

IDENTIFIKÁCIA NÁVRHOVÝCH VZOROV STATICKOU ANALÝZOU ZALOŽENOU NA ICH DÔLEŽITÝCH VLASTNOSTIACH

Študijný odbor: 25-35-900 Softvérové inžinierstvo

Autor: Ing. Jaroslav Jakubík

Vedúci diplomovej práce: prof. Ing. Pavol Návrat, PhD.

Január 2009

Cieľom dizertačnej práce bolo realizovať identifikáciu návrhových vzorov statickou analýzou založenou na dôležitých vlastnostiach, navrhnúť a prototypovať zodpovedajúcu metódu.

Dokument prechádza jednotlivými fázami projektu od analýzy rôznych prístupov, cez algoritmy riešiace identifikáciu až k výberu konkrétnej metódy, návrhu, implementácie a testovania jednotlivých rozšírení.

V práci bola pre ďalšie rozšírenie vybraná metóda využívajúca algoritmus hodnotenia podobnosti. Metóda bola rozšírená v podobe viacúrovňového váhovania, neskôr váhovania vo všeobecnosti, filtrovania na základe prítomnosti resp. neprítomnosti vybraných štrukturálnych vlastností a doplnujúcej analýzy v podobe informácií o lexikografickej vzdialenosti názvoslovia návrhového vzoru a identifikovanou inštanciou návrhového vzoru.

Rozšírenia boli experimentálne overené na viac ako 3000 rôznych triedach open source systémov. Na základe overenia boli formulované všeobecné závery vychádzajúce z navrhnutých rozšírení metódy.

IDENTIFICATION OF DESIGN PATTERNS BY STATIC ANALYSIS BASED ON THEIR SIGNIFICANT FEATURES

Degree Course: 25-35-900 Software engineering

Author: Ing. Jaroslav Jakubík

Supervisor: prof. Ing. Pavol Návrat, PhD.

January 2009

The target of this work was to design and prototype specific extensions of a selected method with better and more precise results.

This work documents separate phases of a project - analysis of different design pattern representations, analysis of different algorithms for identification of design patterns in software systems, selection of a concrete method for extensions, design, implementation and tests of an extended method.

In this work, multi level weighting and weighting in general, feature filtering based on selected structural feature presence or absence and additional analysis based on a lexicographical distance between design pattern terminology and identified instance terminology were designed and prototyped..

Extended method was experimentally tested on more than 3000 classes of open source systems from different software engineering areas. Based on the test results, the general conclusion for designed extensions was formulated.