

Oponentský posudek diplomové práce

David Švamberk: Klient pro aktualizace aplikací úložiště komponent

Cílem diplomové práce bylo analyzovat možnosti vzdáleného přístupu a využití služeb úložiště komponent CRCE (i OSGi aplikací obecně) a na základě výsledků analýzy vytvořit klientskou aplikaci, jež by z tohoto úložiště umožnila aktualizaci komponent v klientském zařízení.

Součástí práce je několik rozmanitých srovnání: dostupné API a knihovny pro vývoj REST služeb a klientů, srovnání úložišť softwarových artefaktů (Maven, OBR, CRCE), dále pak rozsáhlé porovnání mnoha klientských aplikací pro instalaci a aktualizaci softwarových modulů včetně např. i balíčkovacích systémů linuxových operačních systémů, a nakonec důkladné změření časové a paměťové náročnosti dvou knihoven pro převod XML formátu do objektové reprezentace (a naopak) v závislosti na množství a velikosti převáděných souborů. Dále autor popisuje stávající architekturu aplikace GUICA a REST rozhraní úložiště CRCE, analyzuje požadavky a dokumentuje architekturu a řešení vlastní implementace.

Výsledná implementovaná aplikace je rozšířením existující aplikace GUICA o podporu vzdáleného úložiště CRCE s využitím REST jako komunikačního rozhraní. Aplikace byla otestována s vlastní sadou vzorových komponent a je funkční – z úložiště CRCE je možné instalovat a aktualizovat jednotlivé komponenty. Zdrojový kód je na velmi dobré úrovni. Výtku bych měl pouze k umístění tříd souvisejících s podporou CRCE do vyhrazeného javovského balíku ve *stávajícím* modulu s ostatními implementacemi podpory úložišť. Mnohem vhodnější by bylo modul dekomponovat, umístit každou implementaci do separátního modulu a umožnit tak instalaci pouze požadované implementace, obzvláště na zařízeních s omezenými HW prostředky. Toto je ale spíše problém architektury aplikace GUICA jako celku – jednotlivé implementace úložišť jsou namísto generického přístupu napevno zakódovány a instancovány v hlavní GUI třídě, čehož důsledkem se jedná o de facto monolitickou aplikaci, byť s využitím komponent. Přepřeprogramování aplikace směrem k lepší modularizaci je námětem na další rozvoj tohoto nástroje.

Text práce je velmi dobře srozumitelný a čtivý, uspořádání kapitol je přehledné. Text neobsahuje žádné faktické chyby a nepřesnosti – až na velmi drobné výjimky, které jsou zřejmým důsledkem nevhodné formulace, nikoliv nepochopením dané látky. Z typografických prohrěšků bych vytknul nevýrazné oddělování odstavců souvislého textu (doporučil bych používat odsazení prvního řádku nebo svislé odsazení odstavce), dále časté používání spojovníku namísto pomlčky a chybějící interpunkci ve výčtech.

Navrhoval bych, aby diplomant v rámci obhajoby blíže vysvětlil:

1. Jaký je fundamentální rozdíl mezi speciálními úložišti typu OBR či CRCE v porovnání s ostatními konvenčními systémy pro správu závislostí, jako např. Maven nebo aplikační balíčky linuxových distribucí – konkrétně ve způsobu deklarování závislostí jednoho artefaktu na jiném, především z pohledu *jednoznačnosti* takové deklarace?
2. Bylo by možné (a jakým způsobem) využít tohoto specifického přístupu k deklarování závislostí v aplikaci GUICA k aktualizaci jednotlivých komponent nebo i celých aplikací s důrazem na vzájemnou kompatibilitu komponent?

Všechny body zadání jsou splněny, autor v práci prokazuje jak schopnost navázat na předchozí diplomové práce, tak i navrhnout vlastní řešení a taktéž analyzovat a s ohledem na možná rizika vhodně zvolit podpůrné knihovny. Práci **doporučuji k obhajobě** a navrhuji známku **výborně**.

V Plzni, dne 25. 8. 2014


Ing. Jiří Kučera

SOUHLASÍ
S ORIGINÁLEM

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
katedra informatiky a výpočetní techniky