

Technická univerzita v Liberci
Fakulta mechatroniky a mezioborových inženýrských studií
Ústav nových technologií a aplikované informatiky

Diplomová práce

Multimediální encyklopedie

Multimedia encyclopaedia

Bc. Jan Klimeš

Vedoucí práce: Ing. Igor Kopetschke

Konzultant: Ing. Tomáš Liška, FoxCom, s.r.o.

Studijní program: N2612 – Elektrotechnika a informatika

Studijní obor: 1802T007 – Informační technologie

květen 2009

Zde patří originální zadání práce

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé DP a prohlašuji, že **souhlasím** s případným užitím mé diplomové práce.

Jsem si vědom toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

V Liberci 29.5.2009

.....

Poděkování

Děkuji Ing. Igoru Kopetschkovi za vedení práce, dále Ing. Tomáši Liškovi z firmy FoxCom, s.r.o. za odborné konzultace. Donaldu Ervinu Knuthovi za naprogramování typografického systému \TeX a Leslie Lamportovi za rozšíření na \LaTeX , díky tomu se psaní diplomové práce usnadnilo.

Dále bych chtěl poděkovat Mgr. Olze Votavové za jazykovou korekturu, svým rodičům a prarodičům za podporu po celou dobu mého studia.

Abstrakt

Cílem diplomové práce je návrh a implementace multimediální encyklopedie za využití moderních technologií v prostředí Adobe Flash CS3 a objektově orientovaného programovacího jazyka ActionScript 3.0.

První část se zabývá definicí pojmů multimédia a encyklopedie. Následuje popis, historie a vlastnosti použitých technologií a jejich porovnání s alternativními nástroji.

Hlavní důraz je kladen na vlastní implementaci multimediálního programu, založené na sestavení datového modelu databáze v systému MySQL, integraci jazyka ActionScript s databázovým serverem a návrh grafického uživatelského rozhraní se základními podpůrnými funkcemi pro běh.

Klíčová slova: multimédia, encyklopedie, Adobe Flash CS3, ActionScript, MySQL

Abstract

The purpose of this diploma thesis is a design of multimedia encyclopaedia in modern technologies - the ActionScript 3.0 object-oriented programming language in the Adobe Flash CS3 environment.

The first section deals with multimedia and encyclopaedia term definitions. Description, history and features of the used technologies and their comparison with alternative tools follows.

The main importance insist on proper multimedia program implementation based on a database model composition in the MySQL system, ActionScript language integration with a database server and a graphical user interface design with basic supportive runtime functions.

Keywords: multimedia, encyclopaedia, Adobe Flash CS3, ActionScript, MySQL

Obsah

Abstrakt	v
Obsah	vii
Seznam tabulek	viii
Seznam obrázků	ix
Zkratky	x
Úvod	1
1 Multimediální encyklopedie	3
1.1 Encyklopedie	3
1.1.1 Definice	3
1.1.2 Obecná charakteristika	3
1.1.3 Historie a současnost	4
1.2 Multimédia	5
1.2.1 Multimediální databáze	6
2 Použité technologie	7
2.1 MySQL	7
2.1.1 Porovnání metod pro ukládání a přístup k datům	9
2.1.2 Databázové modely	11
2.1.3 Porovnání s konkurenčními systémy	12
2.2 Adobe Flash CS3	14
2.2.1 Historie	15
2.2.2 Klíčové vlastnosti	15
2.2.3 Ochrana obsahu	16
2.2.4 Flash a současnost	18
2.2.5 Rich Internet Applications	18
2.2.6 Srovnání s konkurencí - Microsoft Silverlight	21
2.3 Action Script 3	23

2.3.1	Historie	23
2.3.2	Hlavní rysy jazyka	24
2.3.3	Základní datové typy	24
3	Realizace aplikace	26
3.1	Motivace vzniku programu	26
3.2	Požadavky zadávající firmy	26
3.3	Programovací prostředí	27
3.4	Multimediální titul	29
3.5	Návrh aplikace	29
3.6	Návrh a realizace dílčích problémů	31
3.6.1	Návrh databáze	31
3.6.2	Propojení aplikace s databází	34
3.6.3	Grafické zobrazované elementy	37
3.6.4	Prefetch dat	42
3.6.5	Spouštění offline verze z DVD	42
3.7	Zhodnocení a zdůvodnění užitých technologií	43
	Závěr	46
	Literatura	49
	Přílohy	50

Seznam tabulek

2.1	Porovnání MyISAM a InnoDB	10
2.2	Konkurenční databázové systémy	13

Seznam obrázků

2.1	Logo MySQL	7
2.2	Databázové modely	11
2.3	Logo společnosti Adobe a programu Flash	14
2.4	Logo balíku Creative Suite 3	14
2.5	Logo JavaFX	20
2.6	Logo Google Web Toolkitu	20
2.7	Logo Microsoft Silverlight	22
2.8	Ikona ActionScriptu	23
3.1	Schéma aplikace	29
3.2	Diagram toku dat	30
3.3	ERD diagram databáze	32
3.4	Šablona aplikace	38

Zkratky

Ajax	Asynchronous JavaScript and XML
CAT	Computerized Axial Tomography
BSD	Berkeley Software Distribution
CD	Compact Disc
CSS	Cascading Style Sheets
CMR	Customer Relationship Management
DBD	Berkeley DataBase
DVD	Digital Versatile Disc
ECMA	European Computer Manufacturers Association
EXIF	Exchangeable Image File Format
FLV	FLash Video
GIF	Graphics Interchange Format
GPL	General Public License
GUI	Graphical User Interface
HTML	HyperText Markup Language
ID3	IDentify MP3
InnoDB	Databazovy stroj od firmy Innobase
JPG/JPEG	Joint Photographic Experts Group
MMS	Multimedia Messaging Service
MP3	MPEG-1 Audio Layer 3
MyISAM	MySQL Indexed Sequential Access Method
MDA	Personal Digital Assistant

Pokračování na další straně

PAMPA-J	Portable Apache, MySQL, PHP Application plus Jaxer
PHP	Personal Home Page / PHP: Hypertext Preprocessor
PNG	Portable Network Graphics
RIA	Rich Internet Application
SQL	Structured Query Language
SSL	Secure Socket Layer
SWF	ShockWave Flash
USB	Universal Serial Bus
WMV	Windows Media Video
WMA	Windows Media Audio
XAML	Extensible Application Markup Language
XML	Extensible Markup Language
XMP	Extensible Metadata Platform

Úvod

V dnešní době používá internet velká část lidské populace a počet uživatelů do něj přistupujících se den ode dne zvyšuje. Trendy ukazují, že i desktopové programy se pomalu začínají přesouvat do online prostředí v Rich internet aplikacích. Výhodou pro uživatele je datové úložiště dostupné z jakéhokoli místa na Zemi přes webovou aplikaci.

Na internetu vznikají i díla neziskových organizací na poli encyklopedií, příkladem je Wikipedia. Projekt sdružuje množství nadšenců, kteří dobrovolně přispívají do mnohojazyčné webové encyklopedie se svobodným obsahem.

Tato diplomová práce prezentuje projekt encyklopedie, běžící v prostředí Adobe Flash CS3 s daty uloženými v databázovém systému MySQL. Program je vytvořen na základě požadavků zadávající komerční firmy FoxCom s.r.o. Cílem bylo vytvořit digitalizovaný syntetický portál, jež multimediálním obsahem a snadným uživatelským přístupem osloví své zájemce. Jednotlivá hesla by měla obsahovat text a další ilustrační materiál (foto, kresba, video), budou provázána hyperlinky mezi sebou a dále se souvisejícími tématy, vazbami na čas, místo, autory, techniky a bibliografii pramenů. Počítá se i s aplikací na DVD pro platformu Windows.

Text definuje pojmy multimédia a encyklopedie. Dále popisuje použité technologie – databázový systém MySQL, platformy pro vytváření multimediálních prezentací a programů Adobe Flash ve verzi CS3 a objektově orientovaný

programovací jazyk ActionScript 3.0 včetně jejich hlavních vlastností, historie a srovnání s konkurenčními řešeními. Malá část je věnována i tvorbě aplikací *RIA*.

Největší důraz je kladen na popis a návrh všech částí aplikace, od požadavků zadávající firmy, přes návrh databáze, grafického rozhraní, propojení aplikace s databází, až po problém spouštění offline verze z DVD.

Kapitola 1

Multimediální encyklopedie

1.1 Encyklopedie

1.1.1 Definice

[9] *Encyklopedie (řecky **enkyklios** okružní + **paideia** výchova) je strukturované, zpravidla objemné dílo, které se pokouší zevrubně představit lidské poznání týkající se jednoho, více, anebo všech oborů.*

[8] *Reference work that contains information on all branches of knowledge or that treats a particular branch of knowledge in a comprehensive manner.*¹

1.1.2 Obecná charakteristika

Encyklopedie může být definována pomocí čtyř základních vlastností: předmětu, účelu, metody či členění, procesu vzniku. Má za cíl přinést důležitá shromážděná poznání z určitého oboru, tudíž v tomto smyslu se liší podrobností a záběrem. Účel díla může ovlivnit i cílový okruh čtenářů, dětské encyklopedie se s tématem vypořádávají jinak než všeobecné encyklopedie či odborné slovníky.

¹Informační dílo, které obsahuje znalosti všech vědních oborů nebo se snaží dopodrobna zpracovat vědomosti určitého oboru komplexní metodou.

Encyklopedie mohou být obecné, obsahující hesla z různých oborů, nebo se mohou specializovat na určitý obor či okruh hesel, na určitou zeměpisnou či kulturní oblast.

Existují dva rozdílné způsoby organizace tištěných encyklopedií: abecední řazení a hierarchické řazení podle kategorií. Díla se také liší ve volbě svých autorů a editorů, způsobu shromažďování, ověřování a sepisování jednotlivých hesel.

Na rozdíl od slovníku, který přináší pouhou definici hesla, se encyklopedie snaží přiblížit předmět hlouběji a podrobněji. Text může být také lépe strukturován a přinést stručnou, ale pro čtenáře dostatečnou informaci o daném hesle, doplněnou o ilustrace, mapy, přehledy, grafy, bibliografii.

1.1.3 Historie a současnost

První encyklopedická díla začala vznikat již ve starověku. Největší vývoj lze však zaznamenat v období renesance a osvícenství, kdy chtěli lidé co nejdokonaleji popsat svět kolem sebe. Jmenoval bych *Encyklopedie aneb Racionální slovník věd, umění a řemesel* Denise Diderota a jeho spolupracovníků, velkou anglickou encyklopedii *Britannica* a také český počín v době národního obrození – *Ottův slovník naučný*. Ve 20. století bylo nejvýznamnějším počinem vydání *Velké sovětské encyklopedie*. Byla to mnohasvazková knižní díla, která si mohli dovolit jen bohatí.

S příchodem počítačové techniky se začaly encyklopedie digitalizovat. Data se vešla na CD a obsah byl doplněn zvuky a videoukázkami. [24]Nejnovějším fenoménem je Wikipedia – mnohojazyčná webová encyklopedie se svobodným otevřeným obsahem. Ten tvoří a ověřují nadšenci i odborníci, rozšířil se tedy okruh autorů. Elektronické encyklopedie obvykle spojují obě metody organizace řazení, krom toho umožňují vyhledávání, indexaci a křížové odkazy v míře, která je u tištěných děl nepředstavitelná.

1.2 Multimédia

[29] "*Médium zprostředkovává informace mezi zdrojem a příjemcem.*"

V tomto smyslu chápeme například rozhlasové vysílání jako médium. Postupně, s rozvojem dalších věd a s vývojem moderní komunikační techniky, se pojem *médium* začal používat ve třech významech:

- přenosový kanál informace
- materiálně-energetický nosič znaků
- soustava znaků, t.j. určitá zpráva se syntaktickou, sémantickou a pragmatickou strukturou

Spojováním více médií, ve smyslu prvního a druhého významu dostáváme zařazení, kterému se začalo říkat *multimédium*.

[15] Multimédia jsou oblast informačních a komunikačních technologií, která je charakteristická sloučením audiovizuálních technických prostředků s počítači či dalšími zařízeními. Forma tohoto média obsahuje kombinaci textu, zvuku, obrázků a fotografií, animací, videí a interaktivního obsahu. Počítačová zařízení jsou pro prezentaci často nezbytná, a to nejen klasické počítače, ale například i mobilní telefony, domácí multimediální centra na přehrávání videa a audia, projektory a přenosné přehrávače.

Multimédia lze rozdělit do dvou kategorií: na lineární a nelineární. Pojmem lineární mám na mysli projekci, kdy divák pouze konzumuje, co mu je ukazováno. Nelineárním bych nazval interaktivní obsah, který divák sám může ovlivňovat a v průběhu projekce pohotově reagovat. Příkladem jsou počítačové hry.

Multimedia můžeme využít v různých oblastech lidské činnosti. Několik příkladů následuje.

- Inženýrské obory – 3D simulace nových výrobků před jejich reálnou výrobou, modelování toků a proudění
- Armáda – letecké simulátory
- Obchod – grafy a prezentace výsledků práce, webové stránky společností jako informační kanál pro zákazníky, předpovědní mapy
- Výchova a výuka – interaktivní tabule, prezentace výukových materiálů
- Medicína – CAT scan lidského těla, virtuální simulace rozšiřování nákazy v samotném těle nebo v geografických oblastech
- Zábava – využití 3D brýlí v kině, počítačové hry, laser a videoprojekce na diskotékách, speciální efekty ve filmech, MMS

1.2.1 Multimediální databáze

[16]Multimediální databáze je databázový systém, který dokáže spravovat multimediální data. Ta jsou většinou nestrukturovaná, vyznačující se velkým objemem (jedna položka může mít desítky až tisíce megabajtů). Příkladem jsou 2D obrázky, 3D modely, video, audio, dokumenty a jejich kombinace. Tato data obsahují často také metadata² (např. Exif u obrázků, ID3 tag u MP3). Dotazování a získávání informací z kolekcí uložených dat může probíhat podle popisu, obsahu nebo metadat.

²Metadata (z řeckého *meta-* mezi, za + latinského *data* to, co je dáno) jsou strukturovaná data o datech[13]

Kapitola 2

Použité technologie

2.1 MySQL

[17]MySQL je databázový systém založený na relačním modelu dat (2.1.2), tedy na predikativní logice, s přirozenou reprezentací zpracovávaných dat a snadném zpracování a definování vazeb.



Obrázek 2.1: Logo MySQL

Zdroj: <http://www.mysql.com/common/logos/mysql-167x86.png>

Jak už vyplývá z názvu, je použit strukturní dotazovací jazyk SQL. Je to standardizovaný jazyk implementovaný do většiny dnešních databázových systémů. S jeho pomocí lze snadno vytvářet nové databáze a tabulky v nich, spravovat je, stejně tak jako přistupovat k jednotlivým položkám v tabulkách, které se dají načíst; přidat, změnit nebo smazat.³

³metody INSERT, UPDATE, DELETE

Program běží jako serverová aplikace a poskytuje multiuživatelský přístup k datům v databázích. Je vytvořený švédskou firmou MySQL AB⁴ a poskytuje dvojí licencování - bezplatné pod licencí GPL a komerční placenou licenci, ke které je poskytována podpora. Tento databázový systém je multiplatformí, není tedy problém běhu i přístupu k němu z rozličných systémů (Linux, FreeBSD, Windows, Mac OS X, Solaris, Symbian a dalších). Poskytuje i multiuživatelský přístup k databázím, což je dnes samozřejmostí.

Pro snadnou implementovatelnost, výkon a především volnou licenci má vysoký podíl v databázových systémech používaných na internetu a intranetu. Velmi oblíbená a často nasazovaná je kombinace Apache, MySQL a PHP jako základní software webového serveru. Knihovny pro přístup k databázím jsou dostupné pro všechny podstatné programovací jazyky. K administraci se používá nejen příkazová řádka, ale jsou k dispozici i *GUI* aplikace pro snazší vizuální ovládání. Od počátku byl systém optimalizován na rychlost, bohužel za cenu některých zjednodušení funkčnosti. V dnešní době již důležité funkce nechybí.

Přehled podporovaných vlastností:

- cizí klíče
- transakce
- podpora různých znakových sad (včetně Unicode) a časových pásem v datech
- poddotazy
- uložené procedury
- trigger⁵

⁴nyní dceřiná společnost Sun Microsystems – americký výrobce počítačů, komponent, software a služeb v informačních technologiích s velkou zásluhou na poli inovativních platform. Společnost Oracle Corporation začala s akvizicí Sun Microsystem v dubnu 2009. K dokončení transakce je potřeba souhlas patřičných úřadů a očekává se, že k tomu dojde v létě. Oracle vyvíjí vlastní relační databázový systém a *CMR* systémy. Akvizicí získá i další klíčovou technologii a tou je Java.

⁵procedura, která je automaticky spouštěna jako reakce na specifikovanou akci v databázi

- pohledy
- práce s metadaty
- meziplatformí podpora
- nezávislé metody pro ukládání a přístup k datům (viz 2.1)
- podpora SSL
- vyrovnávací paměť pro dotazy
- fulltextové indexace a vyhledávání (MyISAM)

Jednou z mála nevýhod oproti konkurenčním enterprise řešením (viz 2.1.3) je neschopnost paralelizovat jeden dotaz do databáze na více strojích nebo jádrech procesoru, takže tento systém není doporučený pro data warehousing.⁶

2.1.1 Porovnání metod pro ukládání a přístup k datům

MySQL má na rozdíl od jiných placených enterprise systémů další zajímavou vlastnost, a tou je větší počet metod pro ukládání dat. Umožňuje to uživateli vybrat si pro každou tabulku ten nejefektivnější způsob. Tyto programové implementace metod mohou být jednoduše dynamicky spouštěny za běhu. Mezi dvě nejdůležitější patří MyISAM a InnoDB⁷, jejich porovnání je v tabulce 2.1.

⁶skladiště všech elektronických dat z organizace, která využívají různá oddělení ve společnosti pro vytváření reportů a analýz

⁷<http://en.wikipedia.org/wiki/InnoDB> , <http://en.wikipedia.org/wiki/MyISAM>

Tabulka 2.1: Porovnání MyISAM a InnoDB

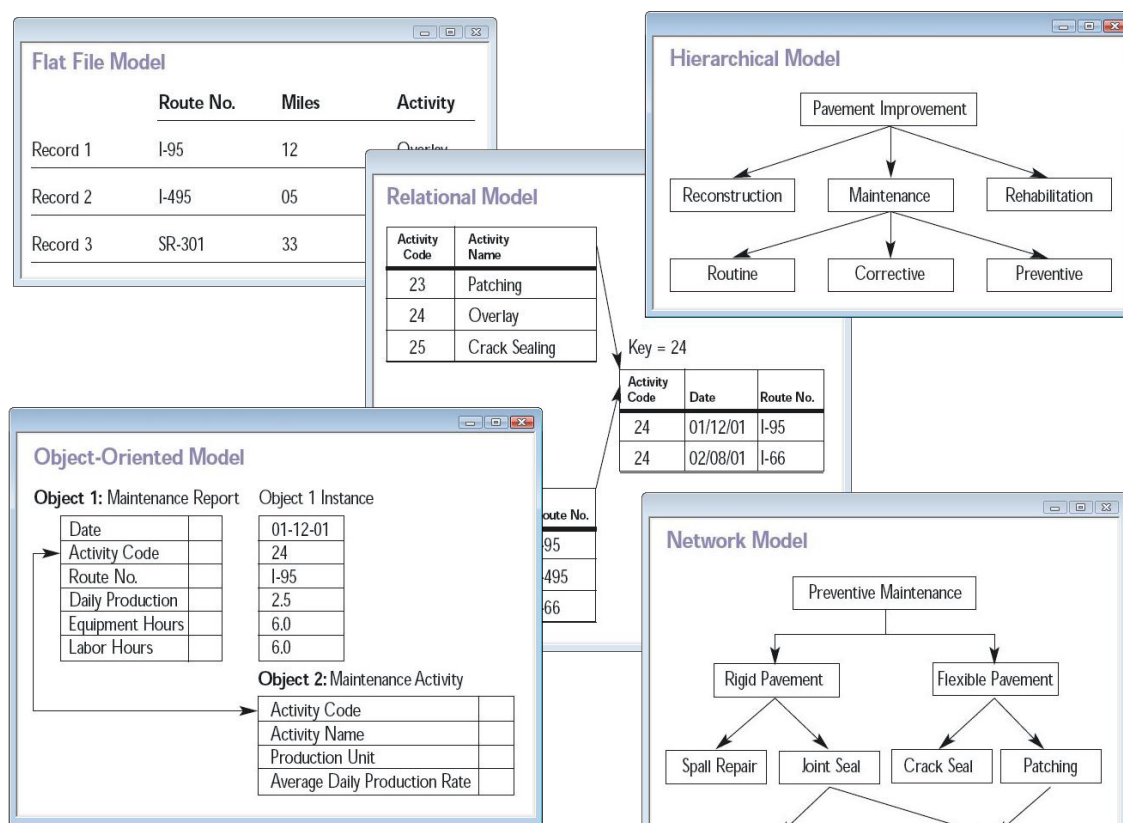
	InnoDB	MyISAM
Zotavení z pádu při nečekaných událostech	znovu provede logované transakce	kontrola a případná oprava všech indexů tabulek, kde proběhla změna, která se nestihla zapsat na disk
Dostupnost a spolehlivost	časově neměnný přístup	snižuje se s narůstáním dat v tabulkách
Vyrovnávací paměť	vlastní metody, kombinace paměti pro sloupce a indexy	spoléhá se na operační systém
Kompresa	není podporována	úspora až 40% prostředků závislé na datech
Transakce	plná podpora	nepodporuje
Cizí klíče	plná podpora	nepodporuje
Fyzické uložení dat	podle pořadí primárního klíče	pořadí, v jakém byla data přidávána
+ Výhody	méně přístupů na disk a do paměti při čtení	při vyhledávání se musí projít celá tabulka
– Nevýhody	řízené ukládání na správné místo, přeuspořádání dat	snadné ukládání přidáváním dat na konec

Další metody:

- Archive – na historická data při malé náročnosti na místo
- DBD – podobné InnoDB s podporou transakcí
- MERGE – odstraňuje omezení velikosti MyISAM tabulek
- HEAP – uložená v paměti, proto je nejrychlejší, všechna data jsou však ztracena při odpojení od zdroje. Nepodporuje typy *auto_increment*, *blob* a *text*.

2.1.2 Databázové modely

[28]Následuje stručné představení databázových modelů s rozepsáním jejich výhod (+), nevýhod (-) či jiných vlastností.



Obrázek 2.2: Databázové modely

Zdroj: http://upload.wikimedia.org/wikipedia/commons/3/3b/Database_models.jpg

Hierarchický model – data uchovávána v souborech, nejvyšší úroveň má jen jeden prvek - kořen, ostatní prvky mají vždy jednu vazbu na nadřazený prvek (rodič) a žádnou, jednu nebo více vazeb na nižší úroveň (potomek), mezi prvky na stejné úrovni vazby neexistují

– nelze definovat vztahy kardinality M:N, při změně se musí reorganizovat celá struktura, běžné operace vkládání a rušení jsou složitými operacemi

Síťový model – založen na souborech a vztazích mezi záznamy, prvek může být propojen vazbou s více prvky z vyšší i stejné úrovně

+ standardizován roku 1972

– dodatečná modifikace vazeb je časově náročná

Relační model – založen na pevném matematickém základu - relační kalkul a algebra, relace - kartézský součin domén (množin hodnot atributů), oddělení dat od jejich implementace, symetrický přístup k datům (stejně přístupové mechanismy), klíčové i neklíčové atributy

Objektový model – přirozenější a věrnější popis skutečného světa, snadnější aktualizace dat, složité datové typy, objekt s atributy a metodami, zapouzdření dat, dědičnost, polymorfismus

2.1.3 Porovnání s konkurenčními systémy

[6] Srovnání parametrů s nejdůležitějšími konkurenčními systémy následují v tabulce 2.2. Všechny systémy podporují referenční integritu, Unicode kódování, trigger, uložené procedury a funkce, transakce, ACID – Atomicitu, Konzistenci, Izolovanost a Trvalost. Podporované operační systémy jsou nahrazeny pro stručnost jen prvními písmeny – W=Windows, X=Mac OS X, L=Linux, B=BSD, U=Unix, S=Symbian.

Atomicita – zaručení transakcí, buď se provedou všechny operace s daty, nebo nedokončené operace neovlivní původní datovou základnu

Konzistence – udržení konzistentního stavu před i po provedení transakce

Izolovanost – zamezení interference mezi paralelně probíhajícími transakcemi pro předcházení ztráty dat nebo jejich chybné interpretaci

Trvalost – data jsou zapsány do databáze až po úspěšném zpracování transakce

Tabulka 2.2: Konkurenční databázové systémy

Databáze	Firma	Licence	První vydání	Podporované operační systémy
DB2	IBM	proprietární	1982	W L U
Microsoft SQL Server	Microsoft	proprietární	1989	W
MySQL	Sun Microsystems	GPL a proprietární	1996	W X L B U S
Oracle	Oracle Corp.	proprietární	1979	W X L U
Postgre SQL	Postgre Global Development Group	BSD	1989	W X L B U

2.2 Adobe Flash CS3



Obrázek 2.3: Logo společnosti Adobe a programu Flash

Zdroj: <http://i.zdnet.com/blogs/adobe-logo.gif>,

http://www.lavatwilight.co.uk/images/logos/flash_logo_adobe1.jpg



Obrázek 2.4: Logo balíku Creative Suite 3

Zdroj: http://www.nts.cz/app//public/user/image/Hlavni_stranka/adobe_logo.png

[2]Flash je multimedialní platforma vytvořena firmou Macromedia, která je nyní vyvíjena a distribuována firmou Adobe Systems. Od svého uvedení v roce 1996 se Flash stal populární v přidávání animací a interaktivity na webové stránky. Nyní je především používán pro reklamní bannery, animace, webové komponenty na přehrávání videí a v poslední době i k vytváření *Rich Internet Application* (viz 2.2.5).

Flash dokáže pracovat s vektorovou i rastrovou grafikou, podporuje obousměrné *streamování*⁸ audia (MP3) a videa (vlastní formát FLV). Všechn flashový obsah je možný přehrávat na různých systémech, zařízeních a produktech, a to díky Adobe Flash Playeru – přehrávači. Tento přehrávač je dostupný pro převážnou většinu

⁸Streaming (z anglického *stream* – proud) je technologie kontinuálního přenosu audiovizuálního materiálu mezi zdrojem a koncovým uživatelem

webových prohlížečů jako *plugin*⁹, případně i v mobilních telefonech. Další možností zpřístupnění obsahu uživateli je vytvoření spustitelného souboru – projektoru, který v sobě uvedený Flash Player integruje společně s multimediálním obsahem.

2.2.1 Historie

Projekt vznikl v podstatě jako aplikace pro grafické tablety se jménem SmartSketch běžící na systému PenPoint OS. Program byl navrhnut, aby zjednodušil kreslení do počítače, aby bylo co nejvíce podobné kreslení na papír. Později byl program portován na Windows a Mac OS X, přesto na trhu nebyl úspěšný. V době, kdy se začal rozvíjet internet, si vývojáři uvědomili potenciál vektorové grafiky, která mohla konkurovat programu Macromedia Shockwave. Tým programátorů se rozrostl a soustředil na vývoj *pluginů* do webových prohlížečů, přehrávání okénko po okénku, grafický uživatelský nástroj na tvorbu a matematický podklad pro manipulaci s vektory.

V roce 1996 produkt převzala pod svá křídla firma Macromedia¹⁰, která tím spojila i své projekty a vydala Flash.

V roce 2007 vše převzala společnost Adobe a v zápětí zakomponovala Flash jako součást svého balíku pro grafiky a vývojáře do Creative Suite 3. Největší novinkou této prozatím předposlední verze je skriptovací jazyk ActionScript 3 (viz 2.3).

2.2.2 Klíčové vlastnosti

[23] Animace – kopírování a vkládání animací, pohybové cesty, efekty časové osy, obrazovky pro nelineární přehrávání

Nástroje na kreslení – křivky a objekty, nástroj pero, galerie základních tvarů, knihovna základních vizuálních objektů, transformace objektů

⁹zásuvný balíček

¹⁰severoamerická firma, která vyvíjela převážně software pro práci s grafikou a navrhování webových stránek

Práce s textem – vstupní textová pole, podpora XML, formátování HTML dynamického textu, podpora kaskádových stylů CSS, Unicode kódování

Vývoj v jazyce Actionscript – pokročilý debugger, skriptovací asistent, panel akcí, komponenty uživatelského prostředí, kódovací nástroje šetřící čas, komponenty pro čtení a práce s daty, integrace do webových služeb

Rastrové obrázky a filtry – efekty filtrů, režimy prolnutí, vyrovnávací paměť pro bitmapy, knihovna přechodů

Pokročilá integrace médií – podpora zvuku MP3, průhlednost (Alfa kanál), pracovní postup pro import videa, pokročilé exporty videa

Další vlastnosti – komprese, adaptivní anti-aliasing (minimalizace deformace obrazů rozmazáním hran) pro zlepšení čtení především malých textů, nápověda včetně online referenční příručky a výukových lekcí, zaznamenávání prezentací z obrazovky, vývoj aplikací s použitím připravených šablon rozhraní pro mobilní telefony a zařízení typu PDA a Palm

Nová verze CS4 vydaná v říjnu 2008 již plně začleňuje Flash do prostředí podobných produktů Adobe a nabízí velký počet nových funkcí – objektově založené animace, 3D transformace, inverzní kinematika s modelem kostry, procedurální modelování pomocí nástrojů *dekorace* a *rozprašovač*, editor pohybu, podpora metadat (XMP), barevná témata, převod animací do jazyka ActionScript.

Od další verze se očekává především nová verze jazyka ActionScript.

2.2.3 Ochrana obsahu

[18]Autoři flashového obsahu využívají možnosti, které jim Flash nabízí v oblasti animace a interaktivity, přesto si nepřejí, aby jejich práce byla snadno dostupná v podobě zdrojových kódů a obrázků, nýbrž pouze jako binární spustitelná aplikace. Jelikož celá práce je obsažena v jediném souboru SWF, který je dostupný přes web, je snadné ho stáhnout do lokálního umístění. Některé dekompilátory umožňují skoro plnou rekonstrukci původních zdrojových kódů, které byly využity při vytváření.

Cílem je snaha o zamezení dekompilování původního obsahu z těchto souborů. K tomu jsou k dispozici *obfuskátory*. Je to softwarová pomůcka, která se pokouší co nejvíce znesnadnit čitelnost zdrojového kódu člověku. Podmínkou smyslu existence obfuskátoru je současně to, aby zdrojový kód byl i po konverzi čitelný pro překladač nebo interpret svého respektivního jazyka. Používají se následující techniky:

- odstranění komentářů a dokumentaci v kódu
- odstranění formátování a prázdných znaků
- přejmenování identifikátorů proměnných, konstant a metod
- přidání specifických pastí pro dekompilátory

Výhody obfuskace kódu:

- + ochrana intelektuálního vlastnictví
- + snížení rizika vyžrazení soukromých informací v kódu (typicky hesla k databázím)
- + redukce velikosti aplikace

Nevýhody:

- nikdy není stoprocentní jistota, že s nastupujícími metodami reverzního inženýrství se k obsahu nedostane neoprávněná osoba
- problém s přejmenováním metod, které jsou linkované z jiných knihoven
- znesnadnění debugování¹¹ za běhu aplikace

Z důvodu poslední nevýhody se tedy obfuskátory používají až v době, kdy je kód dostatečně odladěn. Tyto metody se nepoužívají jen u Flashe, ale i u dalších jazyků, kde se výsledný program distribuuje ve formátu, který se teprve pouští přes nějaký interpret a kde hrozí nebezpečí zpětné rekonstrukce původních dat, například Java.

¹¹vychytání much

2.2.4 Flash a současnost

[3]Podle oficiálních zdrojů je Adobe Flash Player nainstalován na 99% počítačů připojených v internetu a ze 45-55% v nejnovější verzi (závislé na regionu).

Flash Lite [10]je optimalizované běhové prostředí pro mobilní a spotřebitelské elektronické přístroje. Umožňuje vytvořit a nabízet poutavé uživatelské rozhraní, přístup k RIA, video a dynamický webový obsah.

Mezi výhody patří snadnější manipulace s vektorovou grafikou, menší objem dat, zobrazení webových flashových prezentací bez potřebných úprav. Za největší výhodu oproti Java 2 Micro Edition považují lepší podporu pro psaní kódu na veškeré přístroje, které Flash Lite podporují.

Nevýhodou je menší dostupnost zařízení (ve srovnání s rozšířením přístrojů podporujících J2ME) – je dostupný teprve několik let, má horší grafický a zvukový výkon z důvodu větších hardwarových nároků.

2.2.5 Rich Internet Applications

[20] *RIA* je internetová podoba aplikací, které běží nativně v prostředí operačního systému. Typicky se k přenosu dat využívá AJAX¹², sandboxy¹³, virtuální stroje, nebo proprietární zásuvné moduly do webových prohlížečů (to je případ Flashe).

Klíčové vlastnosti

Dostupnost dat pro internetové vyhledávače a rychlý webový přístup pro uživatele

Rozšířená komunikace – optimalizovaný transportní protokol, asynchronní přenos dat, přednahrávání potřebných dat v době nečinnosti pro urychlení (prefetch)

¹²Asynchronní využití JavaScriptu a XML

¹³bezpečnostní mechanismus pro oddělení běžících programů. Ty běží ve vlastním paměťovém prostoru a mají omezené možnosti komunikace s okolními procesy

Komplexnost moderních řešení, která jsou těžší na programování a správu než klasické webové aplikace

Konzistence uživatelského rozhraní v rámci různých operačních systémů

Instalace a údržba zásuvných modulů a virtuálních strojů

Bezpečnost je vylepšená na rozdíl od klasických programů díky běhu v uzavřených paměťových prostorech, na druhou stranu pokusů o napadnutí systému je díky webovému přístupu více

Výkon se může navýšit při používání na rychlé síti, náročnost hardwaru na straně serveru nebo klienta se může lišit mezi implementacemi

[21]Adobe nabízí pro vývoj těchto aplikací kompletní balíček řešení, který se skládá z frameworku¹⁴, nástrojů na programování, serverových řešení a služeb a v neposlední řadě klientskou část v podobě Flash Playeru. Jako příklady aplikací mohu uvést Adobe Photoshop Express¹⁵ – online verzi nástroje pro práci s fotografiemi, a Adobe Kuler¹⁶ – pomocníka při přípravě barevných schémat nejen pro webové stránky.

Adobe Flex [4]je sada technologií, pro vývoj multiplatformních RIA. Narozdíl od Flashe, nemůžeme vytvářet animace, MovieClipy, kreslit apod. Na druhou stranu je připraveno spousta již hotových komponent (objektů) s vlastní grafikou, vlastnostmi a metodami, které nám ušetří spoustu práce při vývoji vlastní aplikace.

¹⁴softwarová struktura, která slouží jako podpora při programování, vývoji a organizaci jiných softwarových projektů

¹⁵<https://www.photoshop.com/express/landing.html>

¹⁶<http://kuler.adobe.com/>

Další platformy pro tvorbu RIA

JavaFX [12] je platforma pro RIA, která může běžet na různých zařízeních. Poslední verze (JavaFX 1.1 z února 2009) umožňuje vytváření aplikací pro počítače, webové prohlížeče a mobilní telefony. Plánuje se podpora pro herní konzole, set-top boxy a Blue-Ray přehrávače. Jednoduchá je instalace aplikací, a to pouhým přetažením z webového prohlížeče do systému. Využívá se především vektorová grafika, pro převod z rastrů existuje mnoho konverzních filtrů.



Obrázek 2.5: Logo JavaFX

Zdroj: http://files.myopera.com/fchoong/blog/javafx_logo_color_1.jpg

Google Web Toolkit [11] je opensource framework pro programovací jazyk Java, který dovoluje vývoj a údržbu komplexních webových aplikací založených na AJAXu. Vše se jednoduše naprogramuje v Javě na libovolné systémové platformě a za pomoci cross-kompilátoru, kompilátoru mezi platformami, se převede na JavaScriptový kód, který je vysoce optimalizovaný. Zdůrazňuje znovuvyužitelnost kódu, asynchronní volání vzdálených procedur, správu historie, vytváření záložek a přenositelnost mezi webovými prohlížeči.



Obrázek 2.6: Logo Google Web Toolkitu

Zdroj: <http://code.google.com/intl/cs/webtoolkit/images/gwt-logo.png>

2.2.6 Srovnání s konkurencí - Microsoft Silverlight

[14, 22, 27] Microsoft Silverlight je platforma navenek podobná Adobe Flash. Nabízí podobné vizuální možnosti, integruje multimédia, grafiku animace a interaktivitu do jednotného běhového prostředí. Program jako takový není kompilovaný, ale reprezentovaný jako text XAML. Podporuje přehrávání WMV, WMA a MP3. Dynamické načítání XML obsahu provádí přes Javascript podobným způsobem jako AJAX. Jako programovací jazyk je použit převážně .NET.

Verze 2.0 byla vydána v říjnu 2008. Hlavní nevýhodou Silverlightu je, že přišel na trh v době, kdy už má Flash vybudovanou slušnou pozici u vývojářů. Plugin je pro prohlížeče pod operačními systémy Windows a Mac OS X, dokonce pro mobilní telefony se systémy Windows Mobile a Symbian. Jeho otevřená implementace *Moonlight* pro operační systém Linux je však pořád ještě ve vývoji, očekávané uvedení verze 2.0 je plánované na září 2009.

Důležité výhody(+) a nevýhody(-) MS Silverlight 2.0 v porovnání s Flashem

- + výstupní kód je XAML – snadno čitelný
- + jednodušší debugování
- + lepší, bohužel placené, programovací rozhraní Microsoft Studio, více podporovaných jazyků – Visual Basic, .NET, C#, Javascript, JScript, IronPython, IronRuby
- + lepší podpora služeb od Microsoftu – datové webové úložiště pro aplikace a multimediální data
- podpora linuxu a méně používaných webových prohlížečů
- testovací nástroje pro zjištění slabin v přístupnosti aplikace pro postižené lidi
- nedokáže zpracovávat zvukové signály
- editování rastrových obrázků na úrovni pixelů, efekty (stín, rozmazání)

– větší datová náročnost z důvodu nulové komprese a velkého objemu XML souborů,
přesto podpora kompresního algoritmu ZIP pro části vyžádaných dat

* časová osa definována v reálném čase(sekundy) na rozdíl od Flashe(snímky)

Nejnovější verze 3.0, která je stále v beta stádiu vývoje (květen 2009), přidává podporu kodeku h.264, tedy video ve vysokém rozlišení i přes celou obrazovku, běh aplikace mimo prostředí webového prohlížeče, 3D grafiku a drobnosti týkající se přístupnosti vytvářených aplikací. Tyto informace nejsou obsaženy v předcházejícím přehledu, jelikož se před vydáním finální verze mohou změnit.

Obě platformy tedy nabízejí podobné možnosti a ukáže jen čas, která bude více využívána. Z konkurenčního boje může těžit jak uživatel, tak i vývojář. Dá se předpokládat velké rozšíření podobného obsahu na webové stránky, což se nemusí líbit příznivcům snadného striktního HTML, případně PHP.



Obrázek 2.7: Logo Microsoft Silverlight

Zdroj: <http://upload.wikimedia.org/wikipedia/en/d/db/Silverlight.png>

2.3 Action Script 3



Obrázek 2.8: Ikona ActionScriptu

Zdroj: http://upload.wikimedia.org/wikipedia/en/0/0f/ActionScript_icon.png

[1, 25, 7] Actionscript je objektově orientovaný programovací jazyk. Primárně se používá při vývoji v platformě Adobe Flash, dále v databázových aplikacích (Alpha Five) nebo v aplikacích na ovládání robotiky (Make Controller Kit). Jazyk je založen na ECMAScriptu, což je skriptovací jazyk standardizovaný Evropskou asociací výrobců počítačů (ECMA). ECMAScript je široce používán na webu a často je zaměňován za JavaScript nebo JScript, což jsou ale dva jeho dialekty.

Kód ActionScriptu je vykonáván pomocí ActionScript Virtual Machine¹⁷, který je zabudován ve Flash Playeru. Nový Virtual Machine pro verzi 3.0 dramaticky navýšil výkon a je nyní při vykonávání kódu až 10-krát rychlejší než starší verze.

2.3.1 Historie

Vývoj jazyka v podstatě kopíroval potřeby využití interaktivity podle toho, co bylo dostupné v té které verzi Flashe a Flash Playeru. Z počátku byl omezen na metody zastavení a puštění přehrávání animace a ovládání událostí myši. Do dnešní doby se rozvinul v objektovou formu. Základní milníky vývoje programovacího jazyka a jeho vlastností následují:

ActionScript 1.0 – rok 2000, deklarace uživatelských procedur a funkcí, vlastní proměnné, editor na psaní kódu

¹⁷virtuální stroj pro ActionScript

ActionScript 2.0 – rok 2003, lepší vybavení pro větší a komplexnější aplikace, kontrola syntaxe, kontrola kódu při kompilaci, zavedení tříd a interface¹⁸, kód se píše přímo k vlastnostem objektů

ActionScript 3.0 – rok 2006, od základu restrukturalizovaný jazyk, omezená podpora hardwarové akcelerace, sjednocení psaní kódu do prvního snímku časové osy, kontrola při kompilaci i při běhu aplikace, balíčky, jmenné prostory, regulární výrazy, sjednocení systému pro manipulaci s událostmi

2.3.2 Hlavní rysy jazyka

Běhové výjimky – lepší podpora hlášení chyb programů a poskytnutí těchto informací programátorovi pro jejich snadnější odstranění

Uzavřené třídy – efektivně naprogramované třídy, které dovolují jen využití jejich metod, nejdou nijak měnit, šetří to potřebnou paměť i čas vykonávání kódu

Regulární výrazy – nativní podpora regulárních výrazů pro snazší hledání v datovém typu STRING

Běhové typy – pro zlepšení systémové bezpečnosti se provádí kontroly při běhu programu, ulehčuje to práci programátorovi, zároveň spoří potřebnou paměť

Jmenné prostory – mechanismus pro kontrolu viditelnosti vnořených deklarací, podobné tradičnímu způsobu specifikace přístupových práv (public, private, protected)

2.3.3 Základní datové typy

Bohužel, na rozdíl od většiny jiných objektových jazyků, ActionScript nedělá rozdíl mezi primitivními datovými typy a odkazovaným typem, vše bere jako referenci. Reference je ukazatel na instanci třídy. Nevytváří se tedy kopie objektu, ale přistupuje se do toho samého paměťového prostoru. Při rušení objektu metodou

¹⁸rozhraní komunikace mezi třídami při dědění a využití polymorfizmu u metod

DELETE se odstraní pouze reference, pak za běhu programu Flash Player kontroluje všechny existující reference v paměti a pokud žádnou nenalezne, objekt je odstraněn *garbage collectorem*¹⁹. Z tohoto důvodu musí být v aplikaci pečlivě plánováno využití paměti.

Boolean – hodnoty pravda a nepravda

int a **uint** – 32-bitové celé číslo se znaménkem a bez

void – nedefinovaná hodnota

Number – 64-bitové číslo formátu základ a exponent

String – řetězec znaků

Null – speciální hodnota neodkazující na žádný objekt

Object – asociativní pole obsahující pár klíč-hodnota (String-jakýkoliv typ)

Array – pole jako seznam dat, všechna stejného typu

Vector – pole dat pevné délky, především pro rychlejší provádění operací nad daty

MovieClip – zobrazovaný animovaný prvek

Bitmap a **Shape** – vizualizované objekty - rastrový a vektorový obrázek

ByteArray – pole binárních dat

TextField – dynamické a interaktivní textové pole

SimpleButton – jednoduché interaktivní tlačítko podporující stavy kliknutí, přejetí myši a držení myši, poskytující tvar cílové aktivní plochy

Error – obecný chybový objekt, při výjimkách za běhu poskytuje informace

RegExp – regulární výrazy

Video – přehrávatelný objekt poskytující transportní protokoly na stažení a *streamování*

XML – přímá podpora XML formátu

¹⁹v překladu sběrač odpadu

Kapitola 3

Realizace aplikace

3.1 Motivace vzniku programu

Cílem zadavatele bylo vytvořit digitalizovaný syntetický portál, jenž osloví své zájemce multimediálním obsahem a snadným uživatelským přístupem. Umožní to propagovat téma zajímavou encyklopedickou formou, čímž zároveň zvýší všeobecné vzdělání dostupné nejen na internetu. Jednotlivá hesla by měla obsahovat text a další ilustrační materiál (foto, kresba, video), budou provázána hyperlinky mezi sebou a dále se souvisejícími tématy, vazbami na čas, místo, autory, techniky a bibliografii pramenů, aby umožnila hlubší studium problému odborníkům. Snahou bude vydat i kompletní obsah dostupný na nějakém médiu (CD, DVD) jakožto bonus k vznikající knižní podobě encyklopedie.

3.2 Požadavky zadávající firmy

- Graficky příjemná aplikace
- Jednoduchá a intuitivní ovladatelnost
- Zobrazení textu a obrázků

- Běh online a na platformě Windows
- Rychlý přístup k datům – *prefetch*²⁰
- Kryptování obsahu
- Časová osa a interaktivní mapy

Rozšířené požadavky, které nebyly prioritní:

- Videoukázky
- V případě potřeby komprese dat
- Aktualizace databáze offline verze
- Fulltextové vyhledávání

3.3 Programovací prostředí

Flash pro programování v jazyce ActionScript 3 nabízí editor obsažený v balíku CS3. Vše se dá jednoduše zobrazit v panelu *akce*. Editor podporuje barevná schémata na zabarvení klíčových slov, funkcí, řetězců String a komentářů. Flash dokáže i rozpoznat funkci, kterou právě píšou, a zobrazí odpovídající radu při psaní kódu. Dále nabízí funkce pro automatické formátování textu podle konvencí a kontrolu syntaxe. Pokud nalezne chyby, vypíše je v panelu chyb.

V levé straně panelu akcí se nachází kompletní seznam kategorií tříd, včetně jejich metod a vlastností. Obsahuje i kategorii *index*, kde jsou abecedně uspořádané všechny metody a funkce, po kliknutí na příkaz se do kódu doplní jeho správná konstrukce. Dalším nástrojem je *skriptovací asistent*. Pokud je zapnutý, po výběru metody ze seznamu kategorií se zobrazí v horní části nad samotným skriptem textová pole a volby, které jsou pro danou položku k dispozici, a je potřeba je vyplnit (nadefinovat).

²⁰přednahrávání potřebných dat v době nečinnosti

Tím však končí výčet možností, které tento editor nabízí. Jelikož jsem měl možnost programovat i v jiných prostředích (NetBeans, Eclipse, Microsoft Visual Studio, Zend Studio), musím konstatovat, že na plnohodnotnou práci je naprosto nedostačující. Může vyhovovat pouze designérovi, který vyvíjí převážně grafické elementy a přidává jen málo funkčnosti v rámci ActionScriptu k animacím.

Po dvou týdnech práce v editoru jsem zjistil, že zcela nevyhovuje. Nešlo kopírovat a vkládat text klávesovými zkratkami CTRL+INSERT a SHIFT+INSERT, známými již z prostředí DOSu, ale musely se používat standardní windowsové zkratky CTRL+C a CTRL+V. Při vytváření nových proměnných automaticky nedefinuje v deklarační části programu použitou třídu, ve které se proměnná nachází. Dobře ani nefunguje kompletace rozepsaného kódu pomocí CTRL+ENTER.

Vyřešil jsem nepříjemnou situaci následujícím způsobem. Adobe poskytuje vývojový nástroj, který nabízí interaktivní mód na debugování krokováním a inteligentní psaní kódu – Flex Builder. Je navržen především pro vizuální návrh uživatelského prostředí pro Flex aplikace. Pro mne však bylo důležité, že dokáže pracovat se skripty psanými v ActionScriptu s odstraněním chyb, které jsem vytýkal samotnému editoru v balíku CS3. Nástroj je založen na javovské platformě Eclipse, se kterou mám už bohaté zkušenosti. Samotný skript edituji pomocí Flex Builderu, ve Flashi ho mám definovaný jako hlavní třídu, takže se mi automaticky načítá k projektu. Ten sestavuji případně i publikuji pro web, nebo jako samostatně běžící aplikaci právě přes Flash.

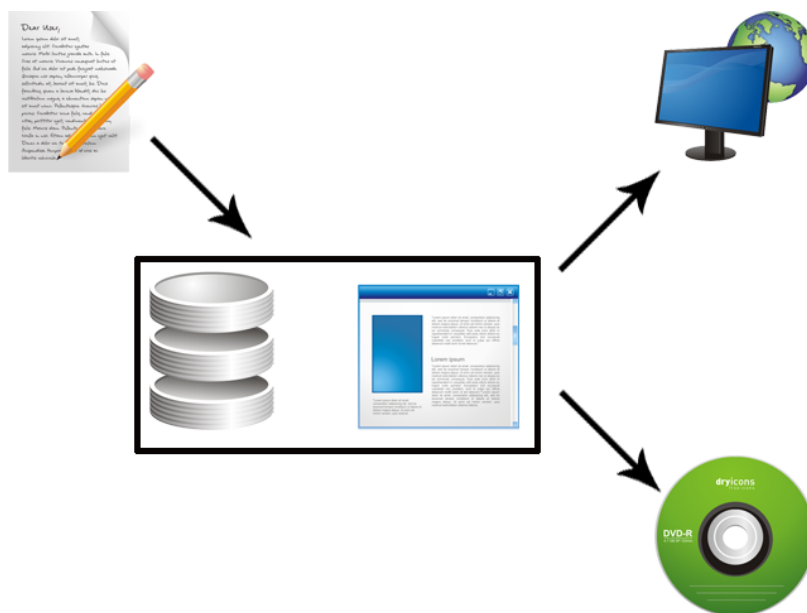
Dále musím vyzdvihnout kvalitně obsažnou nápovědu, která je na disk umístěna při instalaci, což umožňuje fulltextové hledání rychleji než při připojení na internetovou verzi dokumentace. Referenční příručka popisuje třídy v hierarchii, všechny jejich metody, konstanty a vlastnosti, mnohdy i ukázkou kódu s komentáři.

3.4 Multimediální titul

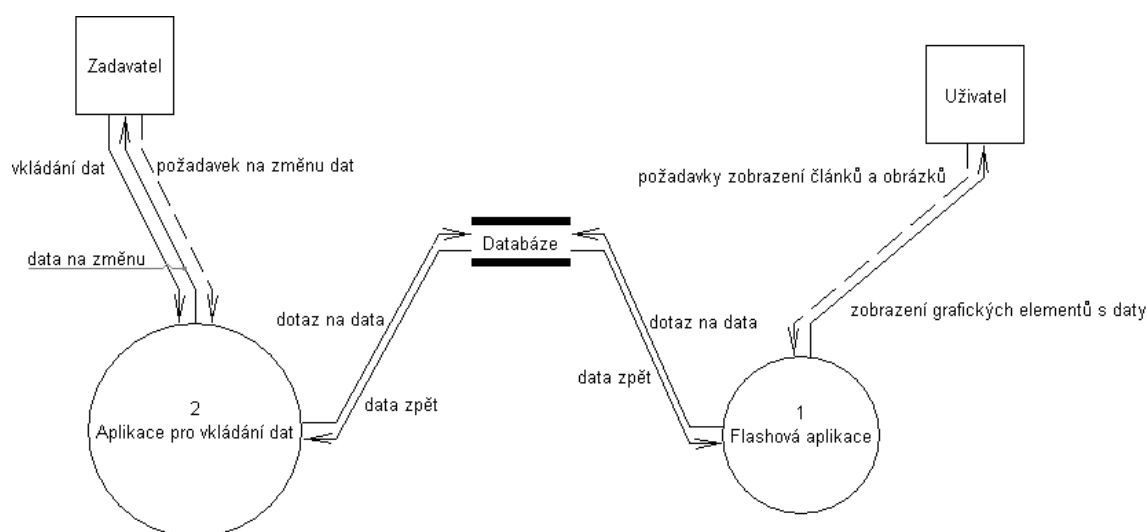
Tvorba multimediálního projektu s sebou nese velké nároky na osoby, které se na něm podílejí. Seznam se velice podobá produkčnímu týmu u filmu. Dlouhý čas také zabere uspořádání a vkládání dat do encyklopedie. Manažer již rozhodl, jaké bude zaměření, pro kterou platformu bude vyvinut a na kterém trhu se má titul uplatnit. Já se starám pouze o návrh samotného systému a základní programovou implementaci. Mým cílem je vymyslet do programu možnou interaktivitu mezi všemi prvky a uživatelem, při snaze o co nejobecnější implementaci na různé platformy pro zvýšení možností distribuce výsledného titulu.

3.5 Návrh aplikace

Aplikace bude naprogramována ve dvou verzích – offline a online verze. Vlastní program bude v obou verzích stejný. Offline verze bude celá i s databází uložena na DVD, zatímco online verze poběží na serveru.



Obrázek 3.1: Schéma aplikace



Obrázek 3.2: Diagram toku dat

Z důvodu tvorby dvou verzí programu bylo využito prostředí Adobe Flash CS3. V ActionScriptu verze 3 je vytvořena aplikace, která bude přistupovat k datům uloženým v databázi. Výhodou exportu aplikací flashové platformy je jak program běžící ve Flash Playeru, který je zásuvným modulem většiny dnešních webových prohlížečů, čehož využijeme v online verzi, tak i export do programu spustitelného na Windows případně Mac OS X pro offline verzi. Už v programu si budu ošetřovat, odkud je spouštěn a podle toho nastavím potřebné parametry přístupu k databázi.

Výhodou oddělení samotných dat od aplikace je především možnost aplikování programu znovu v jiném typu encyklopedie, tedy běh nad jinými daty. Zároveň využitím databázového systému se zvýší rychlost přístupu díky optimalizovaným metodám práce s daty. Tento systém nabízí i kryptování dat, což bylo jedním z požadavků zadávající firmy. To je i důvod, proč jsem nepoužil HTML stránky s přechody přes odkazy. Informace jsou tedy zobrazovány jen přes aplikaci, zdrojová data nejsou pro uživatele snadná získat. Flash mi dále nabízí lepší grafické možnosti pro vlastní program.

Problémem bylo řešení offline verze, která má být celá komplexně na DVD. Pro platformu Windows jsem musel zařídit větší počet aplikací, které se po nastartování spustí. Nejprve bylo potřeba spustit databázový server lokálně

v paměti a poté do něj přesměrovat soubor s uloženou databází vyskytující se na DVD. Vlastní aplikace se pustí přes funkci *autorun*, případně se o start aplikace postará uživatel sám.

3.6 Návrh a realizace dílčích problémů

3.6.1 Návrh databáze

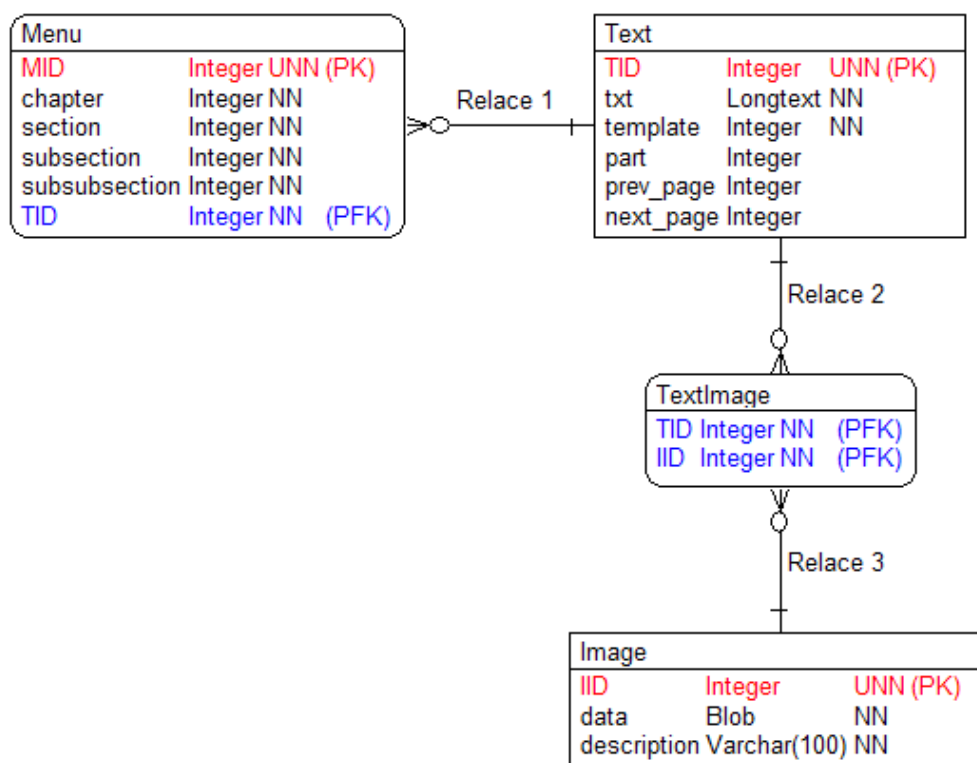
Samotnou databázi používám v podstatě jen jako pasivní úložiště dat. Preferuji především vyšší rychlost čtení, které mi dané řešení nabízí v porovnání se čtením dat uloženými na filesystému, zvláště s přihlédnutím k offline aplikaci pouštěné z DVD. Pro finální aplikaci předpokládám tisíce řádků textu a velké množství obrázků, proto bylo snahou využít optimalizované nástroje. Zároveň upotřebím vnitřní funkci databáze na kryptování a dekryptování dat, neboť zadávající firma chtěla zabránit snadnému úniku z encyklopedie.

V aplikaci se nepočítá se zapisováním dat do databáze, ale pouze se čtením. O nahrání dat se postará firma, pak se databáze optimalizuje na rychlost čtení a daný soubor s databázovými daty se uloží na server, nebo na DVD k aplikaci. Bude tedy muset být vytvořen program, který bude data ukládat ve správném formátu se všemi definovanými atributy nebo je případně upravovat. Předpokládám nějaký jednoduchý WYSIWYG²¹ editor.

Na uložení dat jsem použil již zmíněný databázový systém MySQL, konkrétně ve verzi 5. Strukturu databáze jasně a stručně popisuje ERD diagram, viz obrázek 3.3.

Databáze se skládá ze čtyř tabulek – Menu, Text, Image a TextImage. Následuje popis jednotlivých tabulek s ukázkovým SQL skriptem, který byl použit při jejich vytváření.

²¹akronym anglické věty „What you see is what you get“, česky „co vidíš, to dostaneš“



Obrázek 3.3: ERD diagram databáze

Menu – základní rozvrstvení textových dat do uspořádaných kapitol a částí, předpokládám pouze čtyřúrovňovou hierarchii vlastních článků. Z relace 1 se přenáší klíč TID z tabulky Text pro spojení textu do správného zařazení do Menu.

```

CREATE TABLE MENU (
  MID Int UNSIGNED NOT NULL AUTO_INCREMENT, --menu id
  chapter Int NOT NULL ,                    --kapitola
  section Int NOT NULL ,                    --část
  subsection Int NOT NULL ,                 --podčást
  subsubsection Int NOT NULL ,              --podpodčást
  TID Int NOT NULL ,                       --cizí klíč z tabulky Text
  UNIQUE (MID),                             --unikátní id
  PRIMARY KEY (MID,TID),                   --primární klíč na MID a TID
  INDEX IX_ (TID)
) TYPE = InnoDB;
  
```

Text – vlastní HTML formátovaný text, který se bude zobrazovat. Pokud se text nevejde na jednu stranu do formuláře, bude rozdělen na více stran, z toho důvodu jsou tam další entity *part*, *prev_page* a *next_page*.

```

CREATE TABLE TEXT (
  TID Int UNSIGNED NOT NULL AUTO_INCREMENT, --text
  id txt Longtext NOT NULL ,                --html formátovaný text
  template Int NOT NULL ,                   --zobrazovací šablona
  part Int ,                                --část vícestránkového textu
  prev_page Int ,                           --předchozí strana
  next_page Int ,                           --následující strana
  UNIQUE (TID),                             --unikátní id
  PRIMARY KEY (TID)
) TYPE = MyISAM;

```

Image – prostor pro obrazová data v *blob* formátu, neboli kolekci binárních dat uložených jako jednotná entita. Další entitou je textový popis obrázku.

```

CREATE TABLE IMAGE (
  IID Int UNSIGNED NOT NULL AUTO_INCREMENT, --image id
  data Blob NOT NULL,                       --vlastní data obrázku
  description VARCHAR(100) NOT NULL,        --popis obrázku
  UNIQUE (IID),                             --unikátní id
  PRIMARY KEY (IID)
) TYPE = MyISAM;

```

TextImage – spojovací tabulka pro kardinalitu M:N mezi tabulkami Text a Image

```

CREATE TABLE TEXTIMAGE (
  TID Int NOT NULL ,
  IID Int NOT NULL ,
  PRIMARY KEY (TID,IID),
  INDEX IX_ (TID),
  INDEX IX_ (IID)
) TYPE = InnoDB;

```

Kvůli zjednodušení určení primárního klíče záznamu v tabulkách jsem zvolil uměle vytvořené identifikátory (TID, MID, IID). Zároveň jsem využil i *auto_increment* modifikátor pro zajištění unikátní, postupně se generující identity pro jednotlivé řádky. Není to nejkorektní řešení po stránce relační algebry a návrhu, kde by se měl primární klíč vytvořit z více entit pro jednoznačnou identifikaci řádku. V praxi se však toto elegantní řešení využívá pro zkrácení SQL dotazů.

Domény atributů jsou většinou definovány jako *unsigned integer* neboli číslo bez znaménka, pro text je to *longtext*, nebo *varchar(100)* – variabilní řetězec maximální délky 100 znaků. Doména atributu dat obrázků je *blob*. Atributy, které považuji za důležité, jsem nastavil na *not null*, ty musí být vždy vyplněné.

Na primárním klíči tabulky Menu jsem vytvořil index. Hledání a třídění podle něj bude optimalizováno databázovým serverem. Index se i automaticky vytvořil u spojovací tabulky TextImage pro zrychlení práce.

Pro tabulky Menu a TextImage jsem nastavil strukturu na InnoDB (viz 2.1). Chtěl jsem využít výhody tohoto způsobu ukládání dat a přístupu k nim. Systém sám je optimalizovaný na rychlost. V porovnání s ostatními budou tyto dvě tabulky malé, nebudou obsahovat tolik informací a tak lze mít jejich větší část v paměti. Dále jsem potřeboval podporu cizích klíčů pro referenční integritu mezi dvěma relacemi. Pro tabulky Image a Text jsem využil strukturu MyISAM. Opět jsem chtěl využít možností programové implementace pro fulltextového vyhledávání nad daty, případně podporu komprese dat.

Relace 1 definuje vztah mezi tabulkami Menu a Text. Kardinalita vztahu je 1:N, tedy v jedné části hierarchické struktury menu se předpokládá více textů, ne naopak. Parcialita zároveň umožňuje nějakou část menu nechat prázdnou, zatímco každý článek musí být zařazen do určité části v hierarchii.

Relace 2 a 3 je v podstatě spojovací tabulka pro zajištění kardinality vztahu M:N mezi tabulkami Text a Image. Tedy existuje text, který patří k více obrázkům, a zároveň existuje obrázek, který patří k více textům.

3.6.2 Propojení aplikace s databází

Flashové prostředí v sobě přímo neobsahuje jakékoliv funkce, které by dovolovaly připojovat se k databázovému stroji a SQL dotazy získávat, nebo ukládat data. Je to z větší části zapříčiněno velkým bezpečnostním rizikem, kterému se Adobe snaží vyhnout. Takzvaný Cross server scripting neboli skriptování mezi servery je buď zakázáno, nebo ručně povoleno uživatelem, který tak přebírá všechna rizika z toho plynoucí.

Připojování Flashe k databázím se dá řešit externím skriptovacím jazykem, kterému Flash v HTTP POSTu předá parametry. Skriptovací jazyk pošle SQL dotaz

databázovému stroji a zároveň zpracuje výsledky do formátu, který Flash přijme. Skriptovacím jazykem často bývá PHP.

Flash → HTTP POST → PHP → SQL dotaz → PHP → HTTP → Flash

Toto řešení je provozuschopné na serverech, ale já jsem ho zavrhl z důvodu řešení offline verze, kde jsem se snažil omezit potřeby spouštění externích programů, v tomto případě webového serveru se skriptovacím interpretem PHP.

[5]Nalezl jsem opensource knihovnu AsSQL, která nabízí konektivitu na MySQL databázi z ActionScriptu a prostředí Flashe, bez potřeby použití skriptovací mezivrstvy. Toto řešení sice obchází bezpečnost flashového sandboxu, ale v mém řešení si to mohu dovolit, jelikož mám stále plnou kontrolu nad tokem dat. U online verze počítám s během na jedné doméně, u offline verze počítám s během na lokálním počítači a plný přístup do databáze neumožním.

Bohužel bezpečnostním rizikem této varianty je uživatelské jméno a heslo uložené v textové podobě v aplikaci. Jak už jsem psal v části 2.2.3, je tento kód i přes všechno úsilí potenciálně získatelný, tudíž čitelný. Mou snahou je samozřejmě úniku dat z aplikace zabránit. Proto jsem v databázi vytvořil dalšího uživatele s omezeným přístupem pouze pro čtení.

```
GRANT SELECT ON multimedia.* TO user@localhost IDENTIFIED BY 'password' WITH
MAX_USER_CONNECTIONS 1
--vytvoření nového uživatele pro tabulky databáze 'multimedia' jen pro čtení
--s uživatelským jmenem 'user', přistupujícího z lokálního počítače
--s heslem 'password' a maximálně jedním běžícím připojením
```

Zároveň jsem vytvořil pohled, pomocí něhož se uživatel dostane jen k datům pro něj určeným a znemožním tak případnému útočníkovi získání všech dat jedním příkazem.

```
CREATE VIEW vw_user AS                                --vytvoř pohled s názvem vw_user jako
SELECT *                                              --výběr všeho
FROM textimage                                       --z tabulky textimage
JOIN text ON textimage.tid = text.tid                --spoj na entitě tid s tabulkou text
JOIN image ON image.iid = textimage.iid             --spoj na entitě iid s tabulkou image
```

Vlastní příkaz na získání dat z databáze následuje.

```
SELECT * FROM vw_user WHERE tid = ... --vyber data z pohledu podle entity tid
```

Následující kód ilustruje vytvoření připojitele na databázi a zadání SQL dotazu pro vyhledání dat ve flashové aplikaci s komentáři, které vysvětlují jednotlivé akce.

```
//funkce na připojení k databázi
function ASQLconnect(sql:string):string {
    //definice posluchačů událostí připojení, chyb, správnosti příkazů, dat
    connector.addEventListener( SQLEvent.CONNECT, handleConnect );
    connector.addEventListener( SQLError.SQL_ERROR, handleError );
    connector.addEventListener( SQLEvent.SQL_OK, handleOK );
    connector.addEventListener(SQLEvent.SQL_DATA, handleData);
    connector.connect("server","user","pass","DB",3306); //připojení na server
    //s argumenty server, uživatel, heslo, jméno databáze, a port
}
function handleConnect(evt:SQLEvent, sql:string):void {
    trace("ASQL je připojeno"); //výpis do konzole
    connector.query(sql); //zadání vlastního SQL dotazu
}
function handleError(evt:SQLError):void {
    trace("Zachycena chyba "+evt.text); //výpis do konzole
}
function handleOK(evt:SQLEvent):void {
    trace("Dotaz proběh úspěšně"); //výpis do konzole
}
function handleData(evt:SQLEvent):void {
    trace("Přijata finální data"); //výpis do konzole
    //pro zjištění dat použít evt.data[index].nazev_sloupce
    image.loadBytes(evt.data[0].data); //obrazová data uložena do obrázku
    ta.text = evt.data[0].description; //text přidán do textového pole
    connector.disconnect(); //rozpojení s databází
}

ASQLconnect(.. SQL dotaz ..); //spuštění funkce
```

3.6.3 Grafické zobrazované elementy

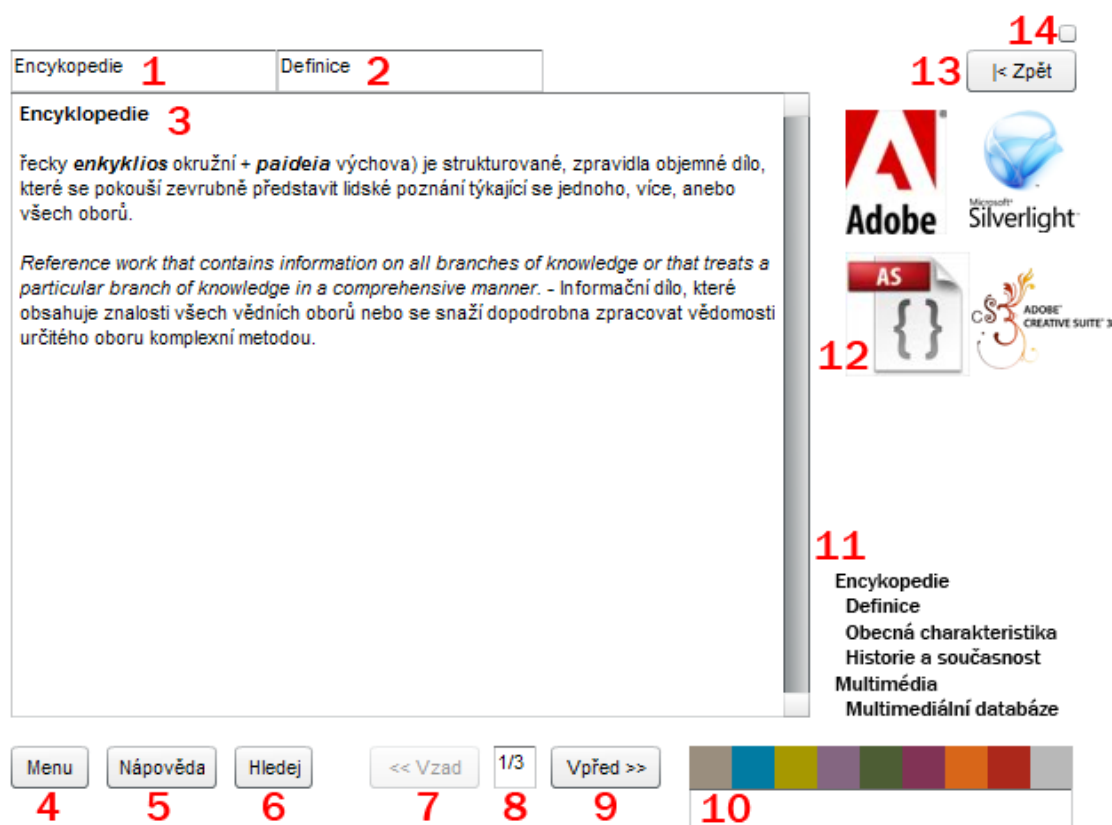
Šablony

Nejprve bylo potřeba rozvrhnout si grafické prostředí, do kterého budu vykreslovat všechny vizualizované objekty. Nadefinoval jsem si několik šablon, ve kterých budu prezentovat celý obsah encyklopedie.

První hlavní šablona zobrazuje menu, je to jakýsi hlavní rozcestník všech kapitol a podkapitol částí dokumentů. Po kliknutí na názvy se program přepne do šablony prezentující vlastní obsah.

Druhou a nejdůležitější šablonou je vlastní zobrazení textu a obrázků. Vše je dobře viditelné na ukázkové obrazovce, na obrázku 3.4. Červená čísla označují elementy, které v dalším výčtu rozepíši.

- 1 Textové pole, které zobrazuje aktuální kapitolu
- 2 Název aktuální části v kapitole
- 3 Vlastní část pole na zobrazení hlavního textu včetně odsazení odstavců, nastavení velikostí a vlastností písma
- 4 Tlačítko pro přímý přístup do menu encyklopedie
- 5 Odkaz na zobrazení stránky s nápovědou ohledně používání programu, tvůrců a obsahu
- 6 Tlačítko přepne na stránku s vyhledávacím polem, po zadání klíčového slova program pokusí najít v databázi relevantní informace a odkáže na ně uživatele
- 7 a 9 Tlačítka pro posouvání aktuálního vícestránkového textu o stranu vpřed a vzad
- 8 Textové pole zobrazující číslo aktuální strany / celkový počet stran tématu



Obrázek 3.4: Šablona aplikace

10 Barevné čtverečky budou po kliknutí myši odkazovat na všechny kapitoly zvlášť.

Po najetí myši na čtvereček se v textovém poli pod ním zobrazí název kapitoly

11 Související témata, hesla či klíčová slova pro snadnější orientaci v informacích, po kliknutí se daný článek zobrazí

12 Odkazy na obrázky, které se po rozkliknutí zvětší a vycentrují na obrazovce i s jejich popisem při částečném zakrytí okolní plochy pozadí díky alfa kanálu (průhlednosti)

13 Tlačítko pro skok na první článek dané kapitoly

14 Tlačítko pro přechod aplikace mezi zobrazením v okně a roztažením na celou plochu obrazovky

K dalším šablonám patří zobrazení vyhledávacího pole a relevantních výsledků, dále obrazovka s textovými informacemi o aplikaci a s nápovědou používání programu. Ty už nejsou pro vlastní uživatelské rozhraní tak důležité.

Obrázek 3.4 i program na přiloženém CD ukazuje zatím beta verzi aplikace. Jsou v ní použity pouze standardní elementy z knihovny Flashe. Při tvorbě výsledné aplikace předpokládám spolupráci s grafikem, který navrhne barevné schéma, nastavení velikosti textu i případné změny v rozmístění všech důležitých prvků.

Text

[26]Flash nabízí objekt *TextArea* na zobrazení textu. Umožňuje formátování HTML. Do HTML textu jde přímo ukládat i informace o stylování částí dokumentu, což bych musel řešit v každé stránce. Proto mojí snahou bylo oddělit samotný kód stránky a styl do separátních souborů CSS, jak je tomu běžné na webu. Samotná komponenta *TextArea* ale stylování tímto způsobem nepodporuje. Nalezl jsem však řešení v podobě třídy *doopingTextArea*, která rozšiřuje možnosti právě o tuto funkčnost. Je vždy nutné nejprve komponentě přiřadit styl, až poté zpracováváný text a ten nakonec vykreslit na scénu.

Není problémem vykreslení obrázků přímo k textu, ale to jsem vyřešil ukázkovými náhledy v části vedle textu.

Následuje ukázka kódu s komentáři.

```

var doopTA:doopingTextArea=new doopingTextArea(); //nová instance textového pole
var cssLoader:URLLoader=new URLLoader; //stahovač dat
var cssRequest:URLRequest=new URLRequest("default.css"); //požadavek na lokální soubor
cssLoader.addEventListener(Event.COMPLETE,cssLoaderComplete); //posluchač události
cssLoader.load(cssRequest); //příprava dat načtením z proudu

//funkce na získání stylu ze souboru a jeho aplikace na textové pole
function cssLoaderComplete(evt:Event):void {
    var css:StyleSheet=new StyleSheet; //instance proměnné stylu
    css.parseCSS(URLLoader(evt.target).data); //rozparsování vlastního CSS souboru
    doopTA.textField.styleSheet=css; //aplikace stylu na textové pole
}

doopTA.setSize(300,300); //nastavení velikosti
doopTA.move(10,10); //posun o 10 pixelů ve směrech X i Y
doopTA.htmlText="..."; //přiřazení textu

addChild(doopTA); //vykreslení na plochu

```

Obrázky

Obrázky v databázi mám uložené jen v jedné verzi, a to v maximální zobrazitelné velikosti. Není potřeba uměle navyšovat datovou velikost databáze další miniaturou nebo náhledem obrázku, který v grafickém rozhraní používám. V ActionScriptu mám naprogramovanou jednoduchou funkci, která mi dynamicky po načtení duplikuje obrázek a zmenší ho na požadovanou velikost. Tato operace není nikterak časově náročná.

Zároveň jsem se snažil využít grafické efekty na zobrazování obrázků. Jedním z nich je alpha blending neboli využití průhlednosti. Při najetí myši na miniaturu se nejen pod myší ukáže nápoděda toho, co se stane po kliknutí, ale vysvítí se i obrázek, který jakoby vystoupí z mlhy. Velký obrázek také vystupuje z pozadí, začíná na 100 procentní průhlednosti, až po úplnou neprůhlednost, po zavření opět zmizí stejným způsobem. Samozřejmě jsem mohl využít mnoho další efektů. Domníval jsem se, že tento bude nejlepší, jelikož se často používá v galeriích na internetu

a uživatelé jsou tedy na něco podobného zvyklí. Kód dokumentuje zobrazení obrázku s použitím alpha blendingu.

```
//definice animace změny v alfa kanálu z 0 na 1, trvající 1 sekundu
var fadeIn:Tween=new Tween(image2,"alpha",Strong.easeOut,0,1,1,true);
var fadeOut:Tween=new Tween(image2,"alpha",Strong.easeOut,1,0,1,true);
//definice opačné změny
fadeIn.addEventListener(TweenEvent.MOTION_FINISH, fadeInFinished); //posluchač
//události dokončení animace
fadeOut.addEventListener(TweenEvent.MOTION_FINISH, fadeOutFinished);

// funkce zobrazení obrázku
function showPicture(evt:MouseEvent):void {
    addChild(image2); //přidání obrázku na plochu
    fadeIn.start(); //spuštění animace
}

// funkce zmizení
function hidePicture(evt:MouseEvent):void {
    fadeOut.start(); //spuštění opačné animace
}

//logická proměnná pro zajištění úplného průběhu obou animací z důvodu
//sekundové prodlevy
var loaded:Boolean = new Boolean();
loaded = false;

// funkce na identifikaci ukončení zobrazení
function fadeInFinished(event:TweenEvent):void {
    trace("finished fading in"); //výpis do konzole pro kontrolu
    loaded=true; //načten
}

function fadeOutFinished(event:TweenEvent):void {
    trace("finished fading out");
    if (loaded){
        removeChild(image2); //zrušení objektu
        loaded = false;
    }
}
```

Tlačítka

Ke každému tlačítku se musí přiřadit posluchač události akce myši. Při aktivaci akce na tlačítku se spustí funkce ovladače události. Zdrojový kód ukazuje přepínání aplikace mezi zobrazením v okně a přes celou obrazovku.

```

var butfull_normal:Button=new Button; //nová instance tlačítka
var full:Boolean = new Boolean; //boolean proměnná
full = false;

butfull_normal.x=610; //nastavení X-ové pozice polohy na ploše
butfull_normal.y=10; //nastavení Y-pozice
butfull_normal.setSize(10,10); //velikost tlačítka
//posluchač události kliknutí na tlačítko
butfull_normal.addEventListener(MouseEvent.CLICK,butfull_normalHandler);

//ovladač události tlačítka
function butfull_normalHandler(event:MouseEvent):void {
    if (full==false) //pokud není v celobrazovkovém režimu
    {
        stage.displayState="fullScreen"; //změna zobrazení plochy přes celou
        obrazovku
        full=true; //je v celobrazovkovém režimu
    }
    else
    {
        stage.displayState="normal"; //změna zobrazení plochy do okna
        full=false; //režim v okně
    }
}
addChild(butfull_normal); //přidání tlačítka na plochu aplikace

```

3.6.4 Prefetch dat

Prefetch neboli přednahrávání potřebných dat je v aplikaci řešen velice jednoduše. V paměti držím pořadí všechny šablony. Jelikož vím, které prvky budou dostupné uživateli na grafickém rozhraní, znám přesně elementy, se kterými může interagovat. Takže vždy po nahrání aktuálních dat na šablonu a její zobrazení začnu na pozadí do pomocných proměnných stahovat z databáze další možná data, zároveň ruším proměnné s daty, která si uživatel nevybral.

3.6.5 Spouštění offline verze z DVD

[19]Nejdůležitější částí pro umožnění fungování offline aplikace bylo zajištění běhu databázového serveru lokálně a dále přesměrování databázových souborů s daty do spuštěného serveru. Tuto důležitou funkčnost mi nabídla skupina programů

s názvem PAMPA-J.²² Je to první přizpůsobitelné a přenosné prostředí složené z programů s otevřeným kódem - webového serveru Apache, skriptovacího interpretu PHP, databázového serveru MySQL a serveru Aptana Jaxer uzpůsobeného na interpretaci Ajaxu, vše běžící na platformě Windows. Kolekce přináší malý program běžící jako tray ikona, kliknutím se dá měnit nastavení, spouštět a zastavovat procesy. Kompletní nastavení je v jediném textovém souboru. Hlavním smyslem vzniku tohoto programu bylo vytvoření jednotného prostředí spustitelného z USB flash disku, CD nebo DVD, dovolujícího vývojářům distribuovat své demo aplikace, software a portály bez omezení.

Vlastní spouštění aplikace zařídím přes funkci *autorun*, která je implementovaná ve Windows, aktivuje se vložením CD nebo DVD. Mám na výběr spustit verzi aplikace v projektoru (spustitelný EXE soubor), nebo aplikaci pod prohlížečem, který má uživatel na svém počítači. Z toho důvodu přidám na DVD i instalaci zásuvného balíčku Flash Playeru.

3.7 Zhodnocení a zdůvodnění užitych technologií

Aplikace byla naprojektována podle požadavků zadavatele, demo verze spuštěná nad testovacími daty je plně funkční. Jsou naprogramovány všechny důležité funkce zajišťující vnitřní běh programu, je vyřešeno i spouštění databázového serveru a připojení aplikace k němu. Grafická a uživatelská přívětivost bude ještě konzultována s grafikem, dá se předpokládat podstatná změna, prozatím byly použity pouze standartní grafické prvky dostupné ve Flashi.

Výhodou oddělení aplikace od zobrazovaných dat je možnost využití programu při běhu nad jinou databází, tedy stejný programový kód použitý v jiném typu encyklopedie. Zároveň se zvýší rychlost zpracování díky optimalizaci metod přístupu k datům v databázovém systému, ten nabízí i kryptování.

²²Portable Apache, MySQL, PHP Application plus Jaxer

Některé neprioritní požadavky zadávající firmy - videoukázky, časové osy a mapy zatím nebyly řešeny, jelikož ještě nejsou k dispozici všechna data, která se budou v encyklopedii zobrazovat, případně není jasné, zda bude tato funkčnost požadována. Předpokládá se implementace fulltextového vyhledávání, které bude řešeno v budoucím vývoji aplikace.

Aktualizace databáze offline verze je netriviální záležitost, taktéž nebyla řešena. Muselo by se zajistit verzování databáze. Aplikace by se pak přes internet připojovala na server a kontrolovala by nové diferenční balíčky. Bud' by se implementoval přenos a ukládání na počítači zároveň se sloučením s původní databází na DVD, nebo by se nová data dynamicky stahovala z internetové databáze, zde by se musel vyřešit problém se skriptováním mezi doménami (viz 3.6.2). Tyto návrhy nejsou pro zadávající firmu akceptovatelné z důvodu nutné implementace přihlašovacích údajů a licencí pro každého uživatele. Snahou bylo zajistit běh aplikace z DVD "tak jak je" bez potřeby ukládání dalších informací na uživatelské počítači.

Zhodnocení výběru platforem

Zadávající firma tlačila především na cenu výsledného řešení s přihlédnutím k případným licencím, takže "cena je až na prvním místě". Snahou tedy bylo využít co nejvíce aplikací s volnou licencí, nejlépe *GPL*.

Jediná částka byla placena za licenci na balíček Adobe CS3, přesto i tato položka mohla být nulová, kdybych použil alternativní nástroje na vytváření flashového obsahu. Offlinová aplikace by se však musela obejít bez samostatně spouštěného programu, data by byla zobrazována webovým prohlížečem, takže pro zajištění běhu by stačilo startovat lokálně databázový server. Tato položka však v celkovém plánovaném rozpočtu hraje zanedbatelnou částku.

Databázový systém MySQL byl vybrán z alternativních nástrojů pro tyto výhody:

- Optimalizace na výkon, tedy rychlost zpracování dat
- Jednoduché nastavení, zvláště pro spouštění offline verze na lokálním počítači
- Podpora mezi různými platformami
- Optimalizované metody pro přístupy k odlišným typům dat
- Díky velké uživatelské základně množství informací i podpůrných aplikací na internetu

Platforma Adobe Flash byla preferována nad výběrem Microsoft Silverlight kvůli těmto klíčovým vlastnostem:

- Ochrana obsahu, programové zdrojové kódy nejsou přímo čitelné
- Větší rozšířenost pluginu Flash Player v prohlížečích

Budoucí vývoj

Jako možné vylepšení programu bych si představoval tvorbu verze offline aplikace pro platformu Mac OS X. Implementace běhu databázového systému nad touto platformou není problémem, stejně tak využití funkce *autostart* pro pouštění aplikací přímo z DVD po jeho vložení do počítače. Varianta běhu na platformě Linux je také možná, pomocí externí placené aplikace by se vytvořil spustitelný program. Tyto alternativy se prozatím neuvažují z důvodu malé penetrace těchto systémů v českém prostředí.

Na aplikaci budu se zadávající firmou spolupracovat až do vydání výsledné verze multimediální encyklopedie.

Závěr

Výsledkem mé práce je využití získaných znalostí k vytvoření programu multimediální encyklopedie v prezentační platformě Adobe Flash s daty uloženými v databázovém systému MySQL. Aplikace vznikla ve dvou verzích, online na internetu a offline pro platformu Windows. Splnila očekávání a požadavky zadavatele.

Všeobecný přínos vidím ve vytvoření aplikace, která není běžně dostupná na softwarovém trhu. Osobním přínosem je prohloubení znalostí a zkušeností týkající se komplexní tvorby programů a návrhu databází. Vybral jsem technologii Adobe Flash především kvůli vysoké penetraci mezi uživatelskou základnou v rámci pluginu do webových prohlížečů a databázový systém MySQL pro optimalizaci metod přístupu k odlišným typům dat, kryptování, jednoduchost nastavení a podporu mezi platformami.

V oblasti programu je velmi zajímavé propojení Flashe s databázovým serverem a zajištění běhu offline verze z DVD podpůrnými nástroji. Aplikaci lze snadno spustit nad jinými daty, tedy prezentovat jiný obsah. Při návrhu i tvorbě jsem se snažil využít open-source nebo volně šiřitelné aplikace, protože zadavatel požadoval především nízkou cenu řešení. Výsledný program nelze brát jako finální produkt, ale jako první krok ve vývoji robustního systému pro multimediální prezentaci encyklopedických dat. Jeho další vývoj bude pokračovat ve spolupráci s firmou FoxCom s.r.o.

Literatura

- [1] ActionScript. Navštíveno 03. 2009.
Dostupné z: <http://en.wikipedia.org/wiki/ActionScript>
- [2] Adobe Flash. Navštíveno 02. 2009.
Dostupné z: http://en.wikipedia.org/wiki/Adobe_Flash
- [3] Adobe Flash Player Version Penetration. Navštíveno 03. 2009.
Dostupné z: http://www.adobe.com/products/player_census/flashplayer/version_penetration.html
- [4] Adobe Flex 3. Navštíveno 02. 2009.
Dostupné z: <http://www.adobe.com/products/flex/>
- [5] AsSQL – Actionscript 3 MySql Driver. Navštíveno 04. 2009.
Dostupné z: <http://code.google.com/p/assql/>
- [6] Comparison of relational database management systems. Navštíveno 05. 2009.
Dostupné z: http://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems
- [7] ECMAScript. Navštíveno 03. 2009.
Dostupné z: <http://en.wikipedia.org/wiki/ECMAScript>
- [8] Encyclopaedia. Navštíveno 02. 2009.
Dostupné z: <http://www.britannica.com/EBchecked/topic/186603/encyclopaedia>

- [9] Encyklopedie. Navštíveno 02. 2009.
Dostupné z: <http://cs.wikipedia.org/wiki/Encyklopedie>
- [10] Flash Lite features tour. Navštíveno 03. 2009.
Dostupné z: <http://www.adobe.com/products/flashlite/>
- [11] Google Web Toolkit - Product Overview. Navštíveno 03. 2009.
Dostupné z: <http://code.google.com/intl/cs/webtoolkit/overview.html>
- [12] JavaFX. Navštíveno 02. 2009.
Dostupné z: <http://www.sun.com/software/javafx/>
- [13] Metadata. Navštíveno 02. 2009.
Dostupné z: <http://cs.wikipedia.org/wiki/Metadata>
- [14] Microsoft Silverlight. Navštíveno 03. 2009.
Dostupné z: <http://silverlight.net/>
- [15] Multimedia. Navštíveno 02. 2009.
Dostupné z: <http://en.wikipedia.org/wiki/Multimedia>
- [16] Multimediální databáze. Navštíveno 02. 2009.
Dostupné z: http://cs.wikipedia.org/wiki/Multimedialni_database
- [17] MySQL. Navštíveno 02. 2009.
Dostupné z: <http://en.wikipedia.org/wiki/MySQL>
- [18] Obfuscated code. Navštíveno 02. 2009.
Dostupné z: http://en.wikipedia.org/wiki/Obfuscated_code
- [19] pampa-j – Portable Apache, MySQL, PHP Application plus Jaxer. Navštíveno 05. 2009.
Dostupné z: <http://code.google.com/p/pampa-j/>
- [20] Rich Internet application. Navštíveno 03. 2009.
Dostupné z: http://en.wikipedia.org/wiki/Rich_Internet_application

- [21] Rich Internet applications. Navštíveno 03. 2009.
Dostupné z: http://www.adobe.com/resources/business/rich_internet_apps
- [22] Silverlight vs. Flash - An Analysis Report. Navštíveno 03. 2009.
Dostupné z: <http://silverlight.net/forums/t/3015.aspx>
- [23] Software Adobe Flash CS4 Professional: podrobnosti o upgradu. Navštíveno 03. 2009.
Dostupné z: <http://www.adobe.com/cz/products/flash/upgrade/?view=compare>
- [24] Wikipedie. Navštíveno 02. 2009.
Dostupné z: <http://cs.wikipedia.org/wiki/Wikipedie>
- [25] Grossman, G.; Huang, E.: ActionScript 3.0 overview. Navštíveno 03. 2009.
Dostupné z: http://www.adobe.com/devnet/actionscript/articles/actionscript3_overview.html
- [26] de Jong, T.: Fixed AS3 Flash TextArea - CSS incompatibility. 2008, navštíveno 04. 2009.
Dostupné z: <http://www.sharedknowhow.com/2008/07/fixed-flash-textarea-css-incompatibility/>
- [27] Kadavý, J.: Akademická roadshow Microsoft - Silverlight 2. 2009, přednáška na Technické univerzitě v Liberci.
- [28] Špánek, R.: Úvod do databázových systémů. Navštíveno 05. 2009.
Dostupné z: <http://sirius.ksi.tul.cz/~roman.spanek/files/dbs/2007/pr1.pdf>
- [29] Švejda, G.: Multimédia ve vzdělávání. 1999, navštíveno 02. 2009.
Dostupné z: <http://www.pf.jcu.cz/stru/katedry/pgps/svejda-ztv.htm>

Přílohy

CD disk s elektronickou podobou diplomové práce, kódem aplikace a demo verzí programu.