

Posudek oponenta diplomové práce

Autor/autorka práce: **Miroslav Krýsl**

Název práce: **Framework pro analýzu závislostí softwarových artefaktů**

Cílem práce bylo umožnit (iterativní) zpracování a analýzy závislostí větších datových sad softwarových komponent, v návaznosti na předchozí diplomovou práci Milana Hotovce (obhájeno 2022). Výsledkem je upravený datový model aplikovaný na data nově ukládaná v grafové databázi ArangoDB, pro niž vznikla implementace datové vrstvy v Java technologii a ověřovací reimplementace analýz kompatibility Java balíčků (JAR souborů), a text práce dokumentující zejména výběr zvoleného řešení a popis jeho klíčových částí.

Kvalita řešení a dosažených výsledků

Zadání práce je poměrně volné, protože práce má z velké zčásti analytický charakter, ale ani v úvodu práce ani v kapitolách 5 a 6 nejsou cíle práce zpřesněny. Přínosy grafové databáze a varianty vhodných řešení jsou v kap. 6 rozebírány velmi pečlivě a podrobně, nicméně jen obecně – pro daný konkrétní účel není přínos analyticky vyhodnocen. Požadavek na výběr databázového řešení „must support processing [of] large amounts of data“ (s.55) není odůvodněn a zpřesněn. Z rozboru v částech 6.3 a 6.5 navíc plyne, že autor zdůvodňuje použití databáze (pro předávání dat mezi nástroji) nutností vyhnout se natažení kompletního datového modelu do paměti – argumentaci považuji za chybnou, nejde o klíčový důvod, slučuje nezávislé aspekty práce s daty v systému a navíc zřejmě vychází z nepochopení způsobu implementace konceptu rour v UNIX-like systémech.

V práci postrádám (kap. 8) důležité experimenty ověřující přínos použití zvolené grafové databáze (např. porovnání výkonnosti apod. vytvořené implementace s parametry předchozí DP, vyhodnocení práce s velkými daty typu OSGi implementace Felix při vyhodnocování kompatibility). Popsané testy ověřily v zásadě pouze fakt, že vytvořená implementace je funkčně a výkonově ekvivalentní výsledku předchozí diplomové práce; neukazují ale (v rozporu s tvrzením v části 8.3) její vhodnost pro pokročilejší úlohy „s důrazem na podporu zpracování rozsáhlých datových sad a podporu vývoje experimentů v různých programovacích jazycích“ (bod 3 zadání).

Rozsah vytvořeného řešení je poměrně malý (implementovaný rámec a nástroj zahrnuje 94 Java tříd s přímočarou funkcionalitou, jedno JSON schema s 5 entitami, a z něj vygenerované kostry datových objektů). Vzhledem k cíli práce bych očekával vytvoření např. klienta pro alespoň jeden další jazykový ekosystém (např. Python nebo .NET) a/nebo reimplementaci aspoň některých parserů/analyzátorů z DP M.Hotovce pro ArangoDB backend. V příloženém archivu ani v textu práce navíc není návod na instalaci a zprovoznění zvolené grafové databáze, resp. implementovaného řešení, takže výsledek praktické části práce není možné samostatně ověřit.

Formální úroveň

Text práce je psán v anglickém jazyce, na slušné (byť ne vysoké) úrovni, s dobrou strukturou a srozumitelností. Jen zřídka se vyskytují překlepy apod., nicméně systematické uvádění jména (vedoucího práce) „Daňek“ je zarážející. Obr. 4.2 není čitelný (velikost písma), některé příklady kódu

mají nevhodnou délku či umístění (str. 34, str. 41). V kapitole 7 popisující implementaci postrádám architektonický přehled řešení vč. vhodných diagramů, schemat a specifikací.

Použité zdroje odpovídají charakteru práce, zahrnují i publikace z akademického výzkumu. Časté jsou ale případy chybného uvádění odkazů na citace za koncem věty a bez předchozí mezery (např. „and processing time.[8]“ – str. 26).

Závěrem konstatuji, že práce v zásadě splnila zadání, ale pouze v minimalistické podobě a bez vyhodnocení skutečných přínosů vytvořeného řešení. Navrhuji hodnocení známkou **dobře** a práci doporučuji k obhajobě.

Dotazy k práci

1. V části 6.3 (obr. 6.3) je navrženo předávání referencí na uložená data, namísto datových souborů samotných. Jakým způsobem je zaručeno, že experimenty konané v různých časech a místech budou sdílet správná data, tj. jak je řešena časová a prostorová unikátnost těchto referencí a jednoznačnost vazby mezi referencí a datovým souborem?
2. Příklad u popisu importéru (část 7.2.1) je velmi jednoduchý. Jak je řešena situace, kdy je potřeba importovat 2 sady dat, které popisují disjunktní množiny objektů (např. z různých zdrojových úložišť) ale shodou okolností obsahují stejné hodnoty „uid“ atributů?
3. Proč je v tab. 8.2 uváděna velikost komponent v kB, a jaký vztah má tato metrika k danému typu jejich zpracování (import tj. extrakce struktury komponent, vyhodnocení kompatibility)?

V Plzni 3.8.2023

Doc. Ing. Přemysl Brada, MSc. Ph.D.