحتاج البرمجيات إلى التحديث والتعديل بعد التسليم لتصحيح الأخطاء وتحسين جودة النظام. حيث تخضع فئات النظام للتعديلات المستمرة وتجعل الشفرة المصدرية معقدة ويصعب صيانتها نتيجة لذلك ، تصبح الفئة كبيرة ويطلق عليها في هذه الحالة فئة كبيرة. تقترح هذه الدراسة طريقة فعالة لتحسين استخراج الفئة الكبيرة باستخدام قواعد تطبيع الصف من أجل تسهيل الصيانة وتحسين جودة البرنامج من خلال إنشاء فئات جديدة ذات سمات تشابه قوية وسلوك مشترك. قدمت الطريقة المقترحة تقنية لاستخراج فئة لها العديد من المسؤوليات التي يتم اختيارها من قبل المطور أو تلقائيًا ، حيث يتم إنتاج جدول سمات الوصول ، ومن ثم يتم حساب مقياس تشابه الجاكارد لإنشاء مصفوفة تشابه للسمات (المتغيرات). بعد ذلك يتم تصميم مصفوفة التشابه الهيكلية لكل فئة مستخرجة لحساب تماسك كل فئة. تظهر النتائج التجريبية أن تطبيق الطريقة المقترحة لتقسيم الفئة الكبيرة إلى العديد من الفئات المتماسكة يوفر أداء أفضل في تطور البرمجيات مقارنة بالطرق الحالية.

الكلمات المفتاحية: اعادة بناء الفئة، الفئة الكبيرة، تطبيع الفئة، التماسك.