

## Posudek oponenta diplomové práce

**Název:** Presenting Result of Software Model Checker via Debugging Interface  
**Autor:** Tomáš Kohan  
**Oponent:** Pavel Jančík

Analýza chyb nalezených pomocí model checkeru je náročný úkol. Zásadním problémem je způsob, jakým jsou informace o nalezené chybě prezentované vývojáři, který má tuto chybu opravit. Cílem této práce je využít IDE a jeho debuggovací rozhraní pro vizualizaci stavu verifikovaného programu v rámci error trace. Autor v práci rozšířil vývojové prostředí Eclipse o možnost spuštění Java PathFinder (JPF) a debuggování nalezené error trace.

Za podstatný problém považuji, že netriviální část textu prvních tří kapitol není autorovo dílo, ale je (téměř) doslovně převzata z různých zdrojů.

Kapitola 1.1 – některé odstavce jsou převzaty ze stránky

[http://en.wikipedia.org/wiki/Software\\_testing](http://en.wikipedia.org/wiki/Software_testing)

Kapitola 1.2.1 – části textu přímo převzaty ze článku

W. Visser , K. Havelund, G. Brat, S. Park, F. Lerda: Model Checking Programs

Kapitola 2.2 – text převzat ze stránky

[http://javapathfinder.sourceforge.net/Model\\_Checking\\_vs\\_Testing.html](http://javapathfinder.sourceforge.net/Model_Checking_vs_Testing.html)

Kapitola 2.3 – texty převážně převzaty z oficiálních stránek JPF (po stranu 18)

<http://babelfish.arc.nasa.gov/trac/jpf/>

Kapitola 3 – text převzat z prezentace

[http://www.eclipse.org/eclipse/debug/documents/ec\\_2006/debugger\\_Basic-Debug-](http://www.eclipse.org/eclipse/debug/documents/ec_2006/debugger_Basic-Debug-Tutorial.pdf)

[Tutorial.pdf](http://www.eclipse.org/eclipse/debug/documents/ec_2006/debugger_Basic-Debug-Tutorial.pdf)

Implementace je v textu popsána srozumitelně, s vhodnou úrovní abstrakce i detailu. Slabinou textu je zdůvodnění designu aplikace. V rámci textu je dostatečně zdůvodněna pouze volba Eclipse jako použitého IDE. Dále autor kvůli přílišné složitosti zamítá standardní debuggovací infrastrukturu JPDA a bez dalšího zdůvodnění přichází s vlastním protokolem, který je v textové podobě přenášen mezi Eclipse a JPF částí implementace. Text dostatečně popisuje instalaci i způsob použití vytvořeného díla.

Implementace je i přes několik lehce odstranitelných chyb použitelná a splňuje zadání. Její ovládání ovšem nedosahuje komfortu standardního debuggeru. Nedořešenou částí je konfigurace JPF, kterou nelze provést způsobem popsaným v textu práce.

I přes existenci rozšíření eclipse-jpf na oficiálních stránkách JPF, které slouží ke spuštění JPF z Eclipse, není eclipse-jpf v textu práce zmíněno. V rámci obhajoby bych očekával, že autor zdůvodní, proč ve své práci nerozšířil eclipse-jpf o podporu pro debuggování a místo toho vytvořil nový projekt.

Dále bych uvítal, kdyby během obhajoby autor zdůvodnil design komunikačního protokolu a alternativní možnosti, které zvažoval (např. serializace či Java RMI místo textového protokolu, implementaci podmnožiny JPDA).

Vzhledem ke zjištěným nedostatkům nedoporučuji, aby byla práce obhájena.

V Praze dne 11.5.2012

Pavel Jančík