

Propojení systému webMathematica s databázovým systémem a demonstrací na aplikaci pro evoluční algoritmy

Linking the webMathematica and database Systems and its
Demonstration on Applications Used for the Evolution Algorithms

Bc. Michal Svátek, DiS.

Diplomová práce
2012



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Michal SVÁTEK, DiS.**
Osobní číslo: **A09722**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**

Téma práce: **Propojení systému webMathematica s databázovým systémem a demonstrací na aplikaci pro evoluční algoritmy**

Zásady pro vypracování:

1. Seznamte se s prostředím webMathematica.
2. Popište kroky integrace s databázovým systémem.
3. Propojte aplikaci evolučních algoritmů v prostředí webMathematica tak, aby bylo možné ukládání dat z výpočtů, nejrůznější možnosti výběrů a zpětné výpočty webMathematicou z uložených dat.
4. Zpracujte návod pro propojení dalších systémů z dalších předmětů.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. ZELINKA, I., OPLATKOVÁ, Z., OŠMERA, P., ŠEDA, M., VČELAŘ, F. Evoluční výpočetní techniky – principy a aplikace. BEN – technická literatura, Praha, 2008, ISBN 80-7300-218-3.
2. ZELINKA, I. Umělá inteligence I. VUT Brno, 1998, ISBN 80-214-1163-5.
3. KVASNIČKA, V., POSPÍCHAL, J., TIŇO, P., Evolučné algoritmy. Bratislava : STU Press, 2000, ISBN 80-227-1377-5.
4. MAŘÍK, V., ŠTĚPÁNKOVÁ, O., LAŽANSKÝ, J.: Umělá inteligence 4., Academia, 2003, ISBN 80-200-1044-0.
5. BERGSTEN H.: JavaServer Pages (3rd Edition ed.), OReilly Media, 2003, ISBN 978-0-596-00563-4.
6. Help sw webMathematica.
7. KOVÁČOVÁ M.: webMathematica 2.2, STU Bratislava, 2007, ISBN 80-969652-1-3.
8. The Mathematica Book, help sw Mathematica.

Vedoucí diplomové práce:

Ing. Zuzana Oplatková, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

24. února 2012

Termín odevzdání diplomové práce:

21. května 2012

Ve Zlíně dne 24. února 2012



prof. Ing. Vladimír Vašek, CSc.
děkan



doc. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

ABSTRAKT

Cílem mé diplomové práce je propojení systému webMathematica s databázovým systémem, popsat ho a vytvořit návrh implementace dalších systémů s webMathematicou. Následně provést propojení webMathematicy s aplikací pro evoluční algoritmy tak, aby bylo možné data z výpočtů ukládat do databáze a tím zajistit i zpětné získání uložených výsledků pro opětovné využití u evolučních algoritmů.

Klíčová slova:

webMathematica, Mathematica, Evoluční algoritmy, Java Server Pages, databáze, MS SQL, MySQL, PostgreSQL

ABSTRACT

This diploma thesis is about connection system of webMathematica with system of database and describe this connection and create plan of combination the others systems with webMathematica. Then make connection webMathematica with application for evolutionary algorithms to will be possible save data from calculations to database and get save results back for use this results again for evolutionary algorithms.

Keywords:

webMathematica, Mathematica, Evolutionary algorithms, Java Server Pages, database, MS SQL, MySQL, PostgreSQL

Zde bych rád upřímně poděkoval svým rodičům a prarodičům za jejich podporu během celé délky studia, díky nimž jsem se stal snad i lepším člověkem.

Další dík patří vedoucí mé diplomové práce Ing. Zuzaně Oplatkové, Ph.D. za vstřícnost, postřehy a pomoc při řešení této práce.

Dále bych také chtěl velice poděkovat mému zaměstnavateli panu RNDr. Ivanu Kuglerovi za možnost studovat při zaměstnání.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 SEZNÁMENÍ S PROSTŘEDÍM WEBMATHEMATICA	11
1.1 WOLFRAM WEBMATHEMATICA	11
1.2 EVOLUČNÍ ALGORITMY	12
1.2.1 Horolezecký algoritmus (hill climbing)	12
1.2.2 Algoritmus simulovaného žíhání (simulated annealing).....	13
1.2.3 Algoritmus rozptylového hledání (scatter search).....	13
1.2.4 Algoritmus diferenciální evoluce	14
1.2.5 SamoOrganizující se Migrační Algoritmus (SOMA)	14
2 POPIS INTEGRACE S DATABÁZOVÝM SYSTÉMEM	15
2.1 DATABASELINK V PROSTŘEDÍ MATHEMATICA	15
2.1.1 DatabaseLink a jeho funkce	15
2.1.2 Technologie	16
2.1.3 Načtení balíčku.....	16
2.1.4 Spojení s databází.....	17
2.1.5 Získávání dat	17
2.1.6 Vkládání dat	18
2.1.7 Aktualizace dat.....	19
2.1.8 Mazání dat.....	20
2.1.9 Dávkové příkazy.....	20
2.1.10 Ukončení	21
2.1.11 Database Explorer	21
2.1.12 Zabezpečení a „autentizace“	21
2.1.13 Secure Socket Layer (SSL).....	22
2.2 KÓDOVÁNÍ A DEKÓDOVÁNÍ POMOCÍ BASE64	23
2.2.1 Příklad	23
2.2.2 Algoritmus base64.....	23
2.2.3 Vlastnosti base64	24
II PRAKTICKÁ ČÁST	25
3 INTEGRACE S APLIKACÍ PRO EVOLUČNÍ ALGORITMY	26
3.1 VÝVOJOVÉ PROSTŘEDÍ	26
3.1.1 PSPad 4.5.6	26
3.1.2 Wolfram Workbench 2.0.....	26
3.1.3 Netbeans 7.1.1	27
3.2 PŘÍPRAVA SERVERU.....	28
3.2.1 JSP server (Apache TomCat)	28
3.2.2 WebMathematica a Mathematica.....	29
3.3 VOLBA DATABÁZE.....	29
3.3.1 Instalace MySQL.....	29
3.3.2 Instalace MS SQL Server 2008 R2	29

3.3.3	Instalace PostgreSQL	31
3.4	WOLFRAM MATHEMATICA	31
3.5	VÝBĚR VHODNÉ METODY PŘIPOJENÍ	31
3.6	PŘEHLED OVLADAČŮ DATABÁZE	31
3.7	PŘIDÁNÍ NOVÉHO OVLADAČE	32
3.7.1	Zjištění výchozího adresáře	32
3.7.2	Vytvoření adresáře pro soubor jar	32
3.7.3	Vytvoření adresáře pro nastavení	33
3.7.4	Export nastavení do souboru	33
3.7.5	Kontrola nainstalování nového ovladače	33
3.8	NAPOJENÍ NA DIFERENCIÁLNÍ EVOLUCI	34
3.8.1	Rozbor uložení dat do databáze	35
3.8.2	ERD diagram pro DE	35
3.8.3	Ukázka dat uložených v databázi	36
4	NÁVOD NA PROPOJENÍ DALŠÍCH SYSTÉMŮ Z DALŠÍCH PŘEDMĚTŮ	37
4.1	ANALÝZA	37
4.2	DOPORUČENÁ VÝVOJOVÁ PROSTŘEDÍ	37
4.3	VYTVOŘENÍ ERD DIAGRAMU	37
4.4	VOLBA DATOVÝCH TYPŮ	37
4.5	KÓDOVÁNÍ BASE64 V MATHEMATICE	38
4.6	VÝBĚR DATOVÝCH TYPŮ DLE TYPU DATABÁZE	38
4.7	PŘÍKLAD UKLÁDÁNÍ OBRÁZKU DO DATABÁZE	38
	ZÁVĚR	41
	ZÁVĚR V ANGLIČTINĚ	42
	SEZNAM POUŽITÉ LITERATURY	43
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	44
	SEZNAM OBRÁZKŮ	46
	SEZNAM TABULEK	47
	SEZNAM PŘÍLOH	48

ÚVOD

Důvodem výběru tohoto tématu diplomové práce byla výzva a hlavně získání nových dovedností a zkušeností, které jsem doposud nikde nenabyl. Líbila se mi myšlenka využívání evolučních algoritmů tam, kde matematická aritmetika není schopna v požadovaných časech získat potřebné výsledky. Zapojení stochastiky a využití evolučních algoritmů při výpočtu reálných úloh je samo o sobě velmi zajímavé.

Při testování aplikace, která využívá evolučních algoritmů, jsem zjistil, že je schopna pouze data poslat jádru Mathematicy a ta následně pomocí webMathematicy jen výsledky zobrazí zpět do prohlížeče. Jelikož některé operace jsou dosti náročné na výpočet, je dle mého názoru škoda je jen zobrazit. Proto jsem se rozhodl nastudovat metodiku propojení webMathematicy s databázovým strojem a v mé diplomové práci tuto metodiku popsat.

V současné době jsem nenalezl dostupnou literaturu k této metodice v češtině. Sepsání této metodiky bude přínosem i pro vývojáře, kteří by rádi vyvíjeli aplikace, ale narazili na jazykovou bariéru či neví jak začít.

Cílem je i vytvoření propojení aplikace pro evoluční algoritmy s databází. Díky tomu vznikne i vedení historie výsledků a bude možno analyzovat získané výsledky a případně je dále využívat k dalšímu výzkumu.

I. TEORETICKÁ ČÁST

1 SEZNÁMENÍ S PROSTŘEDÍM WEBMATHEMATICA

1.1 Wolfram webMathematica

Wolfram webMathematica představuje jeden z neznámějších programových systémů pro provádění numerických a symbolických výpočtů a vizualizací dat. Wolfram Mathematica je výpočetním nástrojem se systémovou technologií, jenž si klade za cíl zajištění spolehlivosti, snadnost použití a výkon. Lze ji použít přímo v infrastruktuře nebo integrovat do samostatné aplikace.¹

Wolfram WebMathematica je chápána jako nadstavba k Wolfram Mathematica, neboť jejím charakteristickým rysem je to, že uživatelé mohou počítat a provádět vizualizace svých dat pouze za pomoci webového prohlížeče, tzn., mohou ji využívat i uživatelé, kteří ji nemají ve svém počítači nainstalovanou a třeba se doposud s Wolfram Mathematica neselekali.

Wolfram Mathematica je vyvíjena již dvacet let a za toto období do ní byla nakomponována spousta nových funkcí, která jsou využívána v různých oblastech (např. věda, finance, statistika, biotechnologie a medicína, atd.).²

Hlavním rozdílem mezi těmito technologiemi je především to, že jsou vytvořeny pro různé typy uživatelů a poskytují rozdílná uživatelská rozhraní. Lze také říci, že pro práci s Wolfram WebMathematica je třeba jen základní znalost HTML a Wolfram Mathematica, aby bylo možné vytvořit plně funkční webovou stránku. Oproti ostatním programům, které vyžadují dostatečnou znalost programování za pomoci JAVA technologie a následně umožňují vytváření pouze malých appletů.

Klíčové výhody Wolfram WebMathematica:

- Možnost Wolfram WebMathematica využívat velkou knihovnu příkazů Wolfram Mathematica, které slouží pro vývoj webových aplikací. Na základě toho lze

¹ Mathematica Features. [online]. [cit. 2012-05-18]. Dostupné z:
<http://www.wolfram.com/mathematica/features/>

² The Mathematica solutions. WOLFRAM RESEARCH, Inc. [online]. [cit. 2012-05-18]. Dostupné z:
<http://www.wolfram.com/solutions/>

vytvářet numerické, symbolické a grafické aplikace snadno a rychle. Za pomoci Wolfram Mathematica lze importovat a exportovat velké množství dat, znaků a obrázků, což umožňuje uživatelům zpracovávat data online.

- Veškeré výpočty jsou prováděny na Wolfram WebMathematica serveru. Proto si koncový uživatel nemusí pořizovat žádný speciální software. Vše co potřebuje je webový prohlížeč a v případě využívání dalších pokročilých funkcí, jako je např. interaktivní 3D grafika, je zapotřebí Java Runtime Environment.
- Všechny prvky uživatelského rozhraní jsou standardní webové GUI, jako posuvníky, textová pole, zaškrtačací políčka či rozevírací seznamy. Ve většině případů nejsou potřebné žádné znalosti Wolfram Mathematica.

1.2 Evoluční algoritmy

Evoluční algoritmy se snaží využít modelů evolučních procesů, aby tak našly řešení náročných a rozsáhlých úloh. Veškeré takové modely mají několik společných rysů:

- pracují zároveň s celou skupinou (množinou) možných řešení zadaného problému místo hledání jednotlivého řešení;
- vygenerovaná řešení postupně vylepšují zařazováním nových řešení, získaných kombinací původních;
- kombinace řešení jsou následovány náhodnými změnami a vyřazováním nevýhodných řešení.

1.2.1 Horolezecký algoritmus (hill climbing)

Odstraňuje problém enumerativních algoritmů, tzn. neprohledává všechny možné kombinace řešení, ale snaží se postupovat ve směru nejlepšího řešení, tedy podle největšího gradientu. Princip tohoto algoritmu je možné přirovnat k chování turistů, kteří se ztratili v horách a snaží se postupovat do nejnižší části údolí, protože ví, že tam je horská chata. Algoritmus pro aktuální řešení vygeneruje okolí sousedních bodů a v tomto okolí hledá nejlepší řešení, minimum nebo maximum. Poté se přesune do nejlepšího řešení a znovu opakuje hledání v okolí nové polohy. Proces se opakuje, dokud neproběhne předem daný počet iterací. Jedná se o gradientní metodu bez výpočtu gradientu. Základní slabinou horolezeckého algoritmu je

stejně jako u jiných gradientních metod riziko uváznutí v lokálním extrému, tedy problém cyklického řešení.

Pravděpodobnost uváznutí v lokálním extrému je možné snížit, avšak ne úplně odstranit, opakováním horolezeckého algoritmu pokaždé z jiného náhodného počátečního řešení. Takto upravená verze se nazývá stochastický horolezecký algoritmus a pro její realizaci je nutné k výše popsanému principu přidat ještě jeden cyklus pro opakování celého algoritmu.

1.2.2 Algoritmus simulovaného žihání (simulated annealing)

Našel inspiraci ve fyzikálním procesu žihání tuhého tělesa, kdy je tuhé těleso zahřáté na vysokou teplotu a postupným pomalým snižováním teploty se odstraňují jeho vnitřní defekty. Tímto chlazením jsou eliminovány nestabilní částice, dochází ke stabilizaci krystalové mřížky a tuhé těleso získává potřebnou kvalitu.

Interpretace v oblasti evolučních algoritmů je taková, že se provádí iterace postupným snižováním teploty. Pro každou teplotu je náhodně vygenerováno nové řešení, které je podrobeno Metropolisovu kritériu, které pro danou teplotu určuje pravděpodobnost nahrazení aktuálního řešení novým.

Algoritmus simulovaného žihání neprohledává okolí, ale nová řešení jsou náhodně generována, což odstraňuje problém uváznutí v lokálním extrému. Důležitý je i plán chlazení, tedy metoda snižování teploty. Teplotu lze snižovat krokově nebo kontinuálně pomocí multiplikátoru. Další variantou algoritmu je zavedení elitismu, kdy se uchovává nejlepší řešení dosažené během činnosti algoritmu. U klasického simulovaného žihání je na začátku algoritmu při vysoké teplotě vysoká pravděpodobnost přijetí horšího řešení a řešení nekonvergují k výslednému řešení, což se mění se snižující teplotou. U simulovaného žihání s elitismem řešení konvergují k výslednému již od začátku algoritmu.

1.2.3 Algoritmus rozptylového hledání (scatter search)

Kombinuje různé způsoby řešení. Je to vektorový algoritmus, který generuje nové řešení na základě pomocných heuristických technik. Rozptylové hledání pracuje s množinou řešení a referenční množinou nejlepších řešení. Lineární kombinací vektorů z referenční množiny vznikne nová množina a z této množiny se vybere podmnožina nejlepších řešení, na které se opět aplikují pomocné heuristické techniky.

1.2.4 Algoritmus diferenciální evoluce

Byl navržen koncem devadesátých let dvacátého století R. Stornem a K. Princem. Diferenciální evoluce (differential evolution) je podobná genetickým algoritmům. Algoritmus probíhá v cyklech, které se nazývají generace, cílem je vyšlechtit co nejlepší populaci, která obsahuje nejlepšího jedince, tedy nejlepší řešení optimalizační úlohy.

1.2.5 SamoOrganizující se Migrační Algoritmus (SOMA)

Vznikl v roce 1999 a je významný tím, že jeho autorem je prof. Ing. Ivan Zelinka, Ph.D. z domovské univerzity. Algoritmus se od klasických evolučních algoritmů odlišuje tím, že noví potomci nevznikají křížením rodičů, ale jeho princip je založen na kooperativním migrování jedinců v prostoru řešení, tedy k prohledávání prostoru řešení. Proto se cykly algoritmu nenazývají generace, ale migrační kola, během kterých se střídají fáze soutěžení a spolupráce. Ve fázi spolupráce si jedinci vyměňují informace o nejlepším řešení a ve fázi soutěže se přesunují směrem k nejlepšímu řešení, přičemž se sami snaží najít nejlepší řešení. Samoorganizace populace spočívá v tom, že se jedinci vzájemně ovlivňují, mohou vnikat skupiny jedinců, které migrují společně, rozpadají se a zase spojují. Dle prof. Zelinky je algoritmus inspirován podobným chováním v přírodě, kdy skupiny jedinců (smečka vlků, kolonie termitů, roj včel) spolupracují při hledání potravy.³

³ MAŇAS, Jaroslav. *Prostředí pro porovnávání evolučních algoritmů v prostředí webMathematica* [online]. Zlín, 2010 [cit. 2012-04-10]. Dostupné z: <http://theses.cz/id/2bgp8k/>. Diplomová práce. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky. Vedoucí práce Ing. Zuzana Oplatková, Ph.D.

2 POPIS INTEGRACE S DATABÁZOVÝM SYSTÉMEM

V následující části je popsán balíček, který je třeba využít pro úspěšnou a bezpečnou práci s databázemi.

2.1 DatabaseLink v prostředí Mathematica

Častým problémem, týkající se velkých společností je především ukládání dat, indexování a následné vyhledávání. Stejně tak, jako se zvyšuje počet členů společnosti, tak se zároveň zvyšují požadavky na práci s daty a jejich správou. Další z problémů, jež v souvislosti s tímto vyvstávají, patří odpovědi na otázky, co daná data představují, jak jsou následně využívána a stejně tak i otázka celkového objemu dat.

Díky masivnímu rozvoji moderních elektronických počítačů v druhé polovině dvacátého století se jako standardními pro správu dat v mnoha organizacích používají nástroje, jako jsou relační databázové systémy (RDBMS) či Structured Query Language (SQL).

V dnešní době se setkáváme s celou řadou databázových systémů. Výčet lze začít u objemných, drahých komerčních aplikací, které jsou vhodné pro špičkové použití volně dostupných open-source nástrojů, které běží na PC s operačními systémy, jako jsou Microsoft Windows nebo Linux.

DatabaseLink je aplikace, která poskytuje soubor nástrojů, které umožňují pohodlnou integraci Wolfram Mathematica s databázovými systémy. Existuje celá řada výhod, které plynou z integrace Wolfram Mathematica a databázového systému. Údaje získané z jedné aplikace jsou uloženy v databázi, následně zde také budou uloženy výpočty a jejich výsledky. Nakonec další aplikace může tyto výsledky extrahovat a může je použít pro další účely.

2.1.1 DatabaseLink a jeho funkce

- pracuje s většinou používaných aplikací SQL,
- HSQL Database Engine (HSQLDB) – lze použít i v případě, že nemáme vlastní databázi,
- zahrnuje ovladače pro podporu databází, včetně MySQL, Open Database Connectivity (ODBC) a HSQLDB,
- rozhraní SQL – není třeba získávat další znalosti nového systému,

- rozhraní Wolfram Mathematica – užitečné pro ty, kteří mají znalosti s aplikací Wolfram Mathematica,
- standardní uživatelské rozhraní GUI.⁴

2.1.2 Technologie

DatabaseLink je založena na běžně používané technologii Java DataBase Connectivity (JDBC). Je zde zároveň široce využívaná technologie Wolfram Mathematica Java toolkit J/link, přesto není vyžadována znalost programování v jazyce Java.

DatabaseLink dává na výběr z řady ovladačů pro databáze, v případě, že není nalezen požadovaný ovladač, lze nainstalovat svůj vlastní. Viz ukázka v praktické části diplomové práce.

DatabaseLink nabízí dva typy rozhraní pro práce s daty:

- rozhraní příkazového řádku, který je flexibilní a je vhodný pro použití databázových příkazů uvnitř programu a
- rozhraní grafické, jehož používání je jednodušší.⁵

2.1.3 Načtení balíčku

DatabaseLink je doplňkovou aplikací Wolfram Mathematica, dříve než lze plně využívat všechny její funkce, je třeba aplikaci načíst následujícím příkazem:

```
In[1]:= Needs["DatabaseLink`"]
```

⁴ DatabaseLink Tutorial: Introduction to DatabaseLink. *Wolfram Research, Inc.*, [online]. [cit. 2012-05-18]. Dostupné z: <http://reference.wolfram.com/mathematica/DatabaseLink/tutorial/Introduction.html>

⁵ DatabaseLink Tutorial: Getting Started - Wolfram Mathematica 8 Documentation. *Wolfram Research Inc.* [online]. [cit. 2012-05-18]. Dostupné z: <http://reference.wolfram.com/mathematica/DatabaseLink/tutorial/GettingStarted.html>

2.1.4 Spojení s databází⁶

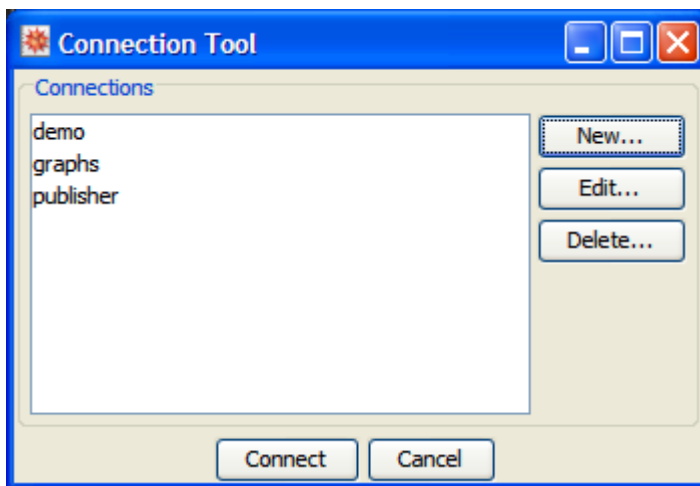
K připojení databáze pomocí příkazového řádku se využívá funkce `OpenSQLConnection[]`. Viz následující ukázka.

```
In[2]:= conn = OpenSQLConnection["demo"]  
Out[2]= SQLConnection[demo, 7, Open, TransactionIsolationLevel -> ReadCommitted]
```

Do databáze je možné se připojit i přes GUI rozhraní, kdy není třeba psát jako parametr funkce `OpenSQLConnection[]` jméno databáze.

```
In[3]:= conn1 = OpenSQLConnection[];
```

Po spuštění výše uvedeného příkazu se otevře Connection tool.



Obrázek 1: Připojení do databáze pomocí GUI – Connection Tool

2.1.5 Získávání dat

Relační databáze se skládá z řady tabulek, každá tabulka obsahuje data v různých kategoriích. Tyto kategorie se nazývají sloupce. Každý řádek tabulky obsahuje datové hodnoty pro jednotlivé sloupce. Databázové aplikace poskytují funkce pro správu těchto dat – jako jsou vytváření, vkládání, aktualizace či mazání.

Pro tyto databáze jsou tabulky zásadní věcí, a proto je zde jednoduchý způsob jak je vytvořit a to za pomoci příkazu `SQLTables` následovně:

⁶ Database Connections: Wolfram Mathematica 8 Documentation. WOLFRAM RESEARCH INC. [online]. [cit. 2012-05-18]. Dostupné z:

<http://reference.wolfram.com/mathematica/DatabaseLink/tutorial/DatabaseConnections.html>

```
In[4]:= SQLTables[conn]
Out[4]= {SQLTable[SAMPLETABLE1, TableType → TABLE]}
```

Informace o konkrétních sloupcích v tabulce lze zobrazit příkazem SQLColumns.

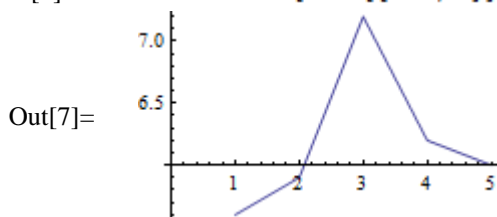
```
In[5]:= SQLColumns[conn, "SAMPLETABLE1"]
Out[5]= {SQLColumn[{SAMPLETABLE1, ENTRY},
  DataTypeName → INTEGER, Nullable → 1, DataLength → Null],
  SQLColumn[{SAMPLETABLE1, VALUE}, DataTypeName → DOUBLE, Nullable → 1,
  DataLength → Null], SQLColumn[{SAMPLETABLE1, NAME},
  DataTypeName → VARCHAR, Nullable → 1, DataLength → 2147483647]}
```

Lze načíst jen určitá data v tabulce za použití příkazu SQLSelect.⁷

```
In[6]:= data = SQLSelect[ conn, "SAMPLETABLE1"]
Out[6]= {{1, 5.6, Day1}, {2, 5.9, Day2}, {3, 7.2, Day3}, {4, 6.2, Day4}, {5, 6., Day5}}
```

Výsledek dotazu do databáze je seznam Mathematica, který může být použit v každém příkazu Mathematica. V následujícím příkladu je znázorněn poslední prvek každého řádku.

```
In[7]:= ListLinePlot[data[[All, 2]]]
```



Následující příklad načte data z tabulky SALES, následně přidá záhlaví sloupců a výsledkem je výstup ve formě tabulky.

```
In[8]:= SQLSelect[ conn, "SAMPLETABLE1", "ShowColumnHeadings" → True] // TableForm
```

```
Out[8]//TableForm=
```

ENTRY	VALUE	NAME
1	5.6	Day1
2	5.9	Day2
3	7.2	Day3
4	6.2	Day4
5	6.	Day5

2.1.6 Vkládání dat

K vložení dat do tabulky se používá příkaz SQLInsert. Výsledkem tohoto příkazu je číslo určující počet řádků. Při jednom vložení to bude 1, neboť najednou můžeme vložit pouze jeden řádek.⁸

⁷ Selecting Data: Wolfram Mathematica 8 Documentation. WOLFRAM RESEARCH INC. [online]. [cit. 2012-05-18]. Dostupné z:

<http://reference.wolfram.com/mathematica/DatabaseLink/tutorial/SelectingData.html>

```
In[10]:= SQLInsert[ conn, "SAMPLETABLE1", {"ENTRY", "VALUE", "NAME"}, {6, 8.2, "Day6"}]
Out[10]= 1
```

Zde je vidět nově vložený řádek.

```
In[11]:= SQLSelect[ conn, "SAMPLETABLE1", "ShowColumnHeadings" → True] // TableForm
```

```
Out[11]//TableForm=
```

ENTRY	VALUE	NAME
1	5.6	Day1
2	5.9	Day2
3	7.2	Day3
4	6.2	Day4
5	6.	Day5
6	8.2	Day6

Je také možné použitím klasického SQL příkazu vložit další data.

```
In[12]:= SQLExecute[ conn,
  "INSERT INTO SAMPLETABLE1(ENTRY, VALUE, NAME) VALUES (7, 6.9, 'Day7')"]
Out[12]= 1
```

2.1.7 Aktualizace dat

Pro aktualizování dat v tabulce se používá příkaz SQLUpdate. Často je použit v kombinaci s podmínkou, že je třeba měnit pouze některé údaje. V tomto příkladu je aktualizován sloupec VALUE s hodnotami, které jsou větší než 7 a 8.⁹

```
In[15]:= SQLUpdate[ conn, "SAMPLETABLE1", {"VALUE"}, {7}, SQLColumn["VALUE"] > 8]
Out[15]= 2
```

A zde jsou vidět uvedené změny.

```
In[16]:= SQLSelect[ conn, "SAMPLETABLE1", "ShowColumnHeadings" → True] // TableForm
```

```
Out[16]//TableForm=
```

ENTRY	VALUE	NAME
1	5.6	Day1
2	5.9	Day2
3	7.2	Day3
4	6.2	Day4
5	6.	Day5
7	6.9	Day7
6	7.	Day6
8	7.	Day8

⁸ Inserting Data: Wolfram Mathematica 8 Documentation. WOLFRAM RESEARCH INC. [online]. [cit. 2012-05-18]. Dostupné z:

<http://reference.wolfram.com/mathematica/DatabaseLink/tutorial/InsertingData.html>

⁹ Updating Data: Wolfram Mathematica 8 Documentation. WOLFRAM RESEARCH INC. [online]. [cit. 2012-05-18]. Dostupné z:

<http://reference.wolfram.com/mathematica/DatabaseLink/tutorial/UpdatingData.html>

Opět je zde možnost využít klasický SQL příkaz.

```
In[17]:= SQLExecute[ conn,
  "UPDATE SAMPLETABLE1 SET VALUE = `1` WHERE VALUE >= `2`", {7, 6}]
```

```
Out[17]= 6
```

```
In[18]:= SQLExecute[ conn, "SELECT * FROM SAMPLETABLE1"]
```

```
Out[18]= {{1, 5.6, Day1}, {2, 5.9, Day2}, {3, 7., Day3}, {4, 7., Day4},
  {5, 7., Day5}, {7, 7., Day7}, {6, 7., Day6}, {8, 7., Day8}}
```

2.1.8 Mazání dat

Pro mazání dat z databáze se používá příkaz `SQLDelete`, často je užit s podmínkou, že je třeba smazat pouze některá data. Následující příklad ukazuje, že budou smazána data, obsahující hodnoty 7 a vyšší.¹⁰

```
In[19]:= SQLDelete[ conn, "SAMPLETABLE1", SQLColumn["VALUE"] >= 7]
```

```
Out[19]= 6
```

Zde jsou vidět změny, které byly učiněny.

```
In[20]:= SQLSelect[ conn, "SAMPLETABLE1", "ShowColumnHeadings" → True] // TableForm
```

```
Out[20]//TableForm=
```

ENTRY	VALUE	NAME
1	5.6	Day1
2	5.9	Day2

Výsledkem tohoto příkazu je číslo, které určuje, kolik řádků bylo smazáno. V případě, že byly odstraněny tři řádky, pak výsledkem je číslo tři. Pokud nebyly smazány žádné řádky, tak výsledkem je nula.

2.1.9 Dávkové příkazy

V případě, že bude stejný příkaz použit několikrát, je rychlejší použít dávkový příkaz.¹¹

```
In[23]:= SQLInsert[ conn, "SAMPLETABLE1",
```

```
  {"ENTRY", "VALUE", "NAME"}, {{2, 5.9, "Day2"}, {3, 7.2, "Day3"}}]
```

```
Out[23]= {1, 1}
```

¹⁰ Deleting data: Wolfram Mathematica 8 Documentation. WOLFRAM RESEARCH INC. [online]. [cit. 2012-05-18]. Dostupné z:

<http://reference.wolfram.com/mathematica/DatabaseLink/tutorial/DeletingData.html>

¹¹ Getting started: Wolfram Mathematica 8 Documentation. WOLFRAM RESEARCH INC. [online]. [cit. 2012-05-18]. Dostupné z:

<http://reference.wolfram.com/mathematica/DatabaseLink/tutorial/GettingStarted.html>

2.1.10 Ukončení

V případě, že pro připojení do databáze využívá příkazu `OpenSQLConnection` je zapotřebí ho také ukončit. Volá se příkaz `CloseSQLConnection`. Jako parametr je opět název připojení.

```
In[26]:= CloseSQLConnection[conn]
```

2.1.11 Database Explorer

Grafické rozhraní k databázi. Spustí se při načtení `DatabaseLink` a zadání příkazu `Database Explorer`.

```
In[27]:= Needs["DatabaseLink`"];
```

```
DatabaseExplorer[]
```

```
Out[28]= - GUIObject -
```

Když se `Database Explorer` otevře, lze pracovat s různými databázemi, které jsou nakonfigurovány v systému.¹²

2.1.12 Zabezpečení a „autentizace“

Mnoho SQL databází může být nakonfigurováno tak, aby při uskutečnění spojení bylo vyžadováno uživatelské jméno a heslo. To je důležité jako prevence nežádoucího přístupu a omezení rozsahu činností, které mohou někteří uživatelé vykonávat. Při práci s hesly je třeba vzít v úvahu, že je zde několik oblastí, která vyžadují určitou úroveň zabezpečení.

Nejpohodlnější způsob jak pracovat s heslem je umístit jej do souboru konfigurace připojení. V případě, že by heslo bylo uloženo ve formátu prostého textu, mohlo by dojít k tomu, že by útočník v případě, že zkontroluje konfigurační soubor, heslo zjistí. Protože se jedná o bezpečnostní riziko „Průvodce novým připojením“ hesla automaticky neukládá, avšak lze upravit konfigurační soubor a toto změnit.

Můžeme vytvořit další ochranu v tom směru, že povolení pro práci s konfiguračním souborem bude mít pouze ten, kdo je oprávněn pracovat s `Wolfram Mathematica`.

¹² Getting started: Wolfram Mathematica 8 Documentation. WOLFRAM RESEARCH INC. [online]. [cit. 2012-05-18]. Dostupné z:

<http://reference.wolfram.com/mathematica/DatabaseLink/tutorial/GettingStarted.html>

Vyšší úroveň zabezpečení je dosaženo tím, že je použito GUI k zadání hesla, které má tu výhodu, že heslo pro přístup do databáze není nikde uloženo ve zdrojovém kódu.

```
In[3]:= conn = OpenSQLConnection["demo", "Username" -> "sa", "Password" -> "$Prompt"]
```

Heslo také může být zadáno do příkazu OpenSQLConnection. Používání zadávání hesla GUI není příliš praktické, jestliže Wolfram Mathematica běží uvnitř serveru. V tomto případě, je zde několik možností zabezpečení.

Je možné umístit heslo do konfiguračního souboru a omezit přístup těm, kteří používají Wolfram Mathematica přes webový server. Další možností je uložení hesla v ověřeném mechanismu poskytovaném webovým serverem.

Např. server Tomcat poskytuje mechanismus založený na JDBC realms. Heslo lze získat z webového serveru, následně spustit Wolfram Mathematica a použít jej v příkazu OpenSQLConnection. Bez porušení mechanismu autentizace webového serveru, nelze zjistit heslo databáze, přes jakoukoliv kontrolu kódu Wolfram Mathematica.¹³

2.1.13 Secure Socket Layer (SSL)¹⁴

Nejvyšší stupeň ochrany zabezpečení. Pro zabezpečení je použit certifikát k identifikaci jednoho či obou stran, tedy jak ze strany serveru, tak ze strany klienta. Některé databáze však SSL nepodporují. Jednou z databází, která nepodporuje SSL je MySQL. V tomto případě, musí dojít k vygenerování certifikátu, k tomu je třeba spolupráce se správcem databáze. Obvykle je třeba 4 kroků pro vytvoření SSL (pro MySQL):

1. Získat autorizovaný certifikát.
2. Generování truststore souboru.
3. Konfigurace Java pro použití souboru truststore.
4. Konfigurace připojení pro použití protokolu SSL.

¹³ Security and Authentification: Wolfram Mathematica 8 Documentation. WOLFRAM RESEARCH INC. [online]. [cit. 2012-05-18]. Dostupné z: <http://reference.wolfram.com/mathematica/DatabaseLink/tutorial/SecurityAndAuthentication.html>

¹⁴ Secure socket Layer (SSL): Wolfram Mathematica 8 Documentation. WOLFRAM RESEARCH INC. [online]. [cit. 2012-05-18]. Dostupné z: <http://reference.wolfram.com/mathematica/DatabaseLink/tutorial/SecureSocketLayer.html>

2.2 Kódování a dekódování pomocí base64¹⁵

Wolfram Mathematica umožňuje ukládání dat do databáze v kódování base64, které zajišťuje správný převod libovolných dat do ASCII kódu a jeho zpětné rozkódování včetně interpretace dat. Ve své diplomové práci budu tohoto kódování využívat pro ukládání složitých matematických vzorců a funkcí.

2.2.1 Příklad

Odstavec z knihy Leviathan (Thomas Hobbes):

Man is distinguished, not only by his reason, but by this singular passion from other animals, which is a lust of the mind, that by a perseverance of delight in the continued and indefatigable generation of knowledge, exceeds the short vehemence of any carnal pleasure.

Je v MIME Base64 zakódován takto:

```
TWFuIGlzlGRpc3Rpbmd1aXNoZWQsIG5vdCBvbmx5IGJ5IGhpcyByZWZzb24sIGJ1dCBieSB0aGlz
IHNpbmd1bGFyIHBhc3Npb24gZnVubSBvdGh1ciBhbmltYWxzLCB3aG1jaCBpcyBhIGx1c3Qgb2Yg
dGhlIG1pbmQsIHRoYXQgYnkgYSBwZXJzZXZlcmFuY2Ugb2YgZGVsaWdodCBpb1B0aGUgY29udGlu
dWVkaGFuZCBpbmR1ZmF0aWdhYm91IGd1bmV5YXRpb24gb2Yga25vd2x1ZGd1LCBleGN1ZWRzIHRo
ZSBzaG9ydCB2ZWwhbWVuY2Ugb2YgYW55IGNhcm5hbCBwbGVhc3VyZS4=
```

Obrázek 2: Zakódovaný text pomocí base64

2.2.2 Algoritmus base64

Z výše uvedeného příkladu je řetězec Man zakódován do řetězce TWFu.

M, a, n jsou zapsány v ASCII jako čísla 77, 97, 110 dekadicky, tedy ve dvojkové soustavě 01001101, 01100001, 01101110. Tyto 3 byte jsou spojeny do 24 bitů 010011010110000101101110. Následně jsou rozděleny do skupin po 6 bitech (6 bitů reprezentuje 64 možných stavů) a převedeny do 4 čísel (24 = 6×4). Tato čísla jsou použita jako index do tabulky znaků:

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/-

kde znaku "A" odpovídá index 0 a znaku "/" index 63.

¹⁵ Wikipedie: Base64. [online]. [cit. 2012-05-17]. Dostupné z: <http://cs.wikipedia.org/wiki/Base64>

Text content	M	a	n
ASCII	77	97	110
Bit pattern	0 1 0 0 1 1 0 1	0 1 1 0 0 0 0 1	0 1 1 0 1 1 1 0
Index	19	22	5
Base64-Encoded	T	W	F

Obrázek 3: Ukázka kódování prvních tří znaků Man

Jak je uvedeno v daném příkladu, kódování base64 převede každé 3 původní bajty (v našem případě 3 znaky ASCII) na 4 kódované znaky ASCII. Pokud počet nevyhází přesně na trojice, zakóduje se poslední jeden (dva) znak a přidají se dvě (jedno) rovnítka. Base64 se však používá v převážné míře k zakódování binárních dat, například multimédií.

2.2.3 Vlastnosti base64

Výsledný řetězec se skládá z tisknutelných znaků ASCII.

- Rozlišuje se velikost písmen.
- Nemá kontrolní mechanismus.
- Kódování probíhá binárně (neřeší se znakové kódování apod.).
- Base64 je binárně bezpečné.
- Délka výsledného řetězce se obvykle navýší o 33 %.

II. PRAKTICKÁ ČÁST

3 INTEGRACE S APLIKACÍ PRO EVOLUČNÍ ALGORITMY

3.1 Vývojové prostředí

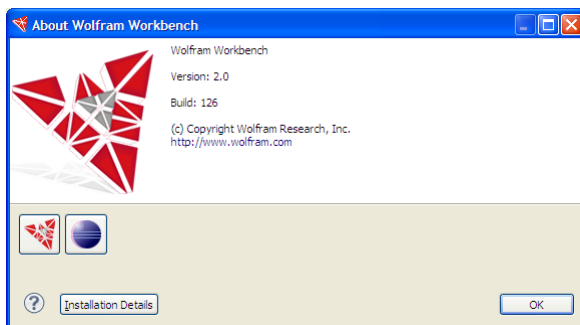
K vývoji JSP stránek je na výběr z několika vývojových prostředí, které jsou dostupné na Internetu. Většinu je možné získat zadarmo. Ovšem každé je určeno pro různě pokročilou skupinu vývojářů.

3.1.1 PSPad 4.5.6

Úplně prvotním záměrem bylo psát a upravovat JSP stránky pomocí vyzkoušeného a oblíbeného editoru PSPad¹⁶. Obsahuje spoustu doplňků a zvýrazňovačů. Umožňuje i vytvoření vlastního zvýrazňovače, případně rozšíření o psaní vlastních skriptů a maker. Ovšem nemá v sobě implementovaný zvýrazňovač pro vývoj JSP stránek. O validátoru ani není možno mluvit. Pro zkušeného programátora je to určitě velmi užitečný nástroj, ale pro nováčka v případě vývoje JSP stránek je nejrozzumnější podívat se po jiném nástroji.

3.1.2 Wolfram Workbench 2.0

Přímo od firmy Wolfram je k dispozici vývojové prostředí Wolfram Workbench 2.0¹⁷.

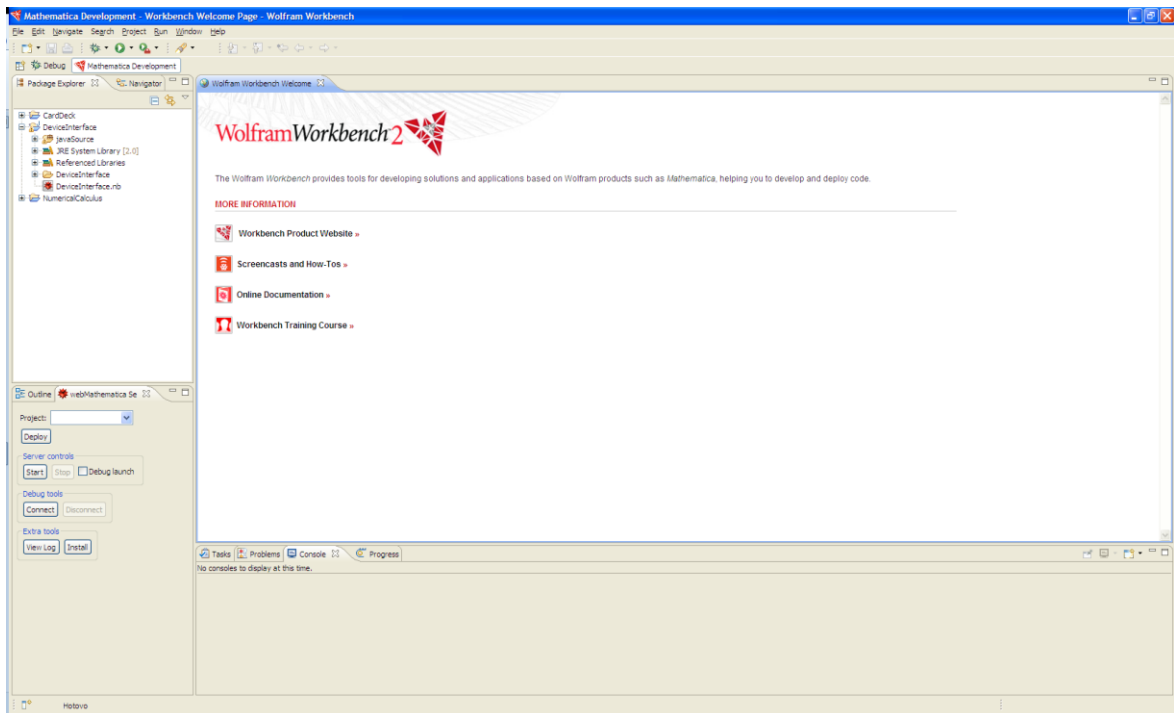


Obrázek 4: Wolfram Workbench 2.0 – o aplikaci

Vždy záleží na jaké vývojové prostředí je programátor zvyklý. Případně jak mu vyhovují další nástroje, které jsou umístěny uvnitř prostředí. Věřím, že tento nástroj je určitě přínosný při vývoji nejen JSP stránek. Je postaven na Eclipse Platform.

¹⁶ Dostupný z: <http://www.pspad.com>.

¹⁷ Více informací je dostupných z: <http://www.wolfram.com/products/workbench/>.



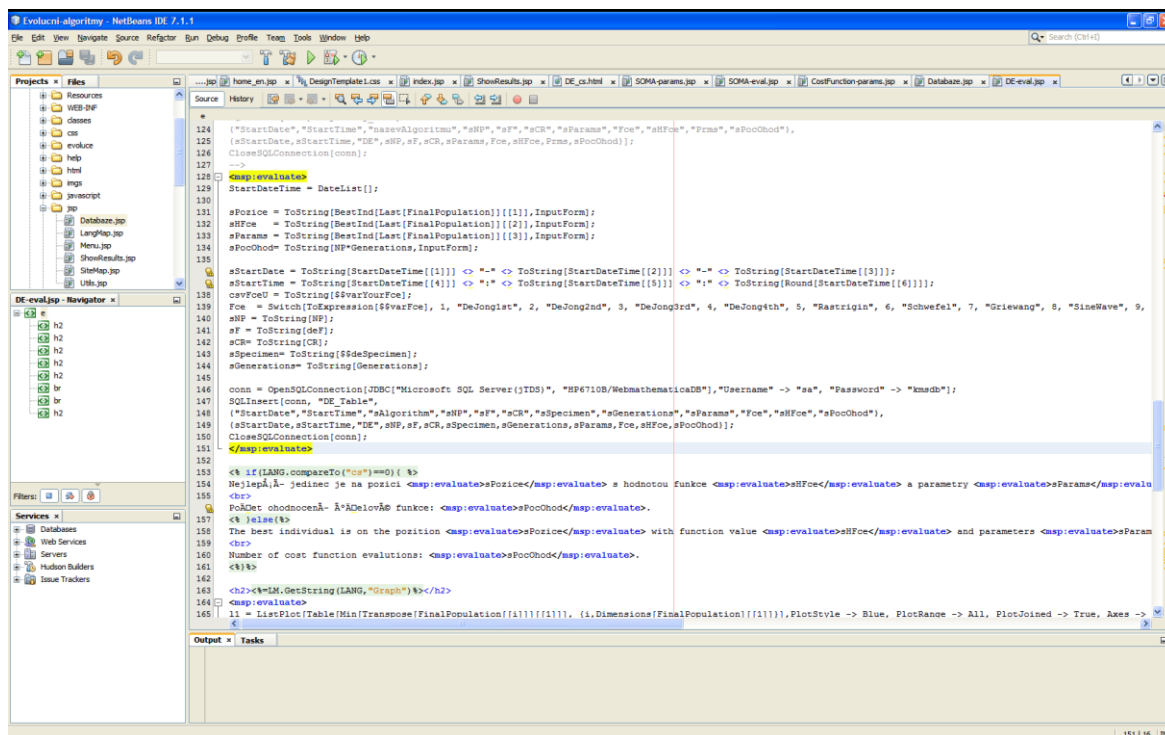
Obrázek 5: Vývojové prostředí JSP stránek – Wolfram Workbench

Priznám se však, že Wolfram Workbench 2.0 mi osobně nevyhovoval. Působil na mě velmi složitě. Nikdy předtím jsem nevytvářel JSP stránky, a toto prostředí nebylo moc sdílné. Pokoušel jsem se hledat návody na internetu, ovšem bez úspěchu. Proto jsem hledal další řešení.

3.1.3 Netbeans 7.1.1

Hned po nainstalování tohoto prostředí jsme seznámeni s možnostmi aktualizace jednotlivých modulů na nejaktuálnější. V předchozím vývojovém prostředí byly také aktualizace, ale ty se nepodařilo ani jednou úspěšně nainstalovat. To byl také jeden z důvodů, proč jsem hledal štěstí jinde.

U Netbeans mě překvapila jeho přehlednost a během pár kliknutí je celé prostředí nastaveno přesně tak, jak si uživatel přeje. Díky tomuto prostředí je možno vytvářet aplikace i přímo v jazyce Java. Proto jeho nainstalováním získáme další vývojářské výhody.



Obrázek 6: Vývojové prostředí Netbeans 7.1.1

Díky velkému množství návodů a video tutoriálů na internetu, bylo zprovoznění a nastavení tohoto prostředí velmi rychlé. Upoutalo mě i zvýrazňování syntaxe a případná validace napsaného kódu. Proto jsem k vývoji a práci na mé diplomové práci využíval tohoto prostředí.¹⁸

3.2 Příprava serveru

Aby bylo možno využívat evoluční algoritmy, musel být zprovozněn počítač, na kterém běží server TomCat, bude nainstalována Wolfram Mathematica, Wolfram WebMathematica a zároveň na něm poběží databáze.

3.2.1 JSP server (Apache TomCat)

Aplikace evolučních algoritmů je napsána pomocí jazyka JSP¹⁹ a pro svůj běh potřebuje mít nainstalovaný JSP server. Je vyžíván volně dostupný Apache TomCat 6.0 server. Při

¹⁸ Prostředí je dostupné z: <http://netbeans.org/>.

¹⁹ Java server pages

instalaci je doporučeno zvolit jiný než výchozí port 8080. Jelikož pokud na instalovaném počítači již běží jiný server, mohlo by docházet ke kolizi.

3.2.2 WebMathematica a Mathematica

Ve své diplomové práci využívám WebMathematicu ve verzi 3.1. Ta je dostupná zdarma jen k některým verzím Mathematicy. Tato verze je plně kompatibilní s Wolfram Mathematica 7.0.

3.3 Volba databáze

Při výběru databáze je třeba volit v závislosti, na jakém operačním systému poběží server. Respektive kde je nainstalovaná Mathematica s Webmathematicou a TomCatem. Jelikož mám rozběhnuté tyto aplikace pod MS Windows XP, tak jsem zvolil databázi MS SQL (MS SQL 2008 R2), se kterou má většina uživatelů dobré zkušenosti. V případě výběru Linuxu, MAC OS, či jiného operačního systému, navrhuji použít volně dostupnou MySQL databázi či PostgreSQL.

3.3.1 Instalace MySQL

K instalaci databáze MySQL, bývají většinou připojeny i další dvě aplikace Apache a phpMyAdmin. V našem řešení je ozkoušena sestava „Complex Web Server 1.6.0“²⁰.

Apache je server, na kterém se provádí PHP skripty. Druhá aplikace využívá PHP skripty k ovládní MySQL databáze. Výchozí nastavení Apache serveru je nastaveno na port: 80. Na základě zkušeností získaných při vypracování této diplomové práce je doporučeno nastavit port na jiný než výchozí. Nastavíme ho tedy při instalaci na port 8090. Port MySQL databáze, můžeme nechat na výchozí hodnotě a to 3306.

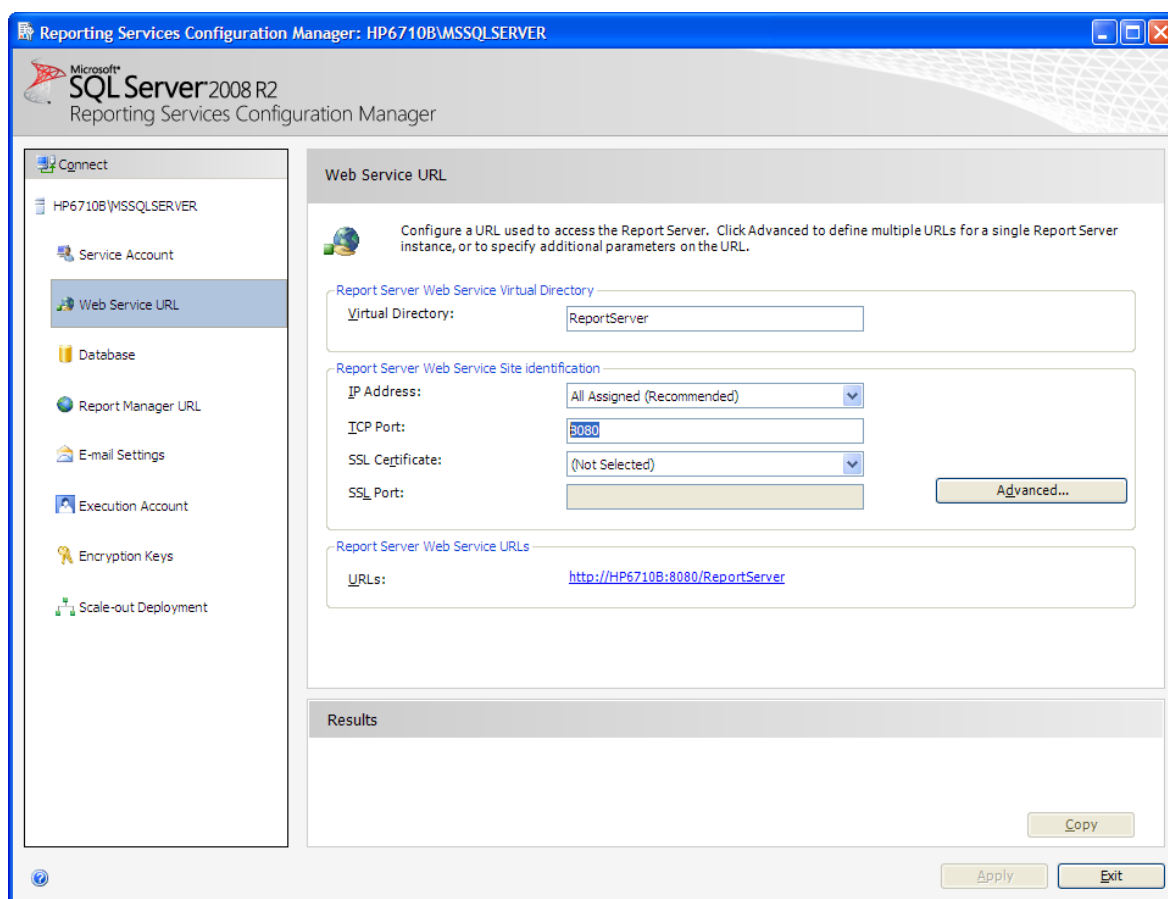
3.3.2 Instalace MS SQL Server 2008 R2

Na server byla nainstalována databáze MS SQL Server 2008 R2 Developer Edition (Placená verze SQL serveru. Je možné nainstalovat i MS SQL Server 2008 R2 Express²¹,

²⁰ Více informací je dostupných z: <http://ponkrac.net/complex-web-server/cs>.

²¹ Dostupný z: <http://www.microsoft.com/en-us/download/details.aspx?id=23650>.

kteřá je dostupná zdarma.). Byla nainstalována včetně všech nástrojů pro správu a vývoj (Integration Services, Analysis Services, Report Services, Replication,...). Důvodem instalace všech těchto nástrojů bylo také ověření a analýza možných kolizí. U Report Serveru je jako výchozí hodnota nastaven port 8080. Ovšem ten je již využíván Apache TomCat Serverem. Z tohoto důvodu je nutné, po nainstalování MS SQL databáze, přejít do C:\Documents and Settings\All Users\Nabídka Start\Programy\Microsoft SQL Server 2008 R2\Configuration Tools\Reporting Services Configuration Manager a změnit port na jiný. Toto změna se provede v záložce „Web Service URL“ viz obrázek na další straně. Nesmíme zapomenout změnit port i na záložce „Report Manager URL“. Doporučuji nastavit na port 8060.



Obrázek 7: Nastavení portu ReportServeru u MS SQL 2008 R2

Aby bylo možné volat z externí aplikace (Mathematica, Webmathematica) MS SQL server, tak bylo ještě zapotřebí spustit TCP/IP podporu serveru, která je standardně vypnutá. To se provede pomocí SQL Server Configuration Manager na záložce SQL Server Network Configuration / Protocols for MSSQLSERVER/ TCP/IP z Disabled na Enabled.

3.3.3 Instalace PostgreSQL

Instalace a nastavení databáze PostgreSQL²² proběhne bez problémů. V našem případě máme konkrétně nainstalovanou databázi PostgreSQL ve verzi 9.1. Databáze běží standardně na portu 5432, který není s žádnou další aplikací v kolizi, a proto ji není třeba měnit.

K administraci databáze je vyvíjen GUI nástroj PG Admin²³ v1.14.2. Díky němu je vytváření databází a tabulek, daleko příjemnější než využívání příkazového řádku.

3.4 Wolfram Mathematica

Důležitým krokem při integraci s databázovým systémem je velmi podstatná analýza. Díky ní bylo zjištěno a upuštěno z předchozího návrhu odchylovat a ukládat data, která vygeneruje Mathematica. Ne jen, že by to nešlo, ale bylo by třeba řešit ještě výjimky, které by byly způsobeny například dlouhým výpočtem, či nereagováním kernel jádra Mathematicy.

Integraci s databází jsem vyřešil přímo pomocí aplikace Mathematica, která bude po vypočtení daného požadavku přímo ukládat data do databáze. Z databáze se budou následně data načítat pomocí Webmathematicy.

3.5 Výběr vhodné metody připojení

Po prostudování velkého množství materiálů jsem se rozhodl, že využiji k ukládání dat do databáze balík DatabaseLink, který je již detailně popsán v teoretické části.

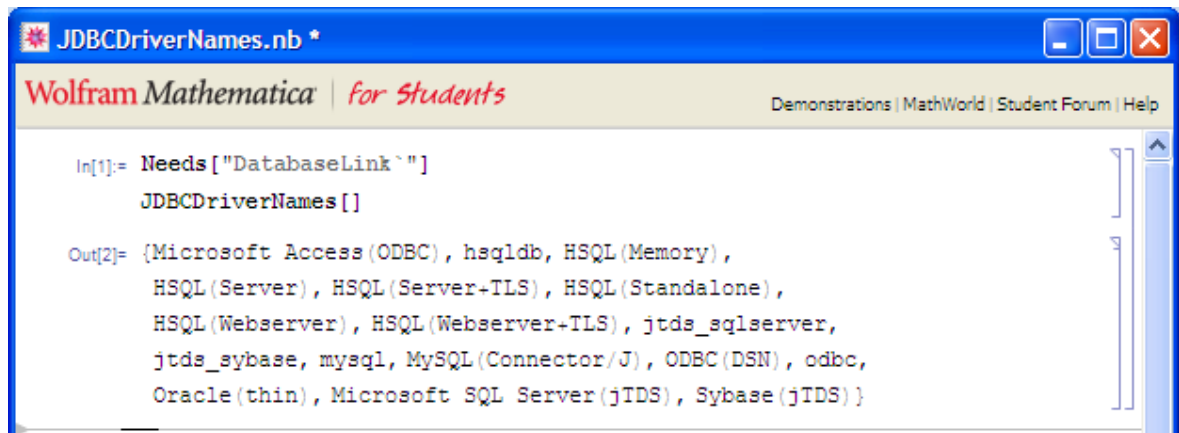
3.6 Přehled ovladačů databáze

Díky velkému množství konektorů do databází, které nabízí DatabaseLink, je možno si vybrat do jaké databáze se budeme napojovat. Prvně je zapotřebí zjistit, jaké ovladače jsou již nainstalované. Do nového notebooku v Mathematice zadáme příkaz

²² Dostupný z: <http://www.postgresql.org/>.

²³ Dostupný z: <http://www.pgadmin.org/>.

„JDBCdriverNames[]“, před kterým musí být ještě řádek s příkazem „Needs[\"DatabaseLink`\"]“. Viz následující obrázek.



Obrázek 8: Výpis nainstalovaných JDBC ovladačů ve Wolfram Mathematice

Na výstupu je vidět, že JDBCdriver má nainstalovanou podporu pro MS Access, HSQL, ODBC, mysql, MS SQL Server a další ovladače. Kdyby nám nějaký ovladač chyběl je možno ho stáhnout z internetu a doinstalovat do Mathematicy.

3.7 Přidání nového ovladače

Pokud v seznamu není ovladač databáze, kterou chceme využít, je zapotřebí doinstalovat ovladač databáze. Například Mathematica ve verzi 7.0 nemá standardně nainstalovaný ovladač pro PostgreSQL. Na následující obrázku je vidět, jaký je postup instalace.

3.7.1 Zjištění výchozího adresáře

Nejdříve je zapotřebí zjistit, kde je nastaven výchozí adresář. Zadáme do Mathematicy následující příkaz.

```
In[1]:= $UserBaseDirectory
Out[1]= C:\Documents and Settings\TaRTiF\Data aplikací\Mathematica
```

3.7.2 Vytvoření adresáře pro soubor jar

V dalším kroku vytvoříme adresáře pro soubor s ovladačem.

```
In[2]:=
$jarDirectory = CreateDirectory@FileNameJoin@{$UserBaseDirectory, "Applications", "Postgre", "Java"}

Out[2]=
C:\Documents and Settings\TaRTiF\Data aplikací\Mathematica\Applications\Postgre\Java
```

Konkrétně tři adresáře do sebe zanořené Applications, Postgre, Java. Jména adresářů je možno měnit, ale je doporučeno ponechat toto nastavení. Zkontrolujeme, zda se nám požadované adresáře vytvořily. Pokud ne, tak je ještě zapotřebí zkontrolovat uživatelská práva do složek a případně jim nastavit přístup k zápisu. Do adresáře Java nakopírujeme soubor postgresql-9.1-902.jdbc4.jar, který jsme si stáhli z domovských stránek²⁴. Vždy je třeba si dávat pozor na to, jakou máme nainstalovanou verzi Javy. Protože by nám ovladač nemusel fungovat.

3.7.3 Vytvoření adresáře pro nastavení

K vytvoření adresářů, využijeme následující příkaz.

```
In[3]:=
$configDirectory =
  CreateDirectory@FileNameJoin@{$UserBaseDirectory, "Applications", "Postgre", "DatabaseResources"}

Out[3]=
"C:\Documents and Settings\TaRTiF\Data aplikací\Mathematica\Applications\Postgre\DatabaseResources"
```

Stejně jako v předchozím případě, se vytvoří tři adresáře respektive už jen jeden a to DatabaseResources.

3.7.4 Export nastavení do souboru

Následujícím příkazem vytvoříme soubor postgresql.m, ve kterém bude uloženo nastavení.

```
In[4]:=
Export[FileNameJoin@{$configDirectory, "postgresql.m"},
  JDBCDriver["Name" → "PostgreSQL", "Driver" → "org.postgresql.Driver", "Protocol" → "jdbc:postgresql://",
    "Version" → 2., "Description" → "PostgreSQL using PostgreSQL JDBC Driver - Version 9.1.902. "],
  "Text"]

Out[4]=
C:\Documents and Settings\TaRTiF\Data
  aplikací\Mathematica\Applications\Postgre\DatabaseResources\postgresql.m
```

3.7.5 Kontrola nainstalování nového ovladače

Nyní je zapotřebí zkontrolovat jestli byl námi požadovaný ovladač úspěšně nainstalován. Zkontrolujeme to příkazem „JDBCdriverNames[]“ následovně.

```
In[5]:= JDBCdriverNames[]
```

²⁴ Domovské stránky jsou dostupné z: <http://jdbc.postgresql.org/download.html>.

Jako odpověď od Mathematicy vidíme hned na prvním místě nově nainstalovaný ovladač.

```
Out[5]=
{PostgreSQL, Microsoft Access(ODBC), hsqldb, HSQL(Memory), HSQL(Server), HSQL(Server-TLS),
  HSQL(Standalone), HSQL(Webserver), HSQL(Webserver-TLS), jtds_sqlserver, jtds_sybase, mysql,
  MySQL(Connector/J), ODBC(DSN), odbc, Oracle(thin), Microsoft SQL Server(jTDS), Sybase(jTDS)}
```

Nyní je potřeba ještě ověřit jeho funkčnost. Pokusíme se tedy do databáze připojit. Do příkazu zadáme:

```
In[6]:=
$connection = OpenSQLConnection[JDBC["PostgreSQL", "localhost:5432/Webmathematica"],
  "Username" -> "postgres", "Password" -> "aktofka"]
Out[6]=
SQLConnection[16, Open, Catalog -> Webmathematica, TransactionIsolationLevel -> ReadCommitted]
```

Je vidět, že se podařilo otevřít databázi Webmathematica s zadáním uživatelského jména a hesla.

Nyní provedeme výběrový SQL dotaz z tabulky a budeme výsledky řadit dle příjmení. V databázi je tabulka phonebook, kterou jsme si předpřipravili a naplnili daty. Ptáme se do ní takto:

```
In[7]:=
SQLExecute[$connection, "SELECT * FROM phonebook ORDER BY lastname;"]
Out[6]=
{{+420 777 123456, Michal, Svatek, Ceska Republika}}
```

V databázi je umístěn pouze jeden záznam, tak nebyla klauzule ORDER BY ani nutná.

3.8 Napojení na Diferenciální evoluci

Nejzásadnějším a nejdůležitějším počinem je samotné rozjetí aplikace pro evoluční algoritmy. To je jeden z nejdůležitějších okamžiků, který zabere poměrně dost času. Následně proběhla analýza celého kódu.

V aplikaci pro evoluční algoritmy jsou pro nás důležité místa, kde jsou v tagovací značce `<msp:evaluate></msp:evaluate>` umístěny příkazy posílané Mathematica. Konkrétně u Diferenciální evoluce je nejpodstatnější část kódu umístěna v souboru DE-eval.jsp.

Tento soubor je třeba modifikovat, aby se výsledek diferenciální evoluce uložil do databáze. V úvodu JSP souboru je třeba uvést, že se bude využívat balík „<< "DatabaseLink`";“. Následně po výpočtu se přenesou výsledky pomocí příkazu SQLInsert do předem vytvořené databáze.

The screenshot shows the Wolfram webMathematica interface for a differential evolution algorithm. The page title is "Diferenciální evoluce". The user has selected the "De Jong Test" function. The parameters are set as follows: NP: 10, F: 1, CR: 0.8, Specimen: $\{((Re: [-5, 12.5, 11])), ((Re: [-5, 12.5, 11]))\}$, and Generations: 5. The results are displayed in two tables:

Počáteční populace

11.6882	24.4653	0.67041	10.0694	7.78383	9.90371	11.1139	16.7368	1.08646	18.3267
(-3.28318, 1.01873)	(4.34126, -2.3704)	(2.38849, -1.03377)	(-2.10865, -2.37307)	(1.52873, 2.33515)	(2.97127, -1.06549)	(-1.68005, -2.89104)	(-4.04661, -0.60124)	(0.0539665, 1.04093)	(2.09911, -3.73101)

Finální populace

7.03076	-4.67845	1.14116	4.58894	4.2878	8.94482	5.21503	1.57842	1.08848	7.42811
(0.557644, 2.58009)	(-2.15583, 0.175673)	(0.944013, -0.500004)	(1.00959, -1.68856)	(2.07038, -0.0363097)	(-2.98479, 0.18881)	(1.5862, -1.64287)	(1.21977, 0.297618)	(0.0539665, 1.04093)	(0.496875, -2.67945)

Vyhodnocení
Nejlepší jedinec je na pozici 9 a hodnotou funkce 1.0864611551507717 a parametry (0.05396646888493788, 1.040934830096961).
Počet ohodnocení účelové funkce: 50.

Graf

Obrázek 9: Náhled na vyřešený příklad pomocí diferenciální evoluce

V závislosti na databázi se mění i přípojovací řetězec. V případě že budeme chtít ukládat data do dvou databází, tak i to je možné. Ovšem je prvně zapotřebí ukončit první připojení a následně navázat nové připojení do jiné databáze.

3.8.1 Rozbor uložení dat do databáze

Na předchozím obrázku vidíme, že vstupními daty jsou parametry funkce. Ty budeme ukládat do databáze. Konkrétně se jedná o hodnoty „NP“, „F“, „CR“, „Specimen“, „Generations“. Budeme ukládat samozřejmě i výstupní data, která získáme výpočtem. A to „hodnotu funkce“, „parametry“, „počet ohodnocení funkce“.


Je tedy zapotřebí vytvořit tabulku v databázi, která bude mít minimálně výše uvedené sloupce. My ji zároveň doplníme o sloupec „ID“, který se bude s vložením každého záznamu automaticky inkrementovat s krokem jedna.

Zároveň bude tabulka obsahovat i datum a čas, kdy se požadovaný algoritmus spouštěl a jaká funkce na něj byla použita.

3.8.2 ERD diagram pro DE

K návrhu entitně relačního diagramu je doporučeno využít nástroj Toad Data Modeler²⁵.

²⁵ Dostupný z: <http://www.quest.com/toad-data-modeler/>

DE_Table		
 ID	Int	NN (PK)
StartDate	Varchar(50)	
StartTime	Varchar(50)	
sAlgorithm	Varchar(50)	
sNP	Varchar(50)	
sF	Varchar(50)	
sCR	Varchar(50)	
sSpecimen	Varchar(50)	
sGenerations	Varchar(50)	
Fce	Varchar(50)	
sHFce	Varchar(50)	
sParams	Varchar(50)	
sPocOhod	Varchar(50)	

Obrázek 10: Ukázka datového modelu tabulky Diferenciální evoluce

Jak je vidět na ERD diagramu, tak všechny datové typy jsou datového typu Varchar 50. To znamená, že délka ukládaného řetězce je pouze 50 znaků. Jedná se pouze o demonstraci uložení dat. Určitě by bylo možné využít jiné, vhodnější datové typy. Ovšem pokud by se načtené hodnoty měly opět využívat pro načítání a opětovné využívání, je nejvhodnější datový typ právě Varchar. Jednoznačná identifikace jednotlivých záznamů je zajištěna při vložení do databáze přiřazením jednoznačného identifikátoru ID, který je zároveň primárním klíčem tabulky.

3.8.3 Ukázka dat uložených v databázi

V případě, že proběhne evoluce v pořádku, tak její data se uloží do databáze. Viz následující detail v databázi.

	StartDate	StartTime	sAlgorithm	sNP	sF	sCR	sSpecimen	sGenerations	Fce	sHFce	sParams	sPocOhod
1	2012-5-20	6:8:3	DE	10	1	0.8	{{{Re. {-10, 10}}}, {{Re. {-10, 10}}}}	5	x[[1]]~x[[2]]	-16.61621632725982	{-9.643940749806577, -6.972375577453243}	50
2	2012-5-20	6:9:52	DE	10	1	0.8	{{{Re. {-5, 12.5, 11}}}, {{Re. {-5, 12.5, 11}}}}	5	DeJong1st	0.478729067094534	{-0.6844577736776198, 0.1012256051935978}	50
3	2012-5-20	6:10:36	DE	10	1	0.8	{{{Re. {-2.048, 2.047}}}, {{Re. {-2.048, 2.047}}}}	5	DeJong2nd	6.101919600694172	{-0.20101232382501033, -0.17545254099649265}	50
4	2012-5-20	7:37:36	DE	10	1	0.8	{{{Re. {-2.048, 2.047}}}, {{Re. {-2.048, 2.047}}}}	5	DeJong3rd	0.3122404196027624	{-0.28914367207909253, 0.02309674752366986}	50
5	2012-5-20	7:38:45	DE	10	1	0.8	{{{Re. {-1.28, 1.27}}}, {{Re. {-1.28, 1.27}}}}	5	DeJong4th	0.008141246287944864	{0.3003757870932906, -0.023255427815893848}	50
6	2012-5-20	7:51:12	DE	10	1	0.8	{{{Re. {-5, 12.5, 11}}}, {{Re. {-5, 12.5, 11}}}}	5	DeJong1st	1.0864611551507717	{0.05398946888493798, 1.0409348300969961}	50

Obrázek 11: Detail tabulky DE_Table z Microsoft SQL Server Management Studio

4 NÁVOD NA PROPOJENÍ DALŠÍCH SYSTÉMŮ Z DALŠÍCH PŘEDMĚTŮ

V této části diplomové práce je zpracován návod, jak postupovat při vytváření, či modifikaci aplikace a na co si dávat pozor a čeho se držet.

4.1 Analýza

Prvním bodem je správně zanalyzovat požadované řešení. V případě, že budeme přepracovávat již hotové řešení, je třeba nalézt místo, kde je vykonáván celý výpočetní mechanismus (Mathematica kód), a kde jsou již všechny výsledky finálně zpracovány.

Následně požadované proměnné převést funkcí „ToString[NázevProměnné]“ na řetězce na novou textovou proměnnou „sNázevProměnné“. Tyto proměnné se v dalším kroku vloží do tabulky (jako jeden záznam). Tabulka bude obsahovat názvy sloupce shodné s názvy atributů, které byly použity v implementaci. Akorát tyto sloupce budou pro lepší odlišení uvozeny též prefixem „s“.

Konkrétně bude-li na výstupu proměnná „Funkce“, bude sloupec v tabulce nazván „sFunkce“ a textová proměnná taktéž.

4.2 Doporučená vývojová prostředí

Na základě zkušeností je doporučeno využít pro úpravu JSP stránek NetBeans IDE 7.1.1 či vyšší.

4.3 Vytvoření ERD diagramu

Na základě předchozí analýzy lze dále v programu Toad Data Modeler navrhnout ERD diagram. Na něm jsou zřetelně vidět všechny atributy konkrétní tabulky včetně datového typu.

4.4 Volba datových typů

Vždy záleží na typu databáze, do které chceme data ukládat. V případě, že budeme využívat MS SQL server, volíme jako datový typ Varchar o určité délce. V případě, že

nevíme, jak bude velká velikost vstupních dat, můžeme definovat datový typ Varchar (max). U MySQL je nejdelším ekvivalentem Varchar (65535).

4.5 Kódování base64 v Mathematice

V případě že očekáváme, že vstupní data budou obsahovat složité matematické vzorce je přímo v Mathematica pro to implementována funkce. Je zapotřebí využít balíčku „<<base64“.

```
In[1]:= ExportString[(a + b) / 2, "Base64"]
Out[1]= KGEgKyBiKS8y
```

Na ukázce výše je vidět jakým způsobem lze převést matematický výraz do kódování base64. Převod zpět do čitelné podoby lze provést následovně.

```
In[2]:= ImportString[%, "Base64"]
Out[2]= (a + b) / 2
```

Původním záměrem bylo využití kódování base64 na ukládání obrázků do databáze. Bylo ale zjištěno, že funkce implementovaná v Mathematica pravděpodobně obsahuje chybu v případě, kdy jsou vkládána data příliš velká, a proto na ukládání obrázků nakonec nebylo možné toto kódování využít. Avšak funkční řešení je dostupné v následujících bodech.

4.6 Výběr datových typů dle typu databáze

Na následující tabulce jsou vyobrazeny datové typy, dle typu databáze, které je vhodné využít na ukládání dat do databáze.

Tabulka 1: Přehled datových typů dle druhu databáze

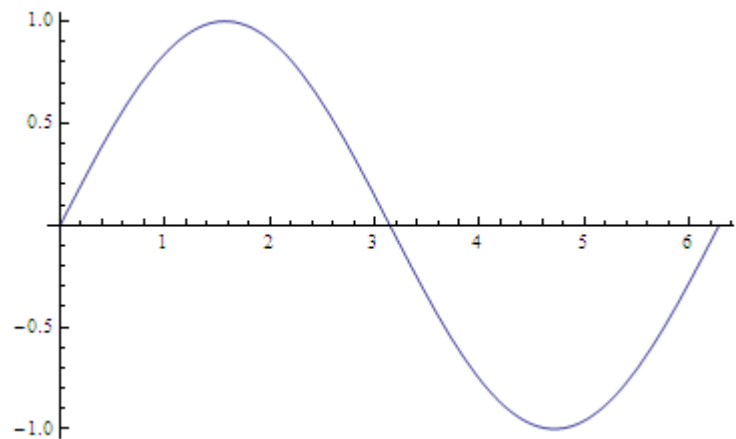
Databáze	Typ řetězec	Typ binární data
MySQL	Varchar(65535)	BLOB
MS SQL	Varchar(max)	Varbinary(max)

4.7 Příklad ukládání obrázku do databáze

Předtím než začneme volat databázi je zapotřebí si v ní vytvořit tabulku „BINTABLE“, která bude mít pouze jeden sloupec „BINCOL“.


```
In[9]:= Show[ImportString[gifData, "GIF"]]
```

```
Out[9]=
```



Zde je vidět výsledek, který byl uložený v databázi z tabulky „BINTABLE“. Pro úplnost jsou uvedeny ještě dva příkazy, které vymažou tabulku a ukončí spojení s databází.

```
In[10]:= SQLDropTable[conn, "BINTABLE"];  
         CloseSQLConnection[conn];
```

Je možné vyzkoušet tento příklad i na MS SQL, kde však je nutno vytvořit sloupec „BINCOL“ datového typu Varchar(max). V příloze PI je uvedeno jak získat data z databáze a uložit si je přímo na disk.

ZÁVĚR

Díky vypracování této diplomové práce jsem se seznámil s velkým množstvím různých vývojových prostředí a musel jsem řešit situace, které si vyžádali mnohdy i několik dní zkoušení a pokusů, než se podařilo dojít k úspěšnému vyřešení. Zdokonalil jsem si přehled o databázích a zároveň se dozvěděl o jejich možnostech a limitech.

Také jsem se naučil ovládat program Mathematica a jeho balíčky. Shledávám tento program velmi užitečným nástrojem ne jen díky příjemnému uživatelskému prostředí, ale i možnostem rozšíření.

V teoretické části jsem se pokusil srozumitelně přepsat z dostupných pramenů teorii a seznámit čtenáře s problematikou práce s databází v prostředí Wolfram Mathematica.

Praktická část obsahuje praktické ukázky a řešení jednotlivých úkolů.

Podařilo se mi proniknout i do tajů psaní JSP stránek, o jejichž existenci jsem se prvně dozvěděl až při vypracovávání této práce.

V mé diplomové práci jsem se pokusil popsat situace, se kterými jsem se během vývoje setkal. Snažil jsem se sepsat postup, jak dané situace vyřešit. Je více než pravděpodobné, že může dojít k dalším situacím, na které jsem nenarazil, a které mohou nastat při další implementaci.

Dle výsledků, kterých jsem dosáhl, doufám, že jsem splnil všechny body práce, které byly zadány. Jsem si vědom, že je určitě možné dále pokračovat na práci a více ji rozšiřovat a optimalizovat.

ZÁVĚR V ANGLIČTINĚ

Through this diploma thesis I recognized a lot of development environment and I had to look for a solution, which needs few days testing and experiments then I found successful solution. I improved knowledge about databases and their alternatives and limits.

I learned to operate with program Mathematica and its packages. I think that this program is very useful for its user friendly interface and possibility of extension.

I tried clearly present reader with work of database in Wolfram Mathematica in the theoretical part.

Practical part contains examples and solution of tasks. Through this diploma thesis I learned how to write JSP pages.

In my diploma thesis I tried to describe the situation, which I discovered through my development. I wanted to write process how to solve the situation. It is possible that it can be the other situation, which I didn't find, but it can be in the other implement.

I hope that I had do everything what I want in my diploma thesis. I know that it was possible to continue in this work and make it extend and more optimize.

SEZNAM POUŽITÉ LITERATURY

- [1] ZELINKA, I., OPLATKOVÁ, Z., OŠMERA, P., ŠEDA, M., VČELAŘ, F.,: *Evoluční výpočetní techniky – principy a aplikace*, BEN – technická literatura, Praha, 2008, ISBN 80-7300-218-3
- [2] MAŇAS, Jaroslav. *Prostředí pro porovnávání evolučních algoritmů v prostředí webMathematica* [online]. Zlín, 2010 [cit. 2012-04-10]. Dostupné z: <http://theses.cz/id/2bgp8k/>. Diplomová práce. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky. Vedoucí práce Ing. Zuzana Oplatková, Ph.D.
- [3] The Mathematica solutions [online]. [cit. 2012-05-18]. Dostupné z: <http://www.wolfram.com/solutions/>
- [4] Mathematica Features. [online]. [cit. 2012-05-18]. Dostupné z: <http://www.wolfram.com/mathematica/features/>
- [5] Introduction to DatabaseLink. [online]. [cit. 2012-05-17]. Dostupné z: <http://reference.wolfram.com/mathematica/DatabaseLink/tutorial/Introduction.htm>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ASCII	Z anglického American Standard od efor Information Interchange – Americký standardní kód pro výměnu informací
ERD	Z anglického Entity-Relationship Diagram – Datový model
FAI	Fakulta aplikované informatiky
GUI	Z anglického Graphical User Interface, Grafické uživatelské rozhraní. Jedná se o rozhraní, ve kterém uživatel za pomoci interaktivních grafických prvků ovládá počítač
HTML	Z anglického HyperText Markup Language – značkovací jazyk pro tvorbu statických webových stránek.
HSQLDB	Z anglického Hypersonic SQL Database – relační databázový systém napsaný v Javě
IS	Z anglického Information System – Informační systém
JAVA	Programovací jazyk a výpočetní platforma vyvinutá společností Sun Microsystems
JDBC	Z anglického Java DataBase Connectivity – Java databázové propojení
JSP	Java Server Pages, technologie pro tvorbu dynamických webových stránek v jazyce Java
MS SQL	Microsoft SQL
ODBC	Z anglického Open Database Connectivity – standardizovaný software API – standardizované aplikační rozhraní pro přístup k databázovým systémům
RDBMS	Z anglického Relational Database Management System – relační databázový systém
SOMA	Samoorganizující se Migrační Algoritmus
SQL	Z anglického Structured Query Language – strukturovaný dotazovací jazyk
SSL	Z anglického Secure Socket Layer – software, který poskytuje zabezpečení komunikace šifrováním a autentizací komunikujících stran

STAG	Studijní agenda
TCP/IP	Z anglického Transmission Control Protocol/Internet Protocol – primární transportní protokol – TCP/protokol síťové vrstvy – IP
URL	Z anglického Uniform Resource Locators – Jednotný popis umístění zdroje. Nejužívanější schéma specifikace dokumentu (jeho umístění a typ) v Internetu.
UTB	Univerzita Tomáše Bati

SEZNAM OBRÁZKŮ

Obrázek 1: Připojení do databáze pomocí GUI – Connection Tool	17
Obrázek 2: Zakódovaný text pomocí base64	23
Obrázek 3: Ukázka kódování prvních tří znaků Man	24
Obrázek 4: Wolfram Workbench 2.0 – o aplikaci	26
Obrázek 5: Vývojové prostředí JSP stránek – Wolfram Workbench	27
Obrázek 6: Vývojové prostředí Netbeans 7.1.1	28
Obrázek 7: Nastavení portu ReportServeru u MS SQL 2008 R2	30
Obrázek 8: Výpis nainstalovaných JDBC ovladačů ve Wolfram Mathematice	32
Obrázek 9: Náhled na vyřešený příklad pomocí diferenciální evoluce	35
Obrázek 10: Ukázka datového modelu tabulky Diferenciální evoluce.....	36
Obrázek 11: Detail tabulky DE_Table z Microsoft SQL Server Management Studio	36

SEZNAM TABULEK

Tabulka 1: Přehled datových typů dle druhu databáze	38
---	----

SEZNAM PŘÍLOH

Příloha PI: Uložení varbinary(max) na disk

Příloha PII: CD se zdrojovými kódy webové aplikace

PŘÍLOHA P I: ULOŽENÍ VARBINARY(MAX) NA DISK

V MS SQL Management Studiu otevřeme nový SQL Query a do něj zkopírujeme následující kód. Vygeneruje se nám na disk e, takový počet gif souborů, jako je počet záznamů v tabulce. Jména souborů budou generována podle času, kdy byly vytvořeny.²⁶

```
sp_configure 'show advanced options', 1;
GO
RECONFIGURE;
GO
sp_configure 'Ole Automation Procedures', 1;
GO
RECONFIGURE;
GO

DECLARE @SQLIMG VARCHAR(MAX),
        @IMG_PATH VARBINARY(MAX),
        @TIMESTAMP VARCHAR(MAX),
        @ObjectToken INT

DECLARE IMGPATH CURSOR FAST_FORWARD FOR
        SELECT BINCOL from BINTABLE

OPEN IMGPATH

FETCH NEXT FROM IMGPATH INTO @IMG_PATH

WHILE @@FETCH_STATUS = 0
    BEGIN
        SET @TIMESTAMP = 'e:\' +
replace(replace(replace(replace(convert(varchar,getdate(),121),'-',''),':',''),'.',''),' ','') + '.gif'

        PRINT @TIMESTAMP
        PRINT @SQLIMG

        EXEC sp_OACreate 'ADODB.Stream', @ObjectToken OUTPUT
        EXEC sp_OASetProperty @ObjectToken, 'Type', 1
        EXEC sp_OAMethod @ObjectToken, 'Open'
        EXEC sp_OAMethod @ObjectToken, 'Write', NULL, @IMG_PATH
        EXEC sp_OAMethod @ObjectToken, 'SaveToFile', NULL, @TIMESTAMP, 2
        EXEC sp_OAMethod @ObjectToken, 'Close'
        EXEC sp_OADestroy @ObjectToken

        FETCH NEXT FROM IMGPATH INTO @IMG_PATH
    END

CLOSE IMGPATH
DEALLOCATE IMGPATH
```

²⁶ Čerpáno z: <http://stackoverflow.com/questions/4056050/script-to-save-varbinary-data-to-disk>