

Strukturovaný posudek oponenta diplomové práce

Jan Zeman

Modul pro transformaci dat z výpočetní do grafické vrstvy

1. Obsah práce

Předložená diplomová práce ve mne vyvolává zmatení. Po přečtení zadání DP jsem nabyl dojmu, že zadavatel zřejmě disponuje SW, který používá pro řešení nějakého problému prostřednictvím konečně prvkových metod, a že v rámci DP by student měl vyvinout obecný modul, který by dokázal zobrazit aktuální stav řešení (viz body zadání 2 a 3) pomocí některého z existujících dostupných vizualizačních nástrojů (viz bod 1). Dále měl student rozšířit vytvořit modul, který by umožnil ukládání / načítání aktuálního stavu řešení (viz bod 4) a ve spolupráci s prvním modulem umožňovat animaci vývoje řešení (viz bod 5). Řešení mělo být dostatečně obecné, pouze v konečné fázi mělo dojít k integraci do stávající SW zadavatele a provedeno otestování a ověření funkcionality (viz bod 6).

Poté, co jsem si důkladně přečetl předložený text práce, jsem však dospěl k následujícímu. Student nevytvářel žádný obecný modul, ale prováděl rozšíření SW zadavatele, konkrétně jedná se o nástroj pro simulaci chování mělké vody, přičemž tento SW již obsahoval možnost vizualizace řešení. Nechápu tedy, co vedoucí práce body zadání 2-3 přesně zamýšlel, a protože z předloženého textu tuto informaci nelze získat, nedokáži splnění těchto bodů posoudit.

Zřejmě proto také byl stěžejním bodem, který student v práci oslovil, bod zadání 4. Způsob, jakým se student tohoto úkolu zhostil, ale odpovídá úrovni bakalářské práce a nikoliv práce diplomové. Pro zadání takového charakteru bych očekával v textu práce rešerši stávajících přístupů pro tvorbu datového úložiště následované formální specifikací vlastního úložiště. Namísto toho dostávám popis, jak student datové úložiště implementoval, a to včetně poněkud rozvláchného popisu parametrů naimplementovaných metod, který by měl spíše přijít do přílohy. Uvážím-li, že student navíc použil široké okraje a poměrně často prázdné řádky, nabízí se otázka, zda práce svým rozsahem vůbec vyhovuje standardu diplomových prací Katedry informatiky a výpočetní techniky na Fakultě aplikovaných věd, Západočeské univerzity v Plzni.

Práce obecně působí nedotaženým dojmem. Skutečně student jejím řešením strávil 468 hodin (18 kreditů = 468 hodin)? Rovněž se zdá, že práce byla řešena na poslední chvíli (viz např. poznámka na str. 39: „načítání z disku jsme z časových důvodů neimplementovali“, ačkoliv odhaduji, že implementace včetně odladění by nemohla zabrat déle než jeden den).

2. Hodnocení kvality řešení a dosažených výsledků

Předložený text je strukturován do čtyř kapitol: úvod, teoretická část, realizační část a závěr. Úvod začíná jedno odstavcovým popisem dynamiky tekutin, přičemž se zmiňuje také o existenci software NTMSH a ve druhém odstavci si student stanovuje za cíl práce doplnit tento software o úložiště a modul pro vizualizace extrahovaných dat. Pokud někdo čeká bližší uvedení do problematiky, bude zklamán.

Po úvodu následuje teoretická část, která nejprve na třech stranách popisuje velmi stručně teorii dynamiky tekutin, kterou považují za naprosto irelevantní vzhledem k řešenému problému. Totéž platí o kapitole 2.2.2, kde je uvedeno, jak se dynamika tekutin počítá na konečných sítích, a do určité míry lze považovat za zbytečné také popis chování jednotlivých vizualizačních nástrojů (požadavkem bylo prozkoumání způsobů výměny dat). Kapitola 2.5 je zbytečně rozsáhlá a popisuje části původního programu NTMSH, které nejsou z pohledu problému DP klíčové. Rozumnější by mi přišlo tuto kapitolu zkrátit a klíčové věci vypíchnout přímo do úvodu, zejména pak podkapitolku 2.5.8, kde je uveden popis současného stavu a řečeno, co bude třeba v rámci řešení DP pozměnit / přidat.

Teoretická část obsahuje místy neúplná, nepřesná nebo dokonce chybná tvrzení. Například neúplná jsou:

- Na str. 13 chybí u obrázku 5 popis, co je $w_1 - w_k$. Na obrázku jsou w_i nějaké body (jaké?), v okolním textu se popisuje w_i jako skalární váha.
- Na str. 14 je řečeno, že případ, kdy je matice singulární, je ošetřen v programu, aniž by bylo uvedeno, jak je toto ošetřeno (souvisí s numerickou robustností, pro kterou není jednoznačné řešení)

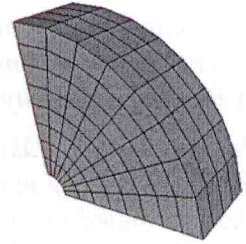
SOUHLASÍ
S ORIGINÁLEM

Phu

- Věta „Spouštění programů trvá 20-30s“ na str. 18 má nulovou informační hodnotu, protože nikde není řečeno, za jakých podmínek bylo tohoto výsledku dosaženo (na jakém počítači, jaké verze programů, v jaké konfiguraci).
- To, že „Jeden modul [MVE-2] umožňuje načítání trojúhelníkové sítě, uložené do formátu VRML“, je sice pravda, ale student se nezmiňuje o modulech pro načítání trojúhelníkové sítě z formátů TRI a STL, které jsou rovněž dostupné.
- Na str. 24 je řečeno, že ParaView umožňuje vygenerovat video do kontejnerů AVI nebo OGG Theora. Chybí uvedení jaký videokodek se používá.
- Tvrzení „Může trvat velmi dlouho, než se zastaví, nezřídka celé dny“ na str. 29 by sneslo zdůvodnění, proč se tak může stát.

Nepřesná jsou:

- Uvedená definice strukturované sítě na str. 11 neplatí pro radiální sítě (viz obrázek vpravo), které jsou rovněž strukturované.
- Definice váženého průměru na str. 13 platí pouze pro sítě ve 2D, zatímco řešení by mělo být obecnější a podporovat rovněž 3D.
- Na str. 24 je uvedeno, že MayaVi2 vyžaduje manuální posun jezdec pro zobrazení sítě v nějakém časovém okamžiku a že toto je významným omezením, protože program musí znovu generovat celý model po každém posunu. Implikaci považuji za nevhodnou. Mohu mít klidně nástroj, kde je rovněž jezdec a přesto se nic nemusí znovu generovat (program si pamatuje mezistavy) stejně tak mohu mít „blbě“ napsanou aplikaci, kde sice jezdec není, ale při změně snímku (automatické) se stejně celý model znovu generuje. Souhlasím s autorem, že jezdec je určitým omezením, ale důvod musí být jiný.



Chybná tvrzení:

- Tvrzení, že „grafické uživatelské rozhraní [VTK] neobsahuje“ (str. 16) není pravdivé. VTK od verze 5.0 obsahuje VTK widgets.
- Strana 18 + 16 uvádí, že VTK formát souborů je textový, a to buď plain text nebo XML a že do budoucna by bylo vhodné použít spíše binární formu. Ale binární forma je již nativně podporována (viz `vtkPolyDataWriter::SetFileTypeToBinary`).
- O větě, že VTK formát představuje standard (viz str. 18), lze polemizovat. Standard v jakém smyslu? ISO ani ECMA standard to není. Modelovací a vizualizační nástroje, které nevyužívají knihovnu VTK, často pracují s formáty STL, PLY nebo OBJ, takže za „standard“ lze spíše považovat STL formát.

Po teoretické části následuje část realizační, kde student představí nejjednodušší možný přístup, který se sám nabízí, pro ukládání mezivýsledků výpočtů nad FEM sítí ať již s dynamickou topologií i geometrií, jen s dynamickou geometrií nebo s pevnou topologií i geometrií. Třebaže diplomant správně poznamenává, že sítě s dynamickou geometrií lze ukládat úsporněji než plně dynamické sítě, tuto myšlenku nerozvíjí a v praxi ukládá sítě s dynamickou geometrií stejně jako plně dynamické sítě, což znamená, že pro každý stav musí uložit souřadnice všech vrcholů, konektivitu sítě a vypočtené hodnoty v uzlech (vrcholech). Výsledný návrh úložiště na mne působí dojmem, že je to něco, na co by měl přijít každý student bakalářského studia během krátké večerní procházky, ale rozhodně ne dojmem, že je to něco, co by se dalo považovat za dobře promyšlený návrh, který vyžadoval několik iterací, srovnání s již existujícími přístupy pro obdobné úlohy, a který je hoden inženýra. Dovedl bych si představit zavedení klíčových a rozdílových mezivýsledků, obdobně jako je tomu při kódování videa, tak, aby pro „Moving“ sítě bylo dosaženo dobrého kompromisu mezi časem načítání a velikostí úložiště. Rovněž tak bych očekával minimálně analýzu možností komprese mezivýsledků, ať již nezávislou na kontextu (viz např. EdgeBreaker techniku od Rossignaca) nebo závislou (viz např. Cuddyac od Váši).

Kapitola 3.4 považuji za implementační detaily, které by měly být uvedeny v příloze. Překvapením pro mne byla rovněž kapitola 3.5, kde jsou popsány unit testy, které student navrhl a naimplementoval, protože unit testy nepovažuji za něco, co bych si představil pod bodem zadání „modul důkladně otestujte a použijte v praxi“. Očekával bych, že student provede ověření modulů na různých datech a důkladně proměří jejich chování z hlediska časové a paměťové složitosti, uživatelské přívětivosti, apod. Jediný experiment spadající do této kategorie testování, který je uveden v kapitole 3.6, nepovažuji za dostačující. Navíc u tohoto experi-

**SOUHLASÍ
S ORIGINÁLEM**

[Handwritten signature]

mentu chybí informace o rychlosti operační paměti a pevného disku, což je vitální informace u všech experimentů, ve kterých je prověřována rychlost I/O operací (serializace mezivýsledků).

Na straně 54 a dále také na straně 57 diplomant připouští, že během testování se ukázalo, že z neznámých důvodů několik věcí nefunguje tak, jak by mělo, např. „transformovaná data se nepodařilo vizualizovat“ a „při obnovení výpočtu od 660. iterace program skončí s výjimkou“. Třebaže oceňuji diplomantovu upřímnost, nemohu se ubránit pocitu, že diplomová práce, která si klade za cíl vytvořit programového vybavení, by neměla být postoupena k obhajobě, dokud platí, že programové vybavení je díky nějakým bugům v praxi nepoužitelné.

Zarážející je tvrzení na str. 58 týkající se možnosti ukládání mezivýsledků v paralelním vlákně, kde je uvedeno: „pokud bychom např. ukládali každou dvacátou iteraci, bylo by dostatek času na to, aby úložiště bylo odložené, než výpočet doběhne do následujícího bodu uložení. Výpočet by však také mohl odkládání předběhnout a musela by se tedy řešit synchronizace vláken.“. Nepřímo z toho vyplývá, že diplomant považuje synchronizaci vláken za něco, čím se nemusí zabývat, pokud bude počet iterací, po kterých má dojít k ukládání, velký.

Zdrojové kódy jsou přehledné a dobře komentovány, nicméně poněkud zarážející je značná duplikace kódu v různých třídách. Domnívám se, že zavedení společné abstraktní třídy by situaci vylepšilo. Nutno však poznamenat, že velká duplikace kódu již byla v původním programu NTMSH, takže diplomant zřejmě stál před rozhodnutím buď překopat zcela jádro aplikace, což však nebylo jeho úkolem, nebo tuto duplikaci kódu ignorovat, což nakonec také udělal, ale to ovšem způsobilo jen další navýšení duplikovaného kódu. Na tomto místě je nutné podotknout, že původní program není nejjednodušší, takže toto rozhodnutí diplomanta, byť jeho důsledky nejsou dobré, bylo v danou chvíli pravděpodobně nejvhodnější.

3. Hodnocení formální stránky předložené práce

Práce se snaží o logické strukturování, nicméně mnoho kapitol a podkapitol je jakoby vytrženo z kontextu, v textu chybí spojovací text, který by z díla učinil něco čitelného a snadno pochopitelného. Jazyková stránka diplomové práce je na velmi dobré úrovni, objevil jsem jen velmi málo překlepů (např. str. 56: „výjmkou“, str. 57: „Neodhlalili“). Práci se nevyhnuly typografické prohřešky, zejména pak přesahy „kódů“ (viz např. str. 49) doleva nebo doprava, což je zřejmě způsobeno autorovo neznalostí Texu, ve kterém je práce vysázena. Za největší problém považuji velmi nízkou kvalitu obrázků (např. obrázky 4, 5 a 14). V obrázku 14 jsou jednotlivé hrany trojúhelníků téměř neznatelné.

Po formální stránce je rovněž považuji za nepřilíš šťastné umístění poděkování a seznamu zkratk za závěr namísto obvyklého umístění vpředu. Pozoruhodné je, že diplomant nepovažuje za podstatné poděkovat kolegyni Hájkové, která byla konzultantem práce. Vzhledem k tomu, že považuje za důležité poděkovat kolegyni Teskové za vysvětlení metody nejmenších čtverců, což je mimochodem něco, co by měl znát již z bakalářského studia, vzbuzuje to otázku, zda diplomant s konzultantem v průběhu řešení práce vůbec jednal, nebo zda přišel až na konci semestru se svázanou diplomovou prací si pro zápočet.

4. Splnění požadavků zadání

Všechny body zadání jsou splněny, ale některé s jistými výhradami. Výčet formátů, kterými se chlubí jednotlivé vizualizační software, nepovažuji za „důkladnou analýzu možných způsobů výměny dat“ (viz bod č. 1). Např. knihovna VTK disponuje možností načíst a uložit STL formát, takže by neměl být problém tento formát použít rovněž v nástroji MayaVi2, třebaže ve výchozím nastavení tohoto nástroje podpora pro STL chybí. Body zadání 2 a 3 nejsem schopen korektně posoudit, protože mne přijde, že tyto body již byly vyřešeny v původním programu. Postrádám rešerši přístupů pro návrh datového úložiště (viz bod 4). Co se týče ostatních bodů zadání, tak k nim mám pouze malé výhrady, které byly již uvedeny výše.

5. Otázky k obhajobě

Během obhajoby by měl diplomant zodpovědět následující otázky:

- Jaký význam má v praxi možnost vrátit se ke k -té iteraci výpočtu, změnit parametry výpočtu a pokračovat v novém výpočtu? Kdy se této možnosti využije při simulacích dynamiky tekutin?
- Načíst 20 000 uzlů za cca 3 sekundy (viz str. 40) považuji za ukázkou neefektivního řešení. Pro 20 000 uzlů totiž mi vychází, že mám 20 K vrcholů, tj. cca 240 KB (souřadnice ve float), 480 KB topologie a

SOUHLASÍ
S ORIGINÁLEM



řekněme 160 KB data. Celkem tedy méně než 1 MB. Při rychlostech současných disků, paměti a procesorů musí efektivní program zvládnout načíst taková data do půl sekundy. Co je příčinou Vašich 3 sekund?

- Proč jste neopravil chybu pro výpočet středového bodu? (viz str. 56)
- Líbí se mi myšlenka, že byste automaticky odkládal úložiště na disk, když Vám bude docházet paměť. Jak navrhujete ale toto udělat? Pokud je mi známo, C# totiž umožňuje a) napsat obsluhu události vyhození `OutOfMemoryException`, která je vyhozena, když dojde paměť, ale jedná se jen o obsluhu události, nikoliv o zpracování výjimky a b) testovat ve vláknech stav garbage collectoru (GC), takže byste mohl provést odložení, když zbývá již jen N bytů volné paměti, ale tento přístup by nezabránil tomu, aby aplikace havarovala, protože někde může být požadováno jednorázové alokování bloku o více než N bytů. Existuje ještě jiný přístup?

6. Závěr

Protože stěžejní část práce, návrh a realizaci datového úložiště, nepovažuji za kvalitně provedený (nulová práce s literaturou, návrh bez řádného promyšlení, implementace zatížená pravděpodobně buggy, otestování jen povrchní prostřednictvím několika jednoduchých unit testů), předloženou diplomovou práci bohužel nemohu doporučit k obhajobě, a proto navrhuji hodnocení známkou **nevyhověl**.

V Plzni dne 21. 5. 2013

Doc. Ing. Josef Kohout, PhD.

KIV-FAV-ZČU

**SOUHLASÍ
S ORIGINÁLEM**



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