

Posudek bakalářské práce

předložené na Matematicko-fyzikální fakultě
Univerzity Karlovy v Praze

posudek vedoucího

posudek oponenta

Autor/ka: Ján Baláž

Název práce: Interpretér stavových automatů popsaných jazykem SCXML

Studijní program a obor: Informatika, Programování

Rok odevzdání: 2010

Jméno a tituly vedoucího/opponenta: Ing. Lubomír Bulej, Ph.D.

Pracoviště: KDSS MFF UK

	e x c e l e n t n í	o d p o v í d a j í c í	s l a b š í	n e v y h o v u j í c í
Náročnost zadaného tématu		X		
Míra splnění zadání		X		
Rozsah práce		X		
Struktura textové části práce			X	X
Analýza			X	X
Vývojová dokumentace			X	
Uživatelská dokumentace		X	X	
Jazyková a typografická úroveň		X		
Návrh a design implementace			X	
Kvalita zpracování softwarové části			X	
Stabilita aplikace		X		

Nejvýznamnější klady:

Nejzávažnější nedostatky:

Struktura práce. Tok informací není souvislý a dobře uspořádaný, často dochází k míchání informací na různých úrovních abstrakce a z různých částí práce. Viz např. kapitola 2, která obsahuje popis použitých komponent, což je spíše materiál do popisu implementace. Popis SCXML v sekci 2.2 zase obsahuje řadu implementačních detailů spojených s volbou Perlu jakožto implementačního jazyka. V celém textu není jediný příklad automatu zapsaného v SCXML, ani ilustrační obrázek zachycující koncepty klasického zápisu automatů a zápisu pomocí SCXML. Kapitola 3 (Analýza návrhu) zachází do implementačních detailů, které by bylo vhodné nechat na později (konfigurace, nahrávání modulů, apod.)

Analýza. Analýza je málo přehledná a rozložena mezi kapitoly 2 a 3, přičemž v kapitole 2 se jedná spíše o popis SCXML proložený implementačními detaily, zatímco v kapitole 3 se řeší problémy spojené s architekturou výsledného systému, opět proložené implementačními detaily (moduly a jejich načítání, ukončení interpretru, komunikace přes sockety). Zdůvodnění řady rozhodnutí působí vykonstruovaně – v řadě případů jsou proti sobě stavěny zjevně špatné a zjevně rozumné alternativy (které by typicky sloužily jako výchozí varianta), nebo se zdůvodňuje volba Perlu i když je Perl předepsán zadáním. Srovnání práce s jinými implementacemi (i open source) chybí.

Návrh a kvalita zpracování. Návrh působí nedotaženě, např. moduly komunikují pomocí socketů, aby se dosáhlo blokujícího čekání (místo implementace vhodné synchronizačního mechanismu), přestože moduly sdílí adresový prostor (běží v různých vláknech, na což zase Perl není příliš vhodný). Implementované třídy jsou s ohledem na jejich funkci často nevhodně pojmenované (MessagePool, Hybrid), mísí anglické názvy s neanglickými (Automat, Modul, Klavesnica), případně mají více zodpovědností (MessagePool).

Vývojová dokumentace. Z převážné části je vygenerována ze zdrojových souborů, což samo o sobě nevádí – problém je v tom, že tato referenční dokumentace je velmi stručná a útržkovitá (což odráží i kvalitu kódu). Popis metod a tříd vesměs rozepisuje název a neposkytuje detailnější informace o zodpovědnosti třídy, kontraktech metod, apod.

Další poznámky:

Celkově je práce hraniční. Návrh a implementace je však spíše slabší a textová část je slabá (především kvůli uspořádání informací v textu). Závěrem je však nutné podotknout, že práce splňuje zadání, obsahuje uživatelskou i programátorskou dokumentaci a nedá se říct, že by autor nějak zjevně volil cestu nejmenšího odporu.

	v ý b o r n ě	v e l m i d o b ř e	d o b ř e	n e p r o s p ě l / a
Návrh známky			X	

Datum: 16. června 2010

Podpis:

