

Posudek disertační práce Ing. Petra Jurnečky

Název práce: Návrhové vzory v paralelních a distribuovaných systémech

Obsah práce

Tématem práce jsou paralelní systémy, které jsou vytvářeny s návrhovými vzory. Cílem práce bylo automatizovat vkládání návrhových vzorů do kódu paralelního programu tam, kde systém zjistí problematické místo, potřebu synchronizace, nebo nechráněný přístup k proměnné více procesy.

Text je na disertační práci poměrně stručný. Má celkem 88 stran a v sedmi kapitolách (mimo úvodu a závěru) představuje výsledky doktorandova konání od počáteční studijní fáze až ke zhodnocení navrženého systému. Text začíná pojednáním o návrhových vzorech a jejich uplatnění v paralelních systémech. Zaměřuje se na takové návrhové vzory, které slouží k řešení obvyklých problémů paralelizmu, jakými jsou synchronizace a výlučný přístup k prostředkům. Díky těmto vzorům by mělo být zabráněno nežádoucím stavům systému jako jsou uváznutí, případně vyhladovění.

Po úvodní kapitole je umístěn rozbor možností analýzy kódu, konkrétně statické analýzy kódu, a je uvedeno několik metod, které jsou v současnosti jsou pro takovou analýzu používány. Třetí kapitola se věnuje návrhovým vzorům pro paralelní programy. Nejprve je poukázáno, že pomocí určitých metrik lze odhadnout, ve kterých místech by bylo vhodné použít některý z návrhových vzorů a dále jsou diskutovány možnosti, jak lze formálně popsat jednotlivé návrhové vzory tak, aby vyhovovaly metodám detekce jejich umístění. Následuje popis patnácti takových vzorů pro paralelní programy a jejich formalizované verze. Po ní se nachází velmi stručná kapitola 4. Ta přináší několik informací o možnostech refaktoringu, tedy automatické modifikaci kódu. Je zde popsáno několik přístupů k refaktoringu objektově orientovaných programů.

Od páté kapitoly je text zaměřen na vlastní přínos doktoranda v oblasti. Teoretická východiska popsaná v kapitole 5 zahrnují jednak analýzu používání zámků během provádění programu a dále zahrnují metriky, na základě kterých by mělo být určováno, kam je vhodné návrhový vzor vložit. Vlastní jádro práce se nachází v kapitole 6, kde je popsáno, jak vytvořit formální model programu a jak na základě tohoto modelu a na základě toho formalizovat zápis návrhového vzoru. Následující kapitola má ukázat, jak lze umožnit automatizované vkládání vzorů do existujících programů a v poslední kapitole před závěrem je předvedeno, jak tento způsob použít u některých ze vzorů.

Zhodnocení

Při čtení textu je patrné, že práce, kterou si toto téma zaslouhuje, mohlo být odvedeno více. Vlastní jádro práce je tvořeno několika metodami, které dohromady mají umožnit určit vhodné nebo žádoucí místo pro umístění některého z návrhových vzorů. Toto jádro by ale evidentně zasloužilo hlubší diskusi, a to jak způsob analýzy kódu, tak propojení analýzy s určováním konkrétního místa pro vložení návrhového vzoru. Z textu a zejména z poslední kapitoly, kde je uváděn příklad, se dá vytušit, že se toto děje na základě dotazů nad vytvořeným modelem programu, ale přesto mám za to, že celý proces měl být popsán důkladněji. Právě jistá nedůslednost ve výkladu vytvořené metody velmi snižuje kvalitu textu a přináší pochyby o celkové kvalitě práce. Fungování výsledného systému je zřejmé až po přečtení závěrečné kapitoly, kde je uvedeno několik stručných příkladů, jak se dotazovat nad vytvořeným modelem a jak na základě takových dotazů určit vhodné místo pro vložení návrhového vzoru. Očekával bych také detailnější experimentální zhodnocení celé práce.

Dále uvedu několik konkrétnějších připomínek k textu.

Některé vyjadřování je nevhodné. Několikrát se v práci doktorand zmiňuje, že využívá predikátovou logiku, a že 'využíváte zejména proměnné, logické spojky, existenční kvantifikátor a predikáty', což mi připadá nevhodné pro tento typ práce. Také tvrzení na straně 43, že predikátový symbol vrací 1, když existuje asociace (relace), není příliš šťastné.

V textu se autor několikrát zmiňuje, že využívá temporální logiku (například na straně 52) k definici chování systému, ale kde a jak ji využívá z textu není patrné. Z podkapitoly 6.2.2, kde by toto mělo být diskutováno, se mnoho nedozvíme.

Na straně 22 se nachází krátká úvaha o tom, že by celý systém šel implementovat v jazyce PROLOG, což lze přijmout, jelikož model systému je postaven na predikátové logice. Dále ale se nedozvíme, proč tedy PROLOG nebyl zvolen pro realizaci popisované metody.

K diskusi navrhuji to, aby doktorand upřesnil strukturu návrhového vzoru. V úvodu kapitoly 6 na straně 49 uvádí, že návrhový vzor je dvojice obsahující vstupní podmínky (preconditions) a že tyto podmínky vychází z analýzy kódu, ale jak tyto podmínky mohou nebo mají vypadat mi není jasné. Dále by mohlo být vysvětleno, v jaké fázi celého procesu jsou tyto podmínky využívány a jak.

Publikační činnost

Doktorand publikoval jádro práce v časopise a na třech konferencích. Toto ukazuje, že jádro práce bylo oponováno a výsledky byly akceptovány v širší vědecké komunitě. To mě také vede k tomu, že i když textová část práce by mohla být kvalitnější, systém vyvinutý doktorandem během jeho doktorského studia je přínosný v oblasti automatického refaktoringu paralelních programů a že student má schopnosti samostatné vědecké práce.

Závěr

Odevzdaná práce splňuje požadavky na tyto texty kladené na spodní hranici. Postrádá zejména důkladnější rozbor navrhovaného řešení. Přesto práci doporučuji k obhajobě s tím, že předpokládám, že se doktorand vypořádá se mnou výše zmíněnými námitkami.

V Brně dne 19.6.2017

doc. Ing. František Zbořil, Ph.D.