Oponentní posudek

doktorské disertační práce

Vývoj spolehlivého a efektivního softwaru v dynamicky typovaných jazycích

doktoranda Ing. Marka Pašky

Mým úkolem je posoudit disertační práci, která se věnuje problematice tvorby softwarových aplikací, které jsou specificky vytvářeny pro tzv. vestavěné systémy a u nichž při uvolňujících se časových (případně paměťových) restrikcích hraje v současné době dominující roli efektivnost jejich vytváření a spolehlivost výsledku. Autor práce charakterizoval své cíle následujícím výčtem:

- Výběr programovacího jazyka vyšší úrovně, který umožňuje použití takových paradigmat softwarového inženýrství, jako jsou *aspektově orientované programování* (AOP) nebo *doménově specifické jazyky* (DSL). Současně s tím také výběr vývojových nástrojů, které zajistí překlad kódu vyšší úrovně do efektivnějšího strojového kódu.
- Návrhová a verifikační procedura zahrnující záruky správnosti cílového kódu. Tato procedura musí být navíc snadno použitelná.
- Prokázat, že generovaný strojový kód je vhodný pro použití ve vestavěných systémech z hlediska paměťových a časových nároků.

Svůj posudek budu strukturovat v duchu poskytnutých pokynů.

Význam disertační práce pro obor

Využívání výpočetní techniky se stále více rozšiřuje od samostatných případně v síti propojených výpočetních systémů do oblasti vestavěných systémů, pro které tvoří výpočetní technika jen doplňující složku větších systémů určenou např. pro vnitřní řízení nebo snazší vnější ovládání funkcí, které mají takové systémy poskytovat. Dříve nepředstavitelná miniaturizace procesorů, pamětí a dalších počítačových komponent doprovázená klesajícími nároky na napájení dává konstruktérům automobilů, praček, mobilních telefonů, MP3 přehrávačů a dalších druhů výrobků nové možnosti, jak zdokonalit nebo rozšířit funkcionalitu těchto výrobků, a tak je učinit atraktivnější pro potenciální kupce.

Tento trend vyvolává zvýšenou poptávku po tvorbě specifických aplikací, na které se již díky zlepšeným hardwarovým možnostem vestavěných systémů nekladou tak přísné paměťové nároky. Na druhé straně se však požaduje celkové zkrácení vývojové části životního cyklu takových aplikací a dodržení potřebných real-time charakteristik spočívajících především v záruce dosažení požadované doby odezvy a současně i spolehlivosti všech poskytovaných funkcí.

Posuzovaná disertační práce se věnuje všem těmto potřebným aspektům. Jako vhodnou třídu vyšších programovacích jazyků pro tyto aplikace uvažuje především jazyky, které označuje jako *dynamicky typované* (v práci zastoupené především jazykem Python), nicméně uvažují se rovněž jazyky jako Java, C++ nebo C. Výběr vývojového prostředí je určen především požadavkem, aby v něm bylo možné vytvářet více-vláknové programy. Disertace se zaměřuje na využití cílené kombinace různých prostředí s ohledem na to, jaké jsou v nich k dispozici doplňující prostředky sloužící především testování programů, případně jejich formální verifikaci. Práce obsahuje i benchmarkové testy, které dovolují srovnat různé varianty přístupu, a dvě případové studie ukazující podrobně jednotlivé kroky navrhovaného vývojového procesu.

Domnívám se, že řešená problematika je aktuální a hraje v rámci oboru důležitou roli. Snaha o účelnou kombinaci různých nástrojů s jejich případným doplněním o potřebné vlastnosti je hlavní charakteristikou posuzované disertace, a výsledky dosažené právě v této oblasti lze považovat za hlavní přínos.

Postup řešení, použité metody a splnění cílů

Disertační práce má celkem 187 stran textu (včetně obsahu a příloh), vlastní text se člení do 11 kapitol. První čtyři kapitoly přinášejí úvod do řešené problematiky, až v kapitole 5 se poprvé prezentuje základní popis navrženého postupu pro vývoj aplikací.

V následující šesté kapitole popisuje autor podrobně proces kompilace v systému PyPy, neboť součástí řešení jeho disertačního úkolu bylo rozšíření tohoto kompilátoru. Jako prostředky popisu se používají tzv. graf toků, abstraktní interpretace, typová inference a grafové transformace. Popisují se rovněž rozdíly při generování cílového kódu v jazyku C nebo v Java byte-kódu. Tuto kapitolu je možné jak pro její rozsah, tak především pro její obsah za klíčovou část celé disertační práce

Kapitola 7 se věnuje popisu úprav PyPy kompilátoru, především s ohledem na různé způsoby vytváření více-vláknových programů v jazycích Python, C a Java. V kapitole 8 se pozornost věnuje možnostem použití formálních metod testování programů, případně dokazování jejich správnosti. Kapitola 9 představuje výsledky provedených benchmarkových testů, podle nichž je možné udělat si určitou představu o paměťových a časových parametrech jednotlivých studovaných implementačních variant, které byly založeny jak na autorem navrhované metodě vývoje vycházející z použití jazyka Python, tak na tradičních metodách opírajících se o přímé kódování v jazyce C případně C++. Některé varianty se liší také využitím různých prostředků pro ovládání paměti (garbage collector).

Kapitola 10 obsahuje dvě případové studie – první se věnuje návrhu a ověření systému NVR (network video recorder) pro řízení záznamu z IP kamer, druhá popisuje postup vytvoření jednoduchého FTP klienta. Poslední kapitola 11 shrnuje výsledky disertace a nastiňuje možnosti směřování dalších aktivit autora.

Metody použité v disertaci vcelku odpovídají výchozím cílům. Základ práce se opírá především o paradigma inženýrského postupu použitého v softwarovém inženýrství, kdy se navrhne, použije, testuje a upravuje nově navržená metoda tvorby aplikací. V určité míře se využívá také teoretického paradigmatu spočívajícího např. ve formálním zavedení základních pojmů potřebných pro popis procesu kompilace v systému PyPy, včetně formální syntaxe a sémantiky nebo pravidel typové inference.

Domnívám se, že se autorovi podařilo v potřebné míře naplnit stanovené cíle disertace, určitý dílčí komentář ke splnění prvního a druhého cíle uvedu v dalších části tohoto posudku.

Stanovisko k výsledkům a k původnímu konkrétnímu přínosu

Podle mého názoru přináší disertační práce nové poznatky v oblasti softwarového inženýrství, které spočívají ve vytvoření nové metody vhodné především pro tvorbu aplikací v oblasti vestavěných systémů. Tato metoda je založena na využití jazyka Python a systému PyPy a spočívá v kombinaci a úpravě specifických vývojových a testovacích nástrojů.

V této souvislosti bych ovšem chtěl autorovi vytknout, že v práci nejsou dostatečně zřetelně odděleny (popř. jinak identifikovány) části převzaté od vlastních výsledků autora. To se týká především kapitoly 6, která je nazvána *Analýza PyPy procesu kompilace* – autor uvádí jako hlavní prameny této kapitoly (cizí) technickou zprávu [49] a (zřejmě vlastní?) analýzu zdrojového kódu PyPy kompilátoru. Text kapitoly však (podle mého názoru) obsahuje i vlastní výsledky autora spočívající ve formalizaci definice grafu toků a jeho transformací, abstraktní interpretace, typovou

inferenci, popis vztahu různých typových systémů, zpracování výjimek a ovládání paměti. Uvítal bych, kdyby se autor k této záležitosti vyjádřil při obhajobě.

Rád bych se také ještě vrátil k otázce naplnění cílů práce. Zajímalo by mne, do jaké míry lze považovat výběr řešených bechmarkových testů za reprezentativní pro uvažovanou aplikační oblast vestavěných systémů a do jaké míry lze časové a paměťové výsledky získané nově navrženou metodou považovat za uspokojivé či nadějné. Podle mého názoru by bylo žádoucí do srovnání zahrnout i odhad nebo skutečně změřenou dobu potřebnou na vývoj nějaké aplikace při použití jednotlivých variant. Uvítal bych, kdyby se autor k této záležitosti vyjádřil při obhajobě.

Použitá verifikační procedura by si podle mého názoru rovněž zasloužila trochu více pozornosti. Ponechám stranou otázku, zda je opravdu tak uživatelsky snadná, jak by vyžadovalo naplnění druhého cíle disertace. Jde mi spíše o to, zda způsob jejího použití nevytváří prostor pro vznik chyb tím, že Java Pathfinder testuje Java byte-kód, ale my potřebujeme zaručit správnost generovaného programu v C. Uvítal bych, kdyby se autor k této záležitosti vyjádřil při obhajobě.

Systematičnost, přehlednost, formální a jazyková úroveň práce

Práce je vhodným způsobem strukturována, výklad postupuje logicky a je psán čtivým a srozumitelným stylem. Formální úprava práce odpovídá standardním požadavkům. Jazyková úroveň angličtiny je velmi dobrá, nalezl jsem jen velmi malý počet prohřešků, které není třeba vyjmenovávat. Výjimku udělám u úsměvného použití anglického termínu *objection* (str. 169 dole), který známe z amerických filmových detektivek, ale místo něhož bych doporučil správný termín *objective*.

Publikační činnost autora

Seznam relevantních publikací autora obsahuje celkem 9 položek, z toho dvě představují příspěvky na významných mezinárodních konferencích, jedna položka je článek předaný k publikaci (není uvedeno, kam) a zbývající jsou mezinárodní akce pořádané v České Republice nebo na Slovensku. Kromě prací majících vztah k disertaci uvádí autor ještě 5 dalších publikací prezentovaných opět na konferencích v České Republice nebo na Slovensku s výjimkou jedné, která se konala ve Varšavě.

Výsledné doporučení

Domnívám se, že autor prokázal schopnosti k samostatné vědecké práci a předložená disertace přináší nové výsledky v oblasti Softwarového inženýrství. Předloženou práci

d o p o r u č u j i k o b h a j o b ě.

V Praze dne 18.2.2013

Doc. RNDr. Josef Kolář, CSc.
ČVUT FIT

Review of the Ph.D. Thesis "Development of Dependable and Efficient Software with Dynamically-Typed Languages"

submitted by Ing. Marek Paška

The thesis addresses important topics associated with deployment of dynamically-typed languages for embedded software development. The idea is to bring in this area a design approach comparable with the current state of the art of general software creation aiming at desk-top computer applications. The contribution of the thesis consists in proposing a software development process based on the Python programming language and its compiler PyPy. It enables to create rapid prototypes in Python and translate them to the efficient machine code. Moreover, this development process supports also advanced testing based on formal methods.

After introduction and embedded systems related terminology, the third chapter concisely introduces branded related work. This part together with the next chapter, which covers high-level dynamic language principles applied, deserve appreciation for the brief, correct but readable style of presentation. The rest of the thesis discusses original contributions of the work presented. Chapter 5 identifies the proposed methodology, specifies target platform and, namely, dynamic-language-driven approach including tool-chain selection, target code generation, and proper support for formal verification, explicitly Java Pathfinder. While chapter 6 analyses the PyPy compilation process in detail, chapter 7 describes customization of the PyPy compiler that makes possible to use it not only for translating the PyPy interpreter from Python to C source code, what was the original main purpose, but also for compiling general multi-threaded embedded programs. Chapter 8 introduces testing based on formal methods, chapter 9 discusses benchmarking and chapter 10 presents case studies. The last chapter reviews the reached goals and mentions future work.

I appreciate the work done by the applicant in the field of dynamically-typed languages for embedded software development, which appears novel contribution both from the viewpoint of presented conception and sophisticated implementation elaborated in detail. My minor objection deals with omitting the area of hard real-time applications that create substantial sub-region of embedded software domain. I understand the high-level tools selected cannot fit that area; on the other hand, at least a sketch how to handle the problem concurrently -- even if outside the developed approach -- could help to draw more attention of application programmers. I like both the thesis structure and presentation style with the only exception of some too trivial demo programs.

In conclusion, I can summarize that the thesis is devoted to the very important and topical subject. According to my opinion, the submitted framework represents considerable contribution to the field of embedded software development methods. Moreover, candidate's publication activities fulfill common requirements for the Ph.D. degree. For these reasons I can express my full confidence in the candidate's capabilities for the scientific work.
I therefore recommend the acceptance of this thesis for the fulfillment of candidate's studies.

Brno, December 5, 2012

prof. Ing. Miroslav Švéda, CSc.
Brno University of Technology
Faculty of Information Technology

Question to the applicant:

Can you depict an idea how to deal with a hard real-time requirement while using your approach?