

## Posudek oponenta bakalářské práce

Jakub Vaněk

# Generování konfigurací softwarových komponent z modelů vlastností

Práce Jakuba Vaňka se zabývá problematikou automatizovaného generování konfigurací softwarových komponent. Zadání pochází z průmyslové praxe, konkrétně ze společnosti ZF Friedrichshafen AG, Engineering Centre Plzeň, a jeho cílem je generovat zdrojový kód softwarových komponent na základě rozsáhle konfigurovatelného feature modelu.

V průvodním dokumentu věnuje autor z celkem 55 stran původního textu zhruba 8 stran teorii modelování vlastností, přičemž značnou část tohoto rozsahu zabírají triviální příklady, které neilustrují ty skutečně podstatné problémy - ty v rozboru zmíněny nejsou vůbec (např. obtížný úkol identifikace kolizních množin hodnot vlastností).

Dále je zhruba 10 stran věnováno popisu modelovacích nástrojů, přičemž je na str. 13 v tab. 3.1 uveden odhad času vývoje vlastního modelovacího nástroje, který považuji za *velmi* nerealistický. Následně autor na 8 stranách uvádí (opět velmi stručně) teorii generativního programování, přičemž stejně jako v případě modelování vlastností se věnuje spíše méně podstatným teoretickým poznatkům.

Zbytek rozsahu díla je věnován popisu implementace předestřené řešení, který je poměrně důkladný.

Programové řešení generátoru CSV souborů s modely vlastností má zhruba 1200 řádek kódu v jazyce Java - nakolik to mohu jako pouze pasivní uživatel tohoto jazyka posoudit, jedná se většinou pouze o zapouzdřující kód či kód generovaný vývojovým prostředím. Kromě zdrojových souborů `FeatureSelector.java` (715 řádek) a `GrammarElement.java` (180 řádek) tedy v podstatě nejde o obsah posouditelný z kvalitativního hlediska.

Zápis zdrojového kódu odpovídá zvyklostem, je dobře čitelný, avšak velmi málo komentovaný. Rozsah programátorské práce odvedené při implementaci však nepovažuji za odpovídající bakalářské práci - objemem kódu jde spíše o práci semestrální, a ještě menšího rozsahu. Nezpochybňuji ovšem rozsah práce analytické při zkoumání vlastností modelovacích nástrojů a dalších podpůrných článků navrženého toolchainu.

Implementovaný nástroj není zřejmě vhodným řešením zadaného problému, což ovšem autor otevřeně přiznává (a to je třeba ocenit). Tento závěr je ovšem bohužel spíše výsledkem implementačního pokusu než správné, dostatečně rozsáhlé a hluboké analýzy.

Průvodní dokument bakalářské práce je vysázen v  $\text{TeXu}$ , takže typografické či estetické stránce nelze nic vytknout. Překlepy či gramatické chyby se v textu prakticky nevyskytují, autorovo vyjadřování je úsporné, ale srozumitelné. Text je dobře čitelný a vhodně strukturovaný. Obrázků, tabulek, grafů a ukázek kódu bych ovšem s ohledem na řešenou problematiku očekával více.

Z technických závad sazby bíjí do očí jen neslabičné předložky zanechané mácešsky na koncích řádek a také - pro mne zcela nepochopitelné - rastrové obrázky ve velmi nízkém rozlišení (tj. masivně znehodnocené artefakty a rastrování), obr. č. 5.1 až 5.9, na nichž je ovšem *text*, zdrojový kód, a tedy mohly být provedeny prostředím `listings` v  $\text{LaTeXu}$ .

Autor cituje celkem 11 zdrojů, které jsou k předloženému tématu relevantní. Nejsem si ovšem jist, zda uvedený výběr pokrývá požadavek prvního bodu zadání, který požaduje, aby se student seznámil s problematikou „modelování vlastností softwaru (feature modeling) a generativního programování“, neboť ta je natolik široká, že 2 knižní publikace (navíc od stejného autora) ji těžkou molinou dostatečně postihnout.

Provedení citací v textu je v pořádku a ve shodě se zavedenými zvyklostmi a požadavky.

Zadání není splněno zcela: Bod 4 („Navrhněte a implementujte uživatelsky komfortní generátor konfigurace komponent.“) dle mého názoru splněn nebyl, neboť výsledkem práce studenta je konzolová aplikace, která žádnou interaktivitu během své činnosti nevykazuje a ovládá se pouze parametry na příkazové řádce (vstupní soubor, mód činnosti, výstupní soubor) - o uživatelském komfortu tedy nelze hovořit. Tato aplikace navíc také de facto není požadovaným generátorem konfigurace komponent, neboť tuto úlohu plní v autorem připraveném toolchainu off-the-shelf nástroj `Pure::Variants`.

Bod 5 zcela jistě (dle textu práce a dostupného obsahu příloženého CD) splněn nebyl - žádné uživatelské testování ani jeho výsledky nejsou nikde v práci uvedeny.

Také zhodnocení dosažených výsledků (bod 6) je velmi stručné a omezuje se jen na několik všeobecných konstatování v části Závěr, str. 55. K autorově cti ovšem slouží, že otevřeně přiznal, že předložené řešení není pro praktické nasazení ani vhodné, ani dobře použitelné.

Práce je dle mého názoru na samé hranici přijatelnosti jako práce bakalářská, uznáme-li rozsah a náročnost provedené analytické činnosti za podstatný (což ale není dokumentem nezpochybnitelně doloženo). Přikláním se však k názoru, že autor zřejmě měl snahu zadaný úkol systematicky řešit a předložené řešení považoval za optimální. Práci proto s velkými výhradami ještě **doporučuji k obhajobě** a hodnotím klasifikačním stupněm

„dobře“.

**Ing. Kamil Ekštejn, Ph.D.**  
KIV FAV ZČU

V Plzni dne 13. srpna 2019

#### **Doplňující otázky:**

1. Řešíte nějak možné kolize (vzájemné vyloučení) některých hodnot konfigurovatelných vlastností? Pokud ano, jakým způsobem?
2. Jakým způsobem (podle čeho dohodou subjektivních hodnotitelů nebo jinak?) jste určil bodové ohodnocení jednotlivých požadovaných vlastností zkoumaných nástrojů, uvedené v tab. 3.3 na str. 19? Je v pořádku, aby váha ceny software byla stejná jako např. nemožnost importovat datový soubor v požadovaném formátu?