

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SEMANTICKÝ WEB TUTORIÁL A DEMONSTRAČNÍ PŘÍKLADY

BAKALÁŘSKÁ PRÁCE

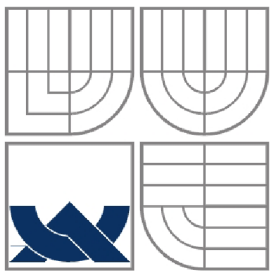
BACHELOR'S THESIS

AUTOR PRÁCE

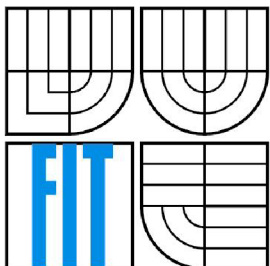
AUTHOR

PETR HANYÁŠ

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SEMANTICKÝ WEB TUTORIÁL A DEMONSTRAČNÍ PŘÍKLADY

SEMANTIC WEB – TUTORIAL AND EXAMPLES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PETR HANYÁŠ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PAVEL OČENÁŠEK

BRNO 2007

Zadání bakalářské práce

Řešitel: **Hanyáš Petr**

Obor: Informační technologie

Téma: **Semantický Web - tutorial a demonstrační příklady**

Kategorie: Web

Pokyny:

1. Seznamte se s možnostmi vývoje SW aplikací v prostředí současného WWW. Seznamte se s principy ontologie a použití a tvorby semantického webu.
2. Navrhněte tutorial zabývající se ontologií a dále návrhem a realizaci aplikací v prostředí sémantického webu (a souvisejících technologií).
3. Navržený tutorial implementujte dle pokynů vedoucího práce.
4. Vytvořte sadu názorně komentovaných demonstračních příkladů pro podporu výuky sémantického webu.
5. Diskutujte další možnosti pokračování projektu.

Literatura:

- Dle doporučení vedoucího práce.

Při obhajobě semestrální části projektu je požadováno:

- Body 1 - 3.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Očenášek Pavel, Ing., UIFS FIT VUT**

Datum zadání: 1. listopadu 2006

Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2



doc. Ing. Jaroslav Zendulka, CSc.
vedoucí ústavu

LICENČNÍ SMLOUVA POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Petr Hanyáš**

Id studenta: 84386

Bytem: Horní Bečva 176, 756 57 Horní Bečva

Narozen: 27. 06. 1985, Ostrava

(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií

se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

Článek 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: Semantický Web - tutorial a demonstrační příklady

Vedoucí/školitel VŠKP: Očenášek Pavel, Ing.

Ústav: Ústav informačních systémů

Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě počet exemplářů: 1

elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel

.....

Autor

Abstrakt

Tato práce se zabývá návrhem a realizací interaktivní prezentace obsahující tutoriál na téma „Sémantický web“ a ním související komentované demonstrační příklady. Práce také obsahuje správu uživatelů a komentářů, testovací systém s náhodným generováním otázek, fulltextový vyhledávač a popisy editorů ontologií. Cílem celého projektu je vytvořit výukové materiály s možností ověření nabytých znalostí. Motivací k této práci byla nedostupnost kompletních, česky psaných materiálů na toto téma. Mým cílem bylo vytvoření právě těchto materiálů, tak aby byly srozumitelné co nejširšímu spektru čtenářů.

Klíčová slova

CSS, MySQL, ontologie, PHP, příklad, sémantický web, test, tutorial, XHTML

Abstract

This thesis deals with design and realization of interactive presentation consists of tutorial which topic is „Semantic Web“ and related demonstration examples with comments. Thesis is also consists of users' and commentaries' administration, testing system with random generation of questions, fulltext search engine and description of ontology editors. The aim of whole thesis is to create educational materials with a possibility of checking reached level of knowledge. Motivation for making this thesis was lack of complete materials in Czech language dealing with this problem. My aim was to create these kinds of materials so it can be intelligible to wide range of readers.

Keywords

CSS, MySQL, ontology, PHP, example, semantic web, test, tutorial, XHTML

Citace

Petr Hanyáš: Sémantický web – tutorial a demonstrační příklady, bakalářská práce, Brno, FIT VUT v Brně, 2007

Sémantický Web – tutorial a demonstrační příklady

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Pavla Očenáška
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Petr Hanyáš
12. 5. 2007

© Petr Hanyáš, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	3
2 Základy sítě Internet.....	4
2.1 Základy sítě Internet.....	4
2.1.1 Poskytované služby.....	4
2.1.2 Komunikace mezi počítači.....	4
2.1.3 Server.....	5
2.1.4 Klient.....	5
2.2 Sémantický web.....	5
2.2.1 Základní součásti.....	6
2.3 Ontologie.....	6
2.3.1 Přední ontologické jazyky.....	7
2.3.2 Jazyk OWL.....	7
3 Reprezentace ontologií.....	9
3.1 RDF Model.....	9
3.1.1 Popis jazyka.....	10
3.1.2 Reprezentace pomocí grafů.....	10
4 Tvorba webových aplikací.....	11
4.1 PHP.....	11
4.1.1 Vlastnosti jazyka.....	11
4.2 SQL databáze.....	11
4.2.1 Historie SQL.....	12
4.2.2 MySQL.....	12
4.3 XHTML.....	13
4.3.1 Jazyk XML.....	13
4.4 Kaskádové styly.....	13
4.4.1 Výhody oproti HTML.....	14
4.4.2 Nevýhody.....	14
5 Současný stav.....	16
6 Návrh řešení.....	17
6.1 Požadavky na systém.....	17
6.2 Vrstvy aplikace.....	17
6.3 Struktura aplikace.....	18

7 Implementace projektu.....	19
7.1 Základní rozvržení.....	19
7.2 Vzor singleton.....	20
7.3 Správa uživatelů.....	20
7.3.1 Role uživatelů.....	20
7.4 Tutoriál a demonstrační příklady.....	21
7.4.1 Fulltextové hledání.....	21
7.4.2 Cache.....	22
7.5 Testy.....	22
7.6 Propojení s databází.....	22
8 Závěr.....	23
8.1 Další rozšíření.....	23
Literatura.....	24
Seznam příloh.....	25
Příloha 1. Demonstrační obrázky.....	26
Příloha 2. Popis tříd.....	28
Příloha 3. Manuál.....	29

1 Úvod

Tato práce si klade za cíl návrh a implementaci interaktivního tutoriálu na téma Sémantický web, prezentovaného přes síť Internet s možností ověření nabytých znalostí pomocí testů.

Motivací k vytvoření tohoto projektu byla nedostupnost kompletních, česky psaných materiálů na dané téma. A také vytvoření systému k ověřování nabytých znalostí s možností zpětného náhledu na výsledky a generováním náhodných otázek.

Inspirací byly již vytvořené tutoriály na jiná témata: **Seriál PHP**, dostupný na adrese: <http://www.linuxsoft.cz/php/>, formátování zdrojových kódů v tutoriálu **C++ Language Tutorial**, jež můžete nalezít na adrese: <http://www.cplusplus.com/> a také systém testování **Cisco Academy**, který naleznete na adrese: <http://cisco.netacad.net/>.

Realizace celé aplikace bude založena na reprezentaci pomocí jazyka XHTML 1.0 a generována pomocí serverového skriptovacího jazyka PHP 5. Všechny údaje, články a demonstrační příklady budou uloženy v databázi MySQL. Pro vytvoření vzhledu prezentace použijí kaskádové styly. Zvolené technologie byly vybrány zejména kvůli jejich dostupnosti a také ceně (pro daný účel jsou dostupné zdarma). Tyto jazyky patří v současné době k nejpoužívanějším a proto jsou vhodné pro reprezentaci takového projektu.

Práce je rozdělena do pěti tematických částí. V první části jsou popsány základy sítě Internet a sémantického webu. Je zde také vysvětlen termín ontologie a s ním související pojmy.

Druhá část obsahuje informace o reprezentaci znalostí ontologií, základy RDF schéma jeho reprezentaci pomocí jazyka XML.

V následující kapitole jsou popsány základní údaje o tvorbě dynamických webových aplikací. Zejména jsou zde informace o serverových jazycích PHP, SQL, značkovacím jazyce XHTML a tvorbě vzhledu pomocí kaskádových stylů.

Další kapitola shrnuje v jednoduchosti současný stav, jak na česky psaných internetových stránkách, tak na internetu všeobecně.

V předposlední kapitole jsou poznatky o návrhu praktické části prezentované práce a rozdělení aplikace na její součásti.

Poslední část se zabývá vlastní implementací tohoto projektu. Jsou zde popsány použité techniky programování a výběr zajímavých částí.

V závěru budou shrnuty získané vědomosti a konečné zhodnocení výsledné aplikace. Jsou zde popsány i důvody, proč se sémantický web rozvíjí pomalým tempem a vize, kterými se bude ubírat.

2 Základy sítě Internet

2.1 Základy sítě Internet

Internet je celosvětová síť, která se skládá z mnoha menších podsítí. Název vznikl spojením dvou anglických slov: **inter** a **network**. Počátkem 70. let vznikla počítačová síť americké armády nazvaná Arpanet, která se později rozdělila na vojenskou a výzkumnou část. Výzkumná část se na přelomu 80. a 90. let stala dostupnou pro širokou veřejnost a zároveň začala získávat dnešní podobu s možností zobrazování WWW¹ stránek, e-mailů, atd.

Na počátku 21. století publikoval Tim Berners-Lee² svoji myšlenku sémantického webu založenou na ontologiích a znalostech, které můžeme využívat. V dnešní době se pomalu začíná pojem „sémantický web“ dostávat do podvědomí širšího spektra tvůrců webových aplikací.

2.1.1 Poskytované služby

Soudobý Internet obsahuje mnoho služeb, jež může využít každý, kdo má přístup k této celosvětové síti. Jedná se zejména o tyto služby:

- SMTP (Simple Mail Transfer Protocol) - poskytuje odesílání elektronické pošty
- POP3, IMAP (Post Office Protocol, Internet Message Access Protocol) – umožňují přijímat elektronickou poštu
- DHCP (Dynamic Host Configuration Protocol) – umožňuje síťovou konfiguraci počítače bez nutnosti zásahu uživatele
- HTTP (HyperText Transfer Protocol) – slouží k přenosu dat, obrázků a textů v uceleném formátu, mezi počítači
- FTP (File Transfer Protocol) – je využíván pro přenos souborů
- DNS (Domain Name System) – překlad doménových jmen na IP adresy

2.1.2 Komunikace mezi počítači

Aby mohly jednotlivé počítače vzájemně komunikovat musí být každý z nich připojen do sítě (v ideálním případě do sítě Internet) jakýmkoliv druhem připojení (pomocí telefonní linky, ISDN, mobilního telefonu, WiFi, kabelové televize, aj.) a mít přidělenou takzvanou IP³ adresu. Ta je tvořena pomocí 4 desítkových čísel v rozsahu 0-255 (např.: 192.168.1.1).

1 WWW – World Wide Web - soustava propojených hypertextových dokumentů

2 Tim Berners-Lee je jedním ze zakladatelů World Wide Webu a vedoucím členem konsorcia W3C

3 IP – protokol pro komunikaci zařízení v Internetu

Ve většině případů musí jeden počítač tvořit Server a druhý se k němu připojuje jako Klient. Tato architektura se nazývá Klient/Server. V našem případě se jedná o model třívrstvé architektury, ve které se konkrétně jedná o vrstvy: prezenční, aplikační a datovou.

2.1.3 Server

Je zařízení na kterém fyzicky nepracuje žádný uživatel, ale pouze se k němu připojuje a využívá jeho služeb. Je schopen zpracovávat obrovské množství požadavků v krátkém čase, ale pouze z jedné oblasti (např.: generování dynamických stránek pomocí PHP). Příkladem může být emailový server, jež zpracovává odesílanou, přijatou a uchovávanou poštu, kterou poté zprostředkovává klientovi, který se k serveru připojuje.

2.1.4 Klient

Je aplikace na počítači se kterým fyzicky pracuje uživatel. Tento počítač nemusí mít velký výpočetní výkon, jelikož jeho jedinou činností je interpretovat výsledky přijaté serverem. Připojuje se k serveru a od něj teprve získává všechny potřebné informace. Například webový prohlížeč, který se dotáže serveru na zadanou webovou stránku. Server mu tuto stránku odešle a klient ji pouze zobrazí uživateli.

2.2 Sémantický web

Myšlenka sémanticky psaného webu byla poprvé oficiálně prezentována v květnu roku 2001. Tim Berners-Lee, jeden z tvůrců současného webu a ředitel konsorcia W3C⁴ spolu s dalšími spoluautory upozornili článkem v magazínu Scientific American na skutečnost, že současná síť WWW je v podstatě jen haldou webových stránek, která neustále roste a je v ní stále složitější nalézt relevantní informace. Východisko z tohoto chaosu spatřují v postupném přerodu stávajícího webu v tzv. sémantický web, jehož uživatelská představa je vyjádřena hned v úvodu jejich článku.

Představa autorů je taková, že jsou inteligentní (v budoucnu zejména mobilní) zařízení schopná navzájem automaticky komunikovat, řešit samostatně, bez pomoci člověka složité úkoly, jež se opírají o informace, znalosti a jejich důvěryhodné sdílení.

Důležitým předpokladem sémantického webu je zejména standardizovaný popis webových zdrojů. Zdrojem se v této souvislosti rozumí vše, čehož můžeme dosáhnout prostřednictvím sítě WWW, tedy textové dokumenty, obrázky, videosekvence, zvukové soubory a další. Každý zdroj má být vybaven stejnými vlastnostmi (autor, typ zdroje, klíčová slova, atd.), které umožní uživatelům internetu pracovat se sítí WWW jako s relační databází a dotazovat se na její obsah prostřednictvím

4 W3C - Organizace jež definuje standardy na Internetu

jazyků podobných SQL. Z toho vyplývají obrovská pozitiva hlavně při vysoké přesnosti a relevance odpovědí na vyhledávací dotazy, což znamená, že by byl uživateli při vyhledávání určité informace vrácen seznam všech zdrojů, které se této informace týkají, a žádný zdroj navíc.

2.2.1 Základní součásti

- **XML** - poskytuje základní syntaxi pro strukturované dokumenty, ale nevytváří žádné sémantické znaky definující tyto dokumenty.
- **XML Schema** - je jazykem pro omezení struktury XML dokumentů a také rozšiřuje XML o datové typy.
- **RDF** - je datový model pro objekty (zdroje) a relace mezi nimi. Poskytuje jednoduchou sémantiku pro tento datový model a tyto modely také mohou být reprezentovány pomocí XML syntaxe.
- **RDF Schema** - je slovník pro popis vlastností a tříd RDF zdrojů se sémantikou pro generalizaci hierarchie těchto vlastností a tříd.
- **OWL** - přidává větší slovník pro popisování vlastností a tříd. Mimo jiné relace mezi třídami, kardinalitu (např.: právě jeden), rovnost, více typů vlastností, charakteristiku (např.: symetrie) a vyjmenované třídy (viz kapitola Jazyk OWL).

2.3 Ontologie

Při studiu ontologií jsem čerpal z [4]. Ve filosofii je ontologie chápána jako nauka (nebo lépe soubor nauk) o „bytí“, popřípadě jako univerzální soustava znalostí popisující objekty, jevy a zákonitosti světa. Naopak v informatice je ontologie specifikována jako "explicitní specifikace konceptualizace"⁵. Konceptualizace (systém pojmů modelující určitou část světa) musí být specifikována explicitně, tj. nikoliv jen "skryta" v hlavě svého autora. Účelem ontologií je podpora porozumění mezi lidmi, podpora komunikace mezi počítačovými systémy a podpora návrhu znalostně orientovaných systémů.

Základním krokem k vytvoření sémantického webu je konceptualizace dat dostupných na internetu. Jedním z klíčových nástrojů konceptualizace jsou právě ontologie. Ontologie lze charakterizovat jako formalizované reprezentace znalostí určené k jejich sdílení a znovupoužití. Jsou často doménového (oborového) zaměření a bývají konstruovány jako pojmové (konceptuální) hierarchie nebo sítě.

5 Gruberova definice ontologie

2.3.1 Přední ontologické jazyky

Během vývoje ontologie, bylo vytvořeno mnoho jazyků, zde si uvedeme nejdůležitější z nich:

- **Cyc** - je jednou z prvních ontologií, která se snažila obsáhnout znalosti o světě ve velkém měřítku. Usiluje o shromáždění všeobecných znalostí, které by fungovaly komplementárně ke znalostem expertním a zabraňovaly vzniku absurdních stavů. Pro formální reprezentaci používá jazyk *CycL* vycházející z jazyka *LISP*.
- **Ontolingua** – jazyk je koncipován jako nadstavba jazyka *KIF*⁶, což je varianta predikátového kalkulu využívající syntaxi jazyka *LISP*⁷. Základními konstrukty tohoto jazyka jsou definice tříd, relací a funkcí, přičemž vymezení podmínek pro příslušnost instancí jsou vyjádřeny právě v jazyce *KIF*.
- **OCML** - základem interpretu jsou algoritmy pro Prologovské dokazování a dědění v hierarchii tříd; třídy a jejich atributy jsou ovšem důsledně chápány jako unární resp. binární relace, takže primární (vnitřní) reprezentací jsou Hornovy klauzule⁸.
- **SHOE** - je prvním jazykem vzniklým specificky pro účely přidání sémantiky k webovým stránkám. Umožňuje začleňovat do zdrojového kódu webových stránek metadata o objektech, kterých se tyto stránky týkají, a také samotné ontologie, jež definují sémantiku těchto metadat.
- **DAML+OIL** - základním kamenem této ontologie jsou pojmenované (jmenné) a anonymní (logický výraz) třídy. Pro vytváření logických výrazů definujících třídy se používají konstruktory, které lze libovolně skládat. Obsahem jsou axiomy vytvořené nad výrazy reprezentující třídu.

2.3.2 Jazyk OWL

Jazyk OWL byl vytvořen pro získávání informací pomocí aplikací (narozdíl od dnešního způsobu, kdy jsou informace získávány člověkem). Tento jazyk je využíván pro explicitní definici termínů ve slovnících a vztahu mezi těmito termíny. Tato definice podmínek a vztahů se obecně nazývá ontologie. Na rozdíl od jazyků RDF, RDFS, XML má OWL větší možnosti, jak vyjádřit sémantiku a význam obsahu. OWL vychází z jazyka DAML+OIL.

Sémantický web staví na schopnostech jazyka XML, definovat vlastní značkovací schéma a flexibilním přístupem k datům pomocí RDF. První úroveň nad RDF vyžadovaná pro funkci správně psaného sémantického webu je ontologie, která dokáže formálně popsat význam názvosloví použitého ve webových dokumentech. Aby mohly stroje generovat kvalitní výsledky, musí jazyk

6 KIF - Knowledge Interchange Format je jazyk, který slouží k výměně informací mezi různými programy

7 LISP - LISt Procesing je funkcionální programovací jazyk používaný v oboru umělé inteligence

8 Hornova klauzule je disjunkce literálů, jež obsahuje pouze jeden pozitivní literál

přesahovat základní sémantiku RDF Schema. OWL používá *Případové* a *Požadavkové* dokumenty poskytující více detailů ontologií, motivuje potřebu pro ontologickou síť jazyka při podmínkách šesti použitých případů a formuluje cíle designu, požadavky a cíle pro OWL. OWL byl navržen přímo jako webový ontologický jazyk a je dle doporučení konsorcia W3C je součástí sémantického webu.

2.3.2.1 Verze jazyka OWL

OWL poskytuje 3 více-méně odlišné jazyky určené pro použití ve specifických komunitách implementátorů a uživatelů:

- **OWL Lite** je primárně určen pro ty uživatele, kteří požadují pouze hierarchii tříd a jednoduché omezení. Například podporuje omezení kardinality, která umožňuje vytvářet mohutnosti o velikosti 0 nebo 1. Verze Lite je jednodušší, a proto je také jednodušší vytvářet nástroje pro podporu toho to jazyka. Například poskytuje rychlou cestu k přestěhování vašeho slovníku Tezaurus⁹ nebo jiné taxonomie. OWL Lite má nižší formální složitost než verze OWL DL, více dále.
- Varianta **OWL DL** je určená těm uživatelům, kteří vyžadují maximální výraznost při co nejmenší výpočetní náročnosti (ovšem jsou všechny závěry 100% vypočítatelné) a bezspornost (všechny výpočty skončí vždy v reálném čase). OWL DL zahrnuje všechny konstrukce jazyka OWL, ale ty mohou být použity pouze s danými omezeními (např.: jedna třída může být podtřídou mnoha dalších tříd, ale třída nemůže být příkladem další třídy). Zkratka DL je zde užitá ve významu *description logics* (logika popisu).
- Verze **OWL Full** je určena uživatelům, kteří požadují maximální výraznost a syntaktickou svobodu (jako u RDF), ovšem nezaručuje žádné výpočetní záruky. Například v OWL Full může být třída současně kolekcí individualit a zároveň individualita v jejich vlastní správě. Dovoluje ontologiím, aby plně vystihla význam předdefinovaného (RDF nebo OWL) slovníku. Bohužel je nepravděpodobné, že vznikne nějaký software, který bude schopen plně podporovat každý rys verze Full.

Každý z těchto 3 jazyků je rozšířenou verzí jeho předchůdce, ve kterém může být plně vyjádřen a zároveň platí následující vztahy:

- Každá platná ontologie napsaná v OWL Lite je platná i v ontologii OWL DL.
- Každá platná ontologie napsaná v OWL DL je platná i v ontologii OWL Full.
- Každý platný závěr ontologie OWL Lite je zároveň platným závěrem ontologie v OWL DL.
- Každý platný závěr ontologie OWL DL je zároveň platným závěrem ontologie v OWL Full.

9 Slovník, který uživateli nabízí seznam synonym a antonym

3 Re prezentace ontologií

3.1 RDF Model

Při studiu této kapitoly jsem čerpal z [9]. Resource Description Framework je doporučením W3C pro reprezentaci struktury webových metadat. Datový model RDF je založen na lingvisticky inspirované konstrukci složené ze tří prvků: **subjektu**, **predikátu** a **objektu**, která se dohromady označuje jako tvrzení ("**statement**"). Univerzálním prvkem RDF je zdroj ("**resource**"), identifikovaný svým URI. Zdrojem může být nejen webová stránka, ale např. celé webové sídlo nebo i dokument umístěný mimo web. Zdroj může vystupovat jak v roli subjektu, tak v roli objektu. Predikáty odpovídají sledovaným vlastnostem subjektů, a objekty hodnotám těchto vlastností. Objektem může být kromě zdroje také literál, tj. primitivní datová hodnota.

Datový model RDF sám o sobě nepředjímá konkrétní reprezentaci – ta může být např. grafová (subjekty a objekty jako uzly, predikáty jako orientované hrany) nebo vyjadřovat tvrzení prostě jako posloupnost tří výrazů na téže řádce. Základní syntaxe doporučovaná W3C je ovšem založena na XML. Hovoří se o tzv. serializaci: jednotlivé prvky tvrzení RDF jsou specifickým způsobem řazeny za sebe do elementů a atributů XML.

Obálkový element RDF obsahuje atributy definující jmenné prostory: *rdf* pro konstrukty samotného RDF, a *dc* pro metadatový standard Dublin Core, z něhož je převzata sledovaná vlastnost „být tvůrcem“. Vlastní tvrzení je pak obsaženo v elementu Description, jehož atribut *about* odkazuje na subjekt, název subelementu odpovídá predikátu, a obsah subelementu objektu.

Specifikace RDF obsahuje řadu možností, jak vyjádřit složitá tvrzení úsporněji. Je zjevné, že pokud chceme témuž subjektu přiřadit různé vlastnosti, můžeme to učinit v rámci jediného elementu *description* s více subelementy. Pokud chceme naopak přiřadit tutéž vlastnost více zdrojům, lze je sdružit do tzv. kolekce. Vedle základní notace také existuje notace zkratková, která hodnoty metadat umísťuje do atributů namísto do subelementů.

Velmi silným prostředkem je v RDF tzv. reifikace. Její podstatou je možnost chápat celé tvrzení jako zdroj, a vypovídat o něm prostřednictvím dalších tvrzení. Motivací vzniku této konstrukce byla zejména potřeba usuzovat o věrohodnosti jednotlivých tvrzení na základě důvěry v osobu, která je „vyslovila“ takto se příslušná „meta-tvrzení“ dostávají na stejnou úroveň jako tvrzení základní, a není potřeba pro ně vytvářet zvláštní syntaktický aparát. Druhou stranou mince je ovšem podstatné zkomplikování formálních vlastností modelu.

3.1.1 Popis jazyka

Základními objekty jsou **zdroje**, které jsou identifikovány pomocí URI. URI může být přiřazeno jakémukoliv objektu, pojmu nebo např. osobě. V praxi může obsahovat URL a nebo také jednoznačný identifikátor (například rodné číslo, unikátní kód, ...). Z toho vyplývá, že jediný požadavek na URI je, aby byl **jednoznačný**.

Výroky v RDF přiřazuje zdroji **vlastnost: hodnota**. Tedy výroky jsou složeny z trojic: zdroj (předmět), vlastnost (predikát) a hodnoty vlastnosti (objekt):

- **Zdroj** - je předmětem, na který se můžeme odkazovat pomocí URI. V praxi je zdroj anonymní a odkazuje se na něj pomocí "_:n" (n je přirozené číslo).
- **Vlastnost** - (predikát) vytváří binární relaci mezi zdroji a atomickými hodnotami poskytnutými definicemi primitivních datových typů v XML. Vlastnost může být dána také pomocí URI.
- **Hodnota vlastnosti** - (objekt) jak již z názvu vyplývá specifikuje vlastnost nějakého predikátu. Objektem může být literál nebo další zdroj.

3.1.2 Reprezentace pomocí grafu

RDF Graf lze popsat ohodnoceně orientovaným multigrafem. Dva uzly mohou být spojeny jednou i více hranami. Ovály se využívají pro zdroje, orientovaně označené hrany jsou vlastnosti a obdélníky obsahují hodnoty vlastností. Zdroje mohou mít i více vlastností a literály se ve výrocih vyskytují jen jako objekty. Hrany jsou vyznačeny pomocí URI, ale pro jednoduchost je lze označit pomocí x:, označující libovolné URI.

4 Tvorba webových aplikací

V dnešní době jsou nepoužívanějšími jazyky pro tvorbu dynamických stránek ASP (Active Server Page) a PHP (Page Hypertext Preprocessor). Jazyk PHP je s využitím databáze vhodný pro začlenění přímo do struktury jazyka (X)HTML. Vzhled již také není tvořen pomocí tabulek, nýbrž je využito kaskádových stylů.

4.1 PHP

V této kapitole jsem čerpal z [6]. Tento jazyk vytvořil Rasmus Lerdorf v roce 1994, přepsáním z PERLu do jazyka C. V první polovině roku 1995 se systém PHP spojil s programem Form Interpreter a vzniklo z něj PHP/FI 2.0. Roku 1997 přepsali Zeev Suraski a Andi Gutmans parser a vytvořili tak základ PHP3. Současně byl název změněn na dnešní rekurzivní podobu: PHP Hypertext Procesor.

Verze PHP3 vyšla v roce 1998 a byla spustitelná i pod operačním systémem Windows. V roce 2000 vyšla verze PHP4 a o čtyři roky později vyšla verze PHP5, která obsahovala přepracovaný objektový model, zpracování výjimek, zlepšení podpory XML a SQL Lite.

4.1.1 Vlastnosti jazyka

Jazyk PHP je dynamicky typový, z toho vyplývá že datový typ dané proměnné se určí v okamžiku přiřazení hodnoty. Proto PHP obsahuje dva typy porovnání „==“, je stejný jako v jazyce C a „===“, který platí pouze jsou-li oba výrazy shodné a mají stejný typ.

Pole jsou heterogenní, proto mohou obsahovat jakékoliv údaje (což se týká i jejich indexů). A řetězce lze uzavírat do uvozovek (pak je obsah parsován) nebo do apostrofů (obsah není parsován).

Od verze 4.2 lze získávat hodnoty ze superglobálních polí, čím máme zaručeno, že data byly odeslány požadovanou metodou (POST, GET, SESSION, COOKIE).

Jazyk také obsahuje rozsáhlou podporu pro komunikaci s databázemi (MySQL, PostgreSQL, SQLite, aj.), LDAP servery, zpracování grafiky a textů.

4.2 SQL databáze

SQL (Structured Query Language) je standardizovaný dotazovací jazyk, který se používá pro práci s daty v relačních databázích. Standardy podporuje prakticky každá relační databáze, ale obvykle nejsou implementovány vždy všechny požadavky normy. A naopak, každá z nich obsahuje prvky a

konstrukce, které nejsou ve standardech obsaženy. Přenositelnost SQL dotazů mezi jednotlivými databázemi je proto omezená.

4.2.1 Historie SQL

V této části jsem čerpal z [7]. V 70. letech 20. století probíhal ve firmě IBM výzkum relačních databází. Z toho to výzkumu vyplynulo, že je nutné vytvořit sadu příkazů pro ovládání těchto databází. Vznikl tak jazyk **SEQUEL** (Structured English Query Language). Cílem bylo vytvořit jazyk, ve kterém by se příkazy tvořily syntakticky co nejbližší přirozenému jazyku (angličtině).

K vývoji jazyka se přidaly další firmy. V r. 1979 uvedla na trh firma Relational Software, Inc. (dnes Oracle Corporation) svůj relační databázový systém Oracle. IBM uvedla v roce 1981 nový systém SQL/DS a v roce 1983 systém DB2. Dalšími systémy byly např. Progres, Infomix a SyBase. Ve všech těchto systémech se používala varianta jazyka SEQUEL, který byl přejmenován na **SQL**.

4.2.2 MySQL

MySQL je databázový systém, vytvořený švédskou firmou MySQL AB. Jeho hlavními autory jsou Michael Widenius a David Axmark. Je považován za úspěšného průkopníka dvojího licencování – je k dispozici jak pod bezplatnou licenci GPL, tak pod komerční placenou licenci.

MySQL je multiplatformní databáze. Komunikace s ní probíhá – jak už název napovídá – pomocí jazyka SQL. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními.

Pro svou snadnou implementovatelnost (lze jej instalovat na Linux, MS Windows, ale i další operační systémy), výkon a především díky tomu, že se jedná o volně šiřitelný software, má vysoký podíl na v současné době používaných databázích. Velmi oblíbená a často nasazovaná je kombinace MySQL, PHP a Apache jako základní software webového serveru. Citace z [7].

4.2.2.1 Úložiště dat

MySQL nabízí více typů tabulek, které se liší způsobem ukládání dat a možnostmi jejich správy:

- MyISAM - nejpoužívanější, bez podpory transakcí
- InnoDB - podpora transakcí
- BerkleyDB - (BDB)
- MEMORY - práci s daty v paměti
- NDB Cluster - úložiště pro clusterované databáze (od verze 5.0)
- ARCHIVE - komprimované tabulky, bez podpory indexů
- CSV - ukládání dat v prostých textových souborech

4.2.2.2 Podporované vlastnosti

Verze MySQL 5 se drží standardu SQL92 a navíc obsahuje některé svoje vlastnosti jako např. automatickou inkrementaci primárního klíče. Dále pak mají tyto vlastnosti:

- Cizí klíče (od verze 3.23 podporovány v tabulkách typu InnoDB)
- transakce (od verze 3.23 podporovány v tabulkách typu InnoDB)
- podpora různých znakových sad a časových pásem v datech (od verze 4.1)
- poddotazy (od verze 4.1)
- uložené procedury (od verze 5.0)
- trigger (od verze 5.0)
- pohledy (od verze 5.0)

4.3 XHTML

XHTML (eXtensible Hypertext Markup Language) je značkovací jazyk, který vznikl sloučením XML (eXtensible Markup Language) a HTML (Hypertext Markup Language). Jedná se o následníka jazyka HTML, který byl označen jako zastaralý. Organizace W3C (World Wide Web Consortium) jej doporučuje pro tvorbu hypertextových dokumentů v prostředí WWW. V tomto projektu je použita verze XHTML 1.0, jež je první specifikací tohoto jazyka, která zachovává zpětnou kompatibilitu a vyhovuje podmínkám pro tvorbu dokumentů v XML.

Jedinými rozdíly jsou povinné párové prvky (musí být uzavřeny), uzavření všech hodnot do uvozovek a veškeré atributy musí být psány malými písmeny.

4.3.1 Jazyk XML

XML je obecný značkovací jazyk, který umožňuje snadné vytváření dalších značkovacích jazyků. Je určen především pro výměnu dat mezi aplikacemi. Nezabývá se vzhledem dokumentu, ale popisuje jeho strukturu a hlavně věcný obsah jednotlivých částí.

4.4 Kaskádové styly

Zkratka znamená Cascading Style Sheet (anglicky), neboli tabulky kaskádových stylů. Tento jazyk pouze ovlivňuje výsledný vzhled generované stránky. Je interpretován v prohlížeči u klienta.

Původně ovlivňoval styl dokumentu jazyk HTML, ale v důsledku nedostatečné podpory standardů v různých prohlížečích a konkurenčního boje mezi nimi se od tohoto záměru upustilo. Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Starší verze HTML obsahují celou řadu elementů, které nepopisují obsah a strukturu dokumentu, ale i

způsob jeho zobrazení. Z hlediska zpracování dokumentů a vyhledávání informací není takový vývoj žádoucí.

Pro různá zařízení lze vytvářet různé verze stylů. Autor má tedy možnost určit na jakém výstupním médiu bude daná prezentace vypadat (např. PDA, tiskárna, standardní prohlížeč). Specifikace dokonce umožňuje prezentovat obsah zřetěženým. Je možné vytvořit styl pro braillovo písmo nebo hlasový sintetizátor.

Vlastní styly lze vytvořit také pro každý webový prohlížeč zvlášť, čímž lze odstranit některé nedostatky a odchylky reprezentace v jednotlivých prohlížečích od standardů. Stačí pouze vytvořit více souborů se styly a podle identifikátoru prohlížeče přiřadit stránce daný styl.

4.4.1 Výhody oproti HTML

CSS nabízí rozsáhlejší formátovací možnosti než samotné HTML. Např. pro formátování bloku textu - tj. určení vzdálenosti od jejich elementu či okraje stránky nenabízí HTML nic. CSS má vlastnosti *padding* a *margin*. V HTML by bylo potřeba vytvořit složitou konstrukci vnořených tabulek.

Na všech stránkách webové prezentace by měly být všechny nadpisy stejné úrovně, seznamy, zdůrazněné části textu apod. stejného stylu. S použitím formátovacích možností HTML je to obtížné – u každého objektu v každém dokumentu se vzhled objektu stále znovu nastavuje. S použitím CSS je to velmi jednoduché. Vytvoří se soubor stylu, který se připojuje k HTML dokumentu. Ve všech dokumentech jsou pak objekty stejného vzhledu.

Provést změnu stylu webu, který pro formátování vzhledu využívá jen možnosti HTML, znamená najít a nahradit všechny značky a změnit atributy mnoha dalších značek. V případě používání CSS znamená změna stylu webu přepsání jediného souboru – souboru stylů.

Výhodou CSS oproti starému formátování v HTML je, že kód a obsah webu je uložen v souboru *.html* a veškerý design a formátování se načítá z jednoho souboru *.css*, který je většinou společný pro celý web. To znamená, že pokud máte v plánu změnu designu webu, stačí změnit pouze jeden soubor *.css* a změna se aplikuje na celý web. Také se soubor CSS uloží do mezipaměti prohlížeče a pokud není změněn, tak se načítá pouze jednou a zobrazení stránek se velmi urychlí.

4.4.2 Nevýhody

Hlavní nevýhodou CSS je zatím stále špatná podpora v majoritních prohlížečích. Různé prohlížeče interpretují stejný CSS kód jinak a je někdy velmi obtížné jej napsat tak, aby se na všech (resp. na několika vybraných) prohlížečích výsledek zobrazil stejně. Situace se ale od roku 2006 značně zlepšuje, v souvislosti s tím se s napětím očekával příchod Internet Exploreru 7, který by měl

postupně vytlačit svého předchůdce IE 6, který byl častým zdrojem problémů. Nicméně ani IE 7 se striktně nedrží definice CSS 2.1. Citace z [1].

5 Současný stav

Situace na webu je značně nepřehledná a při hledání vyhledávače vracejí celou řadu odkazů, které neobsahují informace relevantní zadání. Návrh stávajícího webu byl výsledkem mnoha kompromisů, kvůli nimž byly do pozadí odsunuty stěžejní části. Zejména se jedná úplnost, integritu a kvalitu poskytovaných dat. Na druhou stranu je velmi jednoduché publikovat, rozšiřovat a web je díky decentralizované struktuře odolný vůči zhroucení.

V oblasti sémantického webu se převážně jedná o vývojovou aktivitu, kterou vytvářejí výzkumné oddělení univerzit a v minimálním množství také tvorba aplikací v průmyslové sféře. Jednotlivé vize a směry, kudy by se měl sémantický web dále odebírat jsou předmětem diskuzí mezi softwarovými a vědomostními inženýry.

V současné době je málo autorů webových stránek, kteří vytvářejí sémanticky správně psané weby. Taktéž existuje minimum serverů, které obsahují slovníky založené na ontologiích. Je to zapříčiněno přílišnou složitostí a z toho vyplývající náročností na vzdělání autorů.

Je předpokládán exponenciální růst webových zdrojů, které mohou zpracovávat softwaroví agenti¹⁰. Očekává se rozšíření sémantického webu i do fyzického světa. To znamená, že URI bude moci odkazovat i na objekty z reálného světa, ale popis pomocí RDF může být aplikován na reálná zařízení (mobilní telefony, stroje, atd.). Tato zařízení pak mohou být řízena stejně jako webovými agenti. Pro tyto účely již také byly vytvořeny první standardy.

Na českém internetu se sice začínají objevovat první texty věnující se tomuto tématu, jsou však pouze úvodem do problematiky. Neobsahují žádné obsáhlejší materiály, jež by byly vhodné pro výuku širšího spektra uživatelů, než je akademická sféra.

Současný vývoj webu lze charakterizovat z různých perspektiv:

- **Lokalizace zdrojů** - způsob, jak lidé vyhledávají informace (věci) na webu, se mění od jednoduchého, založeného na volném textu a klíčových slovech, směrem k více sofistikovaným sémantickým technikám.
- **Uživatelé** - webové zdroje se vyvíjejí od primární podoby určené pro vnímání člověkem do podoby použitelné jak pro člověka, tak pro počítače.
- **Webové úlohy a služby** - samotný web prochází proměnou, kdy již nebude chápán jen jako místo pro nalezení věcí, ale bude schopen věci tvořit.

Všechny tyto nové možnosti závisí podstatným způsobem na myšlence sémantiky – web se vyvíjí od informačních zdrojů, které explicitní sémantiku obsahují velmi málo nebo vůbec, ke zdrojům s bohatou sémantickou infrastrukturou. Citace z [3].

10 Softwarový agent je program, který vykonává dané úlohy za uživatele

6 Návrh řešení

6.1 Požadavky na systém

Aplikace by měla mít jednotný vzhled, který se zobrazuje stejně v používaných prohlížečích. Zároveň musí být zobrazitelná i na monitorech s nižším rozlišením. Velikost stránek musí být taková, aby se načetly v krátkém čase i na počítačích s pomalejším připojením. Celá prezentace musí být čitelná i s vypnutými styly, javascriptem i se zakázanými obrázky.

Výsledný produkt by měl obsahovat tyto funkce:

- Tutoriál obsahující jednotlivé články (s vysvětlením termínů a zkratk) a stromově generované menu.
- Demonstrační příklady vztahující se k jednotlivým článkům s barevně vyznačenou syntaxí.
- Testovací systém s generováním náhodných otázek a možností zobrazování dosažených výsledků.
- Správu uživatelů (registrace, nastavení práv, atd.).
- Porovnání editorů ontologií s možností vkládání komentářů.
- Administrační rozhraní pro správu uživatelů, komentářů a výsledků testů.

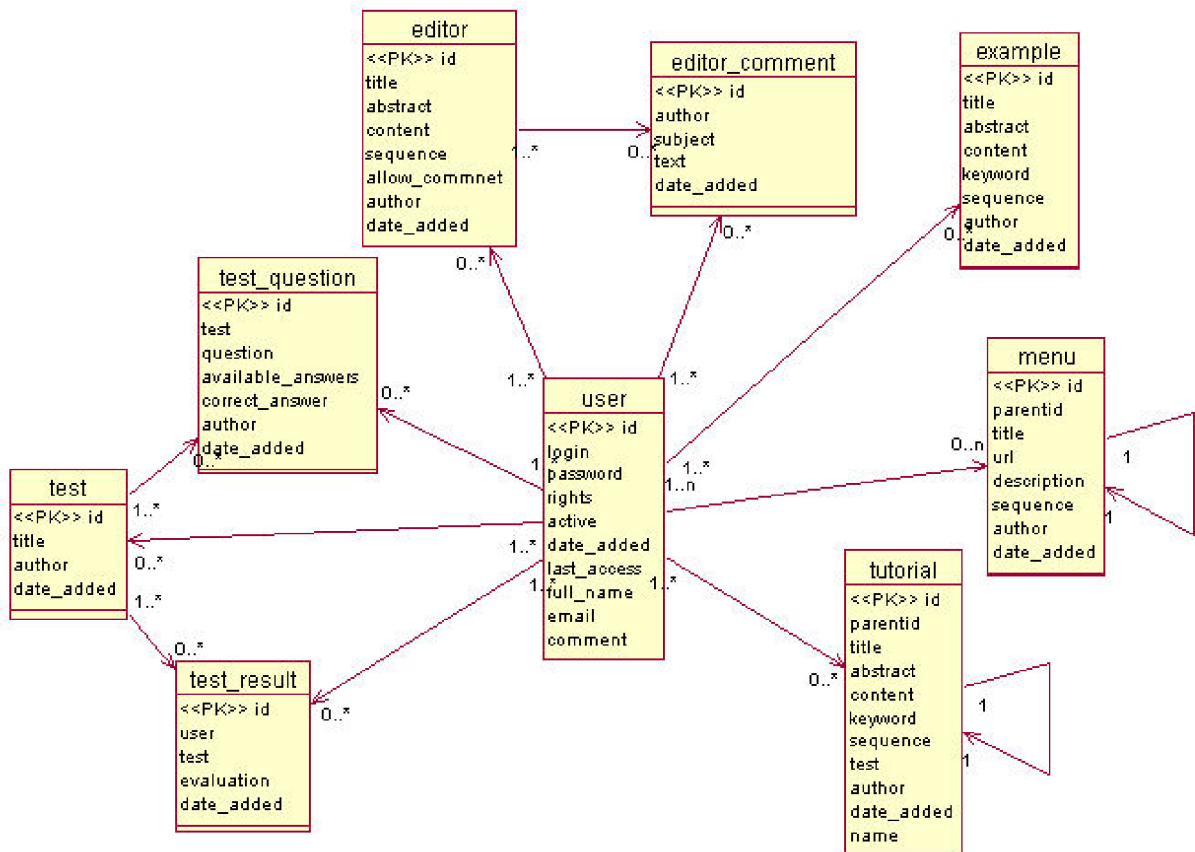
6.2 Vrstvy aplikace

Aplikační vrstva. Hlavní logiku celé aplikace bude tvořit interpret zdrojových kódů v jazyce PHP. Hlavním důvodem pro zvolení toho to jazyka byla dostupnost u takřka všech poskytovatelů webového hostingu. V této práci je použita verze PHP5, jež umožňuje vytváření objektově orientovaných zdrojových kódů, tak jak je známe například z jazyka C++.

Datová vrstva. Je tvořena databází MySQL, ve které jsou uloženy všechny texty a data. Tyto data jsou poté zpracovány pomocí jazyka PHP. Návrh tabulek v databázi si můžete prohlédnout na Obrázku 1.

Prezenční vrstva. Výstupním jazykem je XHTML 1.0. O vizuální stránku se starají styly CSS. Hlavní důraz je kladen na validitu¹¹ výstupního kódu. Vzhledem k možnosti zakázání interpretace javascriptu na klientském počítači, jsem tento jazyk zcela zavrhl. Všechny operace se proto řeší na serveru.

¹¹ Validita je platnost zdrojového kódu podle specifikace

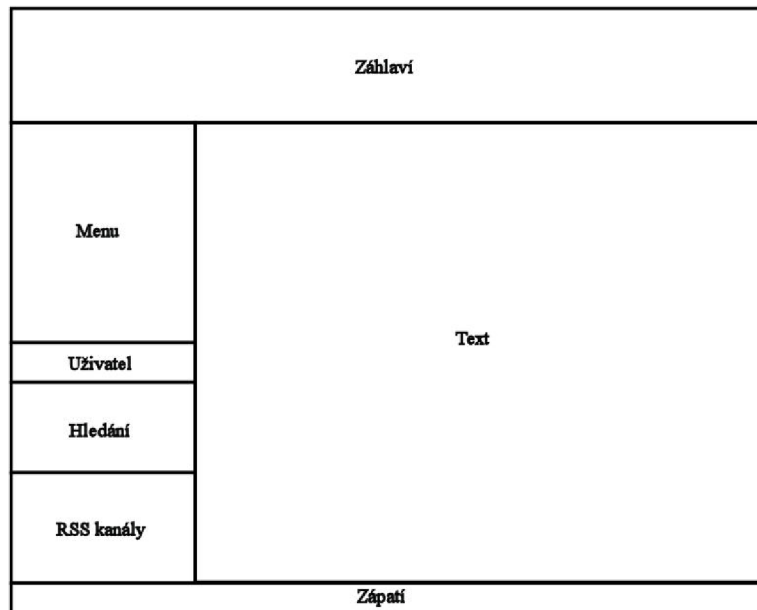


Obrázek 1: Diagram tříd

6.3 Struktura aplikace

Pro rozvržení celého webu jsem zvolil dvousloupcový design se záhlavím a zápatím. Menu je umístěno v levém sloupci. Tato skladba je často používána a proto bude ovládání zcela intuitivní.

Hlavní položky zůstávají neměnné (záhlaví, menu, hledání, zápatí). Mění se pouze obsah umístěný v části text. Šířka byla zvolena s ohledem na uživatele využívající rozlišení monitoru 800x600. Jak je vidět na obrázku 1, přibližně 70% stránky zabírá obsah, 15% statické části a 15% sloupec s menu a hledáním.



Obrázek 2: Rozložení webu

7 Implementace projektu

Celý projekt je implementován jako webová aplikace fungující na serveru Apache s modulem PHP5 a daty uloženými na databázovém serveru MySQL. Základem je objektový návrh doplněný o funkce pro vykreslování jednotlivých stránek.

Pro vykreslování složitějších struktur, jako jsou formuláře, výsledky fulltextového hledání nebo tvorba RSS kanálů jsou vytvořeny vlastní třídy. Zvláštní třídy jsou vytvořeny pro správu připojení k databázi a práci s daty v ní uloženými. Dále pak práce obsahuje vlastní třídu pro autorizaci a správu uživatelů.

Pro práci s jednotlivými částmi webové stránky (tisk fulltextových výsledků, seznam příkladů, generování testů, atd.) jsou vytvořeny vlastní funkce, které vyhodnotí daný úkol a vrátí výsledek jako pole řádků (ty se pouze vloží do dokumentu).

7.1 Základní rozvržení

Hlavním stavebním kamenem stránky je instance třídy **Page**, která je při každém provedení skriptu vytvořena pouze jednou (viz kapitola Vzor singleton). Při jejím vytváření se definuje jazyk, ve kterém bude výsledný dokument (webová stránka) odeslána klientovi.

Tato třída obsahuje funkce pro vkládání obsahu do jednotlivých částí XHTML dokumentu (hlavičky stránky, titulku, záhlaví, menu, obsahu a zápatí). Každá z těchto částí se ukládá do vlastního pole, proto můžeme kdykoliv během skriptu vkládat obsah do jakékoliv části stránky.

Před odesláním výsledku klientovi se vloží do XHTML kódu obsahy všech polí v pořadí, jak budou nakonec vykresleny. Výsledný kód se poté „obalí“ počátečními a koncovými značkami, tak aby odpovídal standardu XHTML 1.0.

Nastavení konfigurace je uloženo ve zvláštním souboru a do skriptu je vloženo pomocí příkazu **include**. Toto nastavení je vloženo na začátek skriptu jako definice jednotlivých konstant. To nám umožňuje měnit konfiguraci celé prezentace. Z toho vyplývá, že je jednoduché přemístit celou aplikaci na jiný server s minimální náročností (stačí pouze změnit několik hodnot v tomto souboru).

7.2 Vzor singleton

Návrhový vzor Singleton umožňuje autorovi tvořit objekty, tak že je u nich zajištěna absolutní unikátnost. To znamená, že je možné kdykoliv během skriptu zavolat první a zároveň jedinou instanci dané třídy.

Funkce obsahuje jedinou podmínku a to kontrolu, zda již neexistuje instance této třídy. Pokud neexistuje pokusí se vytvořit novou, jinak neprovede nic. V obou případech vrací odkaz na instanci této třídy.

7.3 Správa uživatelů

Přístup k některým částem webu je povolen pouze přihlášeným uživatelům. Jak bylo uvedeno výše, je pro správu uživatelů vytvořena vlastní třída. Ta obsahuje funkce pro přihlašování, odhlášení a získávání informací o uživateli. Protože je HTTP protokol bezstavový (aktuální stránka nemůže získat žádné informace o stavu na předchozí stránce), jsou pro mechanismus zachování identifikátoru uživatele využity **sessions** (relace, trvalé spojení mezi serverem a klientem).

V této relaci se uchovává unikátní identifikátor uživatele, jeho aktuální oprávnění (z čehož vyplývá, že se změna oprávnění projeví až po odhlášení a opětovné přihlášení uživatele) a času, kdy bude automaticky odhlášen.

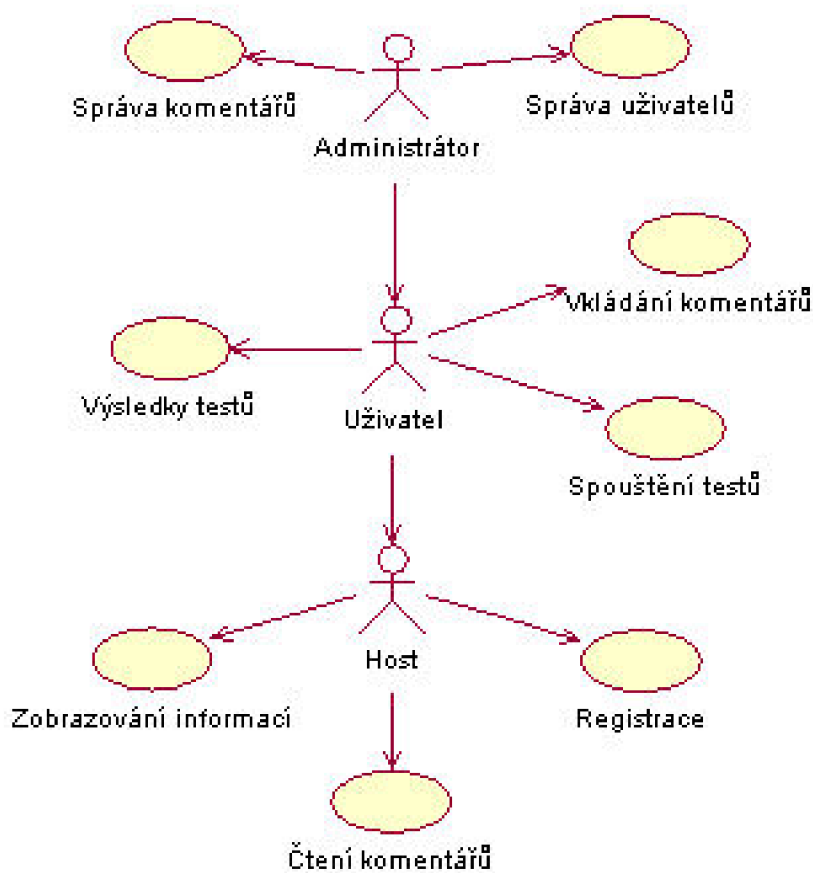
Před odesláním každé stránky (jež vyžaduje přihlášeného uživatele) se zkontroluje, zda pro něj existuje platná session a nevypršel limit pro automatické odhlášení. Je-li stále přihlášen nastaví se nový čas automatického odhlášení, jinak je uživateli zobrazena přihlašovací obrazovka.

7.3.1 Role uživatelů

Aplikace umožňuje vytvářet tři typy uživatelů:

- **Host** – může se pouze přihlašovat a zobrazovat informace o své osobě a číst komentáře
- **Uživatel** – smí spouštět testy, zobrazovat svoje výsledky, vkládat a číst komentáře

- **Administrátor** – má všechny práva jako uživatel a navíc se může přihlašovat do administračního prostředí, ve kterém lze editovat a odstraňovat komentáře a spravovat informace o uživatelích.



Obrázek 3: Use-Case diagram

7.4 Tutoriál a demonstrační příklady

Tutoriál je tvořen formou jednotlivých článků, které jsou vybírány z databáze. Ty obsahují kromě nadpisu, abstraktu a textu také klíčová slova, která se využívají při fulltextovém hledání. Protože je při generování výsledné stránky využito více databázových dotazů, které zaberou nejvíce výpočetního času, jsou tyto části ukládány do **cache** a při dalším zobrazení pouze vytisknuty na výstup.

Obdobně jsou navrženy i demonstrační příklady, které navíc využívají zvýraznění syntaxe pomocí kaskádových stylů.

7.4.1 Fulltextové hledání

Při fulltextovém hledání jsou brány v úvahu části článku v tomto pořadí: klíčová slova, abstrakt, nadpis a nakonec text. Při výpisu jednotlivých odkazů na články je brána v úvahu pravděpodobnost s

jakou je výsledek relevantní hledanému výrazu. Ke každému výsledku je pak přiřazena procentuální pravděpodobnost s jakou odpovídá zadání.

7.4.2 Cache

Vlastní cache tvoří již vygenerované části stránek, které jsou ukládány do souborů. Při prvním použití je soubor automaticky vytvořen. Pokud již takovýto soubor existuje a není starší než maximální doba uchovávání cache, je vložen do stránky. V opačném případě je obsah vygenerován znovu, vytisknut do stránky a jeho obsahem je přepsána cache.

7.5 Testy

K procvičení získaných znalostí slouží implementovaný testovací systém s ukládáním dosažených výsledků. Při spuštění tohoto testu jsou z databáze náhodně vybrány otázky, k nim přiřazené možné odpovědi a správná odpověď. Správné odpovědi jsou uloženy v poli, které je poté zamícháno.

Tyto otázky i s odpověďmi jsou uloženy v session. Po odeslání a následném vyhodnocení se aktuální výsledek uloží do databáze.

7.6 Propojení s databází

Veškeré prezentované informace jsou uloženy v databázi. Proto je vazba mezi interpretem jazyka PHP a databází velmi důležitou součástí. Připojení k databázovému serveru je perzistentní a vždy vytvořeno pouze jedno.

Jelikož je časově náročné vykonávat několik dotazů nezávisle na sobě, jsou ve třídě starající se o spojení s MySQL vytvořeny funkce, tak abychom si mohli dopředu připravit více dotazů jakoby do fronty a ty pak vykonat najednou.

Dále tato třída obsahuje funkce pro práci s výsledky, které nám vrátí databázový server. Proto lze přebírat výsledné hodnoty jako vícerozměrné pole, jež poté zpracováváme.

8 Závěr

Vytvořil jsem informační portál, který obsahuje výukové materiály srozumitelné uživatelům seznámeným s ontologiemi a tvorbou sémantického webu, a zároveň uživatelům, kteří se v této problematice neorientují. Mým cílem bylo vytvořit obsah, který by vždy navazoval na předchozí kapitoly a vysvětloval všechny zkratky a použité termíny pomocí demonstračních příkladů nebo vysvětlivek.

Implementoval jsem systém, jež je snadno rozšiřitelný o další položky, jak databázové, tak jednotlivé části webové prezentace. Umožňuje rychle rozšíření do prakticky neomezeného počtu světových jazyků. A je koncipován tak, aby byly stránky generovány co nejrychleji.

Tuto bakalářskou práci jsem si zvolil, protože se chci v budoucnu věnovat tvorbě webových aplikací a na nich založených vyhledávacích systémech. A také proto, že neexistuje česky psaný, ucelený webový portál věnující se tomuto tématu.

Sémantický web se rozvíjí pomalejším tempem než by se očekávalo, vzhledem k jeho obrovskému potenciálu. To je zapříčiněno příliš obtížnou tvorbou ontologií a správně sémanticky psaných webů, ale zejména malým povědomím autorů o těchto technologiích.

8.1 Další rozšíření

Tomuto projektu bych se rád věnoval i v budoucnu. Kromě vkládání dalších článků a tvorbě testovacích otázek bych rád rozšířil projekt o propracovanější administrační prostředí obsahující plnohodnotný editor článků.

Dále bych chtěl přeložit celou prezentaci alespoň do dvou jazyků – angličtiny a němčiny, s čímž souvisí rozšíření databáze o další položky týkající se vyhledávání.

Literatura

- [1] Staniček, P. *CSS Kaskádové styly Kompletní průvodce*, Computer Press, 2003.
- [2] Wikipedia, *Internet* [online], 2007, [cit. 8.5.2007] dostupné z URL:
<<http://cs.wikipedia.org/wiki/Internet>>.
- [3] Sklenák, V., *Metadata, sémantika a sémantický web* [online], 2004, [cit. 9.5.2007], dostupné z URL: <http://www.inforum.cz/inforum2004/pdf/Sklenak_Vilem1.pdf>.
- [4] Svátek, V., *Ontologie a WWW*, Brno, DATAKON, 2002.
- [5] W3 Consortium, *OWL - Web Ontology Language* [online], 2004, [cit. 8.5.2007] dostupné z URL: <<http://www.w3c.org/TR/owl-features/>>.
- [6] Wikipedia, *PHP - PHP Hypertext Preprocessor* [online], 2007, [cit. 7.5.2007] dostupné z URL: <<http://cs.wikipedia.org/wiki/PHP>>.
- [7] Gilmore, W. J. *PHP a MySQL 5*, Zoner Press, 2005.
- [8] Schlossnagle, G. *Pokročilé programování v PHP5*, Zoner Press, 2004.
- [9] W3 Consortium, *RDF - Resource Description Framework* [online], 2006, [cit. 8.5.2007] dostupný z URL: <<http://www.w3c.org/RDF/>>.
- [10] W3 Consortium, *Semantic web* [online], 2001, [cit. 5.5.2007] dostupný z URL: <<http://www.w3c.org/2001/sw/>>.
- [11] Smrž, P., Pitner T. *Sémantický web a jeho technologie*, ÚVT MU, 2004.
- [12] Pokorný, J. *Směrem k sémantickému webu*, MFF UK, 2005.
- [13] Hruška, T. *Studijní opora, Internetové aplikace (WAP) IV, část PHP* [online], 2006, [cit. 7.5.2007] dostupné z URL:
<<https://www.fit.vutbr.cz/study/courses/WAP/private/opory/OporaWAP4ProgramovaniServer.pdf>>.
- [14] Wikipedia, *XHTML* [online], 2007, [cit. 7.5.2007] dostupný z URL: <<http://cs.wikipedia.org/wiki/XHTML>>.
- [15] Wikipedia, *XML* [online], 2007, [cit. 7.5. 2007] dostupný na URL: <<http://cs.wikipedia.org/wiki/XML>>.

Seznam příloh

Příloha 1. Demonstrační obrázky

Příloha 2. Popis tříd a funkcí

Příloha 3. Manuál

Příloha 1. Demonstrační obrázky



Výsledky testu: Značkovací jazyky

Ve které části HTML dokumentu se musí být titulek stránky?

Vaše odpověď: **Hlavička (SPRÁVNĚ)**.

Ve kterém jazyce jsou psány dokumenty pro bezdrátový přenos?

Vaše odpověď: **WML (SPRÁVNĚ)**.

V HTML je znakčka <p> typem značky:

Vaše odpověď: **Sémantické (ŠPATNĚ)** správná odpověď zní: **Strukturální**.

Jaký je význam zkratky SVG?

Vaše odpověď: **Scalable Vector Graphics (SPRÁVNĚ)**.

K čemu slouží RSS kanály?

Vaše odpověď: **Získávání informací (SPRÁVNĚ)**.

Z jakého jazyka vychází XHTML 2.0?

Vaše odpověď: **XML (SPRÁVNĚ)**.

Spojením sémantiky a aplikace kterých jazyků vzniklo XHTML?

Vaše odpověď: **XML+HTML (SPRÁVNĚ)**.

Jaké konstrukce slouží k popisu značkovacích jazyků?

Vaše odpověď: **Nezadáno** musíte vybrat jednu z možností!

Z celkového počtu **8** otázek jste zodpověděl(a) **6** správně.

Vaše úspěšnost je: **75%**.

[Vaše výsledky](#)

Obrázek 4: Náhled výsledků testu



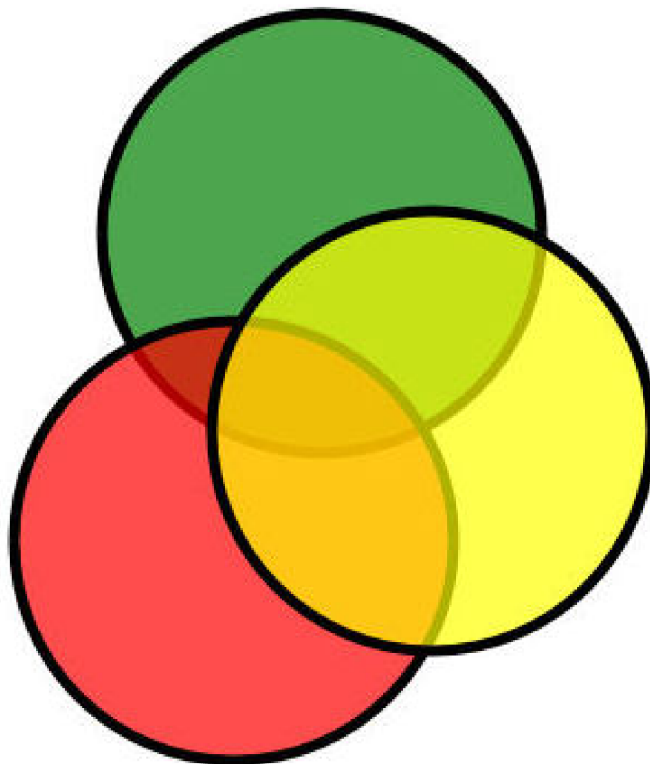
Demonstrační příklady

Scalable Vector Graphics

30. 4. 2007 23:37 • Petr Hanyáš

```
<?xml version="1.0"?>
<svg xmlns="http://www.w3.org/2000/svg">
  <g style="fill-opacity:0.7; stroke:black; stroke-width:2px;">
    <circle cx="150px" cy="100px" r="100px" style="fill:green;" />
    <circle cx="100px" cy="250px" r="100px" style="fill:red;" />
    <circle cx="200px" cy="200px" r="100px" style="fill:yellow;" />
  </g>
</svg>
```

Element <g> sdružuje jeho podelementy do jedné skupiny a ty po něm dědí všechny jeho vlastnosti (průhlednost 70%, obrys černou barvou o šířce 2 pixely). Každý kruh má definovaný střed pomocí X a Y souřadnice a barvu jakou má být vyplněn. Takto potom bude vypadat výsledek:



Obrázek. 5: Náhled demonstračního příkladu

Příloha 2. Popis tříd

Popis metod objektu Page:

add_page_xhtml_head(\$text)

Vloží do pole hlavičky stránky příslušný řádek.

add_page_text(\$text)

Vloží do výsledné stránky řádek jakéhokoliv textu (může obsahovat pole řádků).

set_language(\$language)

Je volán ihned po vytvoření instance třídy Page a nastaví jazyk výstupu stránky.

wrap_elements()

Obalí všechny použité elementy (tato funkce je volána před vlastním vykreslením stránky).

print_page()

Vytiskne stránku na výstup, odkud se rovnou odešle do prohlížeče.

create_cache(\$content)

Nejprve zkontroluje, zda neexistuje soubor s cache dané části a pokud ne, tak ji vytvoří

return_cache()

Je volán prostřednictvím funkce **create_cache()** a vrací obsah cache z disku nebo hodnotu false pokud takovýto soubor nebyl nalezen.

Popis metod objektu Auth:

login(\$login, \$password)

Pomocí předaných parametrů se pokusí přihlásit uživatele a vytvoří session. Při úspěchu otevře vyžadovanou stránku. V opačném případě načte přihlašovací formulář s chybovým hlášením.

logout()

Odhlásí uživatele a zruší session.

keep_alive()

Posune čas automatického odhlášení na maximální dobu specifikovanou v konfiguraci.

is_logged()

Vrací informaci, zda je uživatel přihlášen.

user_id()

Vrací unikátní identifikátor aktuálního uživatele.

update_last_access()

Změní v databázi hodnotu posledního přístupu na stránky na aktuální čas.

user_info(\$language)

Vrací pole informací o přihlášeném uživateli v jazyce předaném parametrem.

Příloha 3. Manuál

Požadavky na systém:

Apache 2.0

Modul PHP5 nebo vyšší

MySQL 4.1 nebo vyšší

Instalace:

Z CD přiloženého k této práci nakopírujte obsah složky *web* do veřejného adresáře Apache. Obsah složky *data* nakopírujte do neveřejného adresáře serveru Apache a otevřete si k editaci soubor *config.inc*. V tomto souboru nastavte absolutní cesty k veřejnému a neveřejnému adresáři webu, dále pak vložte správný název domény a nakonec zadejte správné konfigurační údaje pro databázi.

Nyní můžete importovat do MySQL databáze soubor *seweb.sql* nacházející se ve složce *sql* (to provedete například pomocí aplikace phpMyAdmin). Nyní by se měla po zadání správné adresy do webového prohlížeče zobrazit nainstalovaná prezentace.