DIGITALIZACE ČASOPISU FALSTAFF

Aleš Keprt

Katedra informatiky, FEI, VŠB – Technická Univerzita Ostrava
17.listopadu 15, 708 00 Ostrava–Poruba
Aley@Keprt.cz

Abstrakt
Příspěvek představuje řešení projektu Digitalizace časopisu Falstaff, jehož cílem bylo převedení článků z literárního časopisu Falstaff do digitální podoby a jejich zpřístupnění online přes běžné rozhraní www. Příspěvek se zaměřuje na podrobnosti realizace tohoto projektu z hlediska analýzy a designu řešení, u kterého bylo vsazeno na moderní technologie na bázi XML. Zároveň je provedeno srovnání s loni představenou databází sborníků konference TSW. Hlavním cílem příspěvku je zejména pojmenovat klíčové koncepty (key concepts), podstatné pro řešení projektů tohoto typu, a jejich odlišení od věcí nepodstatných.

1 Úvod

V rámci loňského jubilejního ročníku konference Tvorba softwaru došlo i ke zveřejnění CD s elektronickou databází všech sborníků této konference. V příspěvcích [1] a [3] jsme se pak mohli dočíst, jakými prostředky byla tato databáze vytvořena a jak vlastně vznikala. Tento projekt přitom nepřímo navázal na předchozí podobné projekty stejného autora v uplynulém desetiletí.

Čtením zmíněných příspěvků se nám mohou vybavit vzpomínky na vlastní předchozí projekty podobného typu – v našem případě to byl konkrétně projekt Digitalizace časopisu Falstaff, který byl realizován před třemi lety (začátkem roku 2002). Tento příspěvek se snaží porovnat oba zmíněné projekty a zdůraznit podstatné prvky a rysy shodné i rozdílné, což, jak věříme, může přispět při pracích na budoucích podobných projektech. Zároveň je problém rozveden i z hlediska softwarové analýzy, která v loňských příspěvcích [1] a [3] chyběla – ta se snaží především odkrýt, co jsou klíčové koncepty (key concepts) vedoucí k úspěšnému řešení.

2 Analýza

Proces digitalizace obsahu sborníků či časopisů se zjevně rozpadá do dvou zcela odlišných částí:

1. Převedení obsahu (textů, obrázků, vzorců, …) do vhodné elektronické podoby.
2. Vytvoření softwaru, který tento elektronický obsah zpřístupní čtenáři.

Klademe-li si za cíl vytvořit systém, se kterým budou uživatelé spokojeni, je třeba začít s analýzou ze strany uživatele. Samotnou technickou realizaci systému tedy sledujeme až ve druhé řadě.

2.1 Formát obsahu

Pojmem „obsah“ rozumíme texty, obrázky a veškerý další obsah publikací, které chceme zpřístupnit elektronicky. Největší podíl na obsahu má samozřejmě text, avšak ostatní typy

obsahu se od textu natolik liší, že není možno se omezovat jen na text. Například obrázky nebo vzorce se do obyčejného textu kódují velmi obtížně.

Jak známo, největší chyby v realizaci softwarových projektů vznikají hned na začátku analýzy. Stejné riziko je i v našem případě – chceme-li se vyhnout problémům v dalších fázích práce, je třeba velmi pečlivě uvážit, jaký formát obsahu použijeme.

Databáze TSW pokrývá všechny ročníky konference, tj. od roku 1975. U novějších ročníků od roku 1991, kdy se objevilo Windows 3.1 a masivně se rozšířily WYSIWYG textové editory pro Windows, jsou všechny příspěvky napsány ve stylu používaném dodnes. Tyto příspěvky tedy lze do jisté míry automaticky konvertovat do požadovaného tvaru, případně naskenovat a pomocí nějakého vhodného OCR[1] programu převést na text. (Případně je možno použít přímo výchozí tvar, což je obvykle MS Word. Tato zdánlivě „snadná“ varianta však není moc praktická.)

Se staršími příspěvky je to složitější. Vzhledem ke kvalitě současných nástrojů OCR je zřejmě hlavní problém ve skenování (a případně předzpracování, bylo-li by nutné), převod do textové podoby by měl být bezproblémový.

Pozornost je však třeba věnovat vzorcům – jsou-li vysázeny jako plovoucí (floating) prvky, považujeme je za obrázky a stejně je i zpracujeme. Jsou-li vzorce přímo v textu, je situace složitější. Můžeme se buď spolehnout na kvalitu OCR, nebo i tyto vzorce považovat za obrázky. Převod staršího dokumentu plného vzorců do elektronické podoby je však skutečně extrémně pracný. V takovém případě se pak nabízí náhradní řešení – celý dokument uchovat ve formě naskenovaných obrázků. Jedině tuto možnost máme i u dokumentů, které jsou natolik nečitelné, že jejich převod do textu není technicky možný. Nutno podotknout, že i v případech, kdy není technicky možné použít OCR, je třeba mít v databázi textovou verzi každého dokumentu, která může být použita při fulltextovém vyhledávání.

Hovoříme-li o formátu obsahu, máme na mysli volbu formátu, ve ktrém budou dokumenty v databázi uchovány. Jako jednoznačně nejlepší se nám jeví formát XML [11], který umožňuje jednoznačně definovat jak obsah, tak jeho formu (strukturu – je-li to třeba) a lze jej snadno strojově zpracovat (je-li to třeba) nebo zobrazovat jako HTML (což je vhodné v každém případě).

2.2 Databáze

Hovoříme-li o databázi článků, není tím předem určeno, jaký formát má tato databáze mít. Primárním cílem je především umožnit, aby uživatel našel požadované dokumenty – ať už pomocí menu, nebo vyhledáváním dle nějakých kritérií. Databáze tedy kromě samotného obsahu (diskutováno v sekci 2.1) musí obsahovat třídicí a vyhledávací údaje, případně rejstříky, pomocí kterých se uživatel k obsahu dostane.

Nabízejí se nejméně tři odlišné způsoby realizace, každý samozřejmě v mnoha variantách:

1. Relační databáze – dnes nejběžnější forma databází pro veškeré účely.
2. Síťová databáze – technologie před-relačních databází. V nových systémech se prakticky neobjevuje, třeba ale právě databáze sborníků TSW ji využívá.
3. XML databáze [4] – nejnovější ze tří zde zmíněných systémů. Zásadní problémem je zatím zejména nedostatek kvalitního softwarového vybavení.

Samotný obsah databáze (dokumenty) můžeme buď ukládat přímo do databáze (pokud to databázový software technicky umožňuje), nebo mít jednotlivé dokumenty v samostatných

[1] OCR = Optical Character Recognition. OCR software převádí obrázky obsahující text na skutečný text. Vstupem OCR jsou obvykle dokumenty naskenované z tištěných materiálů, výstupem pak dokumenty v nějaké textové formě (jako RTF, DOC, PDF, HTML, TXT…). OCR dokáže zachytit i běžné způsoby formátování (rozpozná tučné písmo, nadpisy, odstavce, jednoduché vzorce, apod.)

souborech na disku a v databázi mít na ně jen odkaz. První varianta bude zřejmě víc zatěžovat databázi, druhá varianta však neumožňuje přímé fulltextové vyhledávání (bude diskutováno později). Při práci se soubory navíc můžeme narazit na problém spravování jejich názvů.

2.3 Softwarové rozhraní

Softwarové rozhraní můžeme řešit jako offline, nebo online. Podobu offline verze je možné vidět na loňském CD s databází sborníků 30 ročníků TSW. Jak je popsáno v [1] a [3], systém HTML stránek vzájemně propojených odkazy byl vytvořen programem v Cobolu ze vstupních dat v XML formátu. Vstupní XML soubory však v tomto případě neobsahovaly samotný obsah, ale jen jakési „hlavičky" dokumentů, údaje pro rejstříky, apod.

Online verzi je možno vidět na stránkách časopisu Xan [10]. Software zajišťující online verzi je v mnohém jednodušší, neboť nemusí řešit výstavbu celého webového sídla najednou, všechny odkazy jsou tedy dynamické a není třeba řešit jedinečná jména všech souborů apod. Naopak těžší je tato varianta z hlediska technické realizace – program v Cobolu není možné použít, neboť spouštění programů v Cobolu není pochopitelně běžným webovým serverem podporováno.

3 Koncepty, na kterých stojí databáze článků časopisu Falstaff

Projekt Digitalizace časopisu Falstaff proběhl na začátku roku 2002 jako nekomerční aktivita tehdejších studentů informatiky na Univerzitě Palackého v Olomouci. Celý projekt byl hotový za zhruba 5 dní, z toho 3 dny probíhala analýza, půl dne analýza přecházela v design, 1 den zabralo kódování a poslední půlden proběhlo závěrečné testování a ladění[2]. Tento čas je příliš krátký na to, aby přinesl nějaké zásadní „vynálezy", přesto se výsledný systém zdá být velmi zdařilý. Vzhledem k tomu, že kódování proběhlo za pouhý jeden den, úspěšnou realizaci systému nutno přiřknout zejména použití moderních technologií. Zásadní koncepty, na kterých je systém založen, jsou popsány v této kapitole.

3.1 XML databáze přináší konzistenci

Všechny dokumenty jsou uloženy v XML podobě – každý dokument v jednom souboru, ve kterém je jak samotný obsah dokumentu, tak i hlavička obsahující jméno autora a všechny další informace vhodné pro kategorizaci, indexaci a vyhledávání dle různých kritérií. Informace o dokumentu nejsou uloženy nikde mimo tento soubor, přitom neexistují ani odkazy na jména souborů – souhrnem všech těchto souborů je XML databáze v podobě jediného velkého XML stromu, který je pouze na fyzické úrovni rozdělen do souborů popisujících jednotlivé dokumenty (neboť samostatné soubory se snáze spravují, editují a doplňují).

Tento způsob je přehledný a především zajišťuje konzistenci dat tím, že související údaje jsou vždy na jednom místě. Kdyby tomu tak nebylo, například kdyby rejstříky pro vyhledávání (např. jmenný rejstřík, věcný rejstřík, atp.) nebo soupisky příspěvků z jednotlivých ročníků byly uloženy ještě navíc i mimo samotných dokumentů, konzistenci by dlouhodobě nebylo možno zaručit.

[2] Navíc nutno dodat, že na analýze pracovali jen dva lidé (tímto děkuji za spolupráci Martině Chlupové), a další fáze projektu jsem dokončil sám – u malých nekomerčních projektů se tento postup jeví jako poměrně vhodný.

3.2 Data warehousing jako nástroj pro zrychlení práce s XML

Zásadním problémem při práci s XML daty je, že v podstatě neexistuje žádný vhodný software, který by jednoduše řečeno „uměl s XML databází to, co SQL umí s relační databází“ (čili SŘBD/DBMS). I běžné relační SŘBD (Oracle, MS SQL) sice podporují uložení a dotazování XML dat, pro indexovaní XML dat jsou tam však použity jednoduché relační přístupy [1], proto tyto systémy nemohou využít všech vlastností poskytovaných jazykem XML. Nové nativní XML databáze zatím není možné srovnávat s funkčností léta vyvíjených relačních SŘBD.

Samotnou XML databázi jako úložiště XML dat potřebujeme doplnit o nástroj, který umožní dokumenty kategorizovat, řadit, vyhledávat v nich apod. K tomuto účelu by měly sloužit XML dotazovací jazyky, jako XPath nebo XQuery, ovšem jejich použití nebylo v době našeho projektu vůbec možné. Rozhodli jsme se tedy zavést jakýsi jednoduchý data warehousing – jednoduchou formu datového skladu, jehož vstupem jsou XML dokumenty naší databáze. Úkolem datového skladu je vytáhnout (myšleno analyticky) z těchto dokumentů co nejvíc údajů, které mohou být použity při běhu systémů na odpovídání dotazů a příkazů uživatele. Datový sklad tedy vytvoří jakousi kopii XML databáze v relační podobě, se kterou lze dále pracovat běžnými nástroji pro práci s daty. Konzistence databáze je zajištěna tím, že datový sklad je vždy z hlediska aplikace „read–only“, tj. data v něm uložená jsou jen pro čtení a obnovují se vždy po updatu primární XML databáze. Jelikož časopis Falstaff vycházel v průměru jen dvakrát do roka a fungoval tedy podobně jako odborné konference, není obnovení datového skladu po každé změně či doplnění XML databáze žádným problémem.

K problematice datových skladů ještě dvě poznámky:

1. Data uložená v datového skladu by již měla být v takové podobě, aby nad nimi nebylo nutné provádět další složité relační operace (jako kombinované skládání projekcí ze spojení nebo obecně denormalizace apod.). Proto vůbec není třeba používat relační systémy s SQL a je možné použít i jednodušší formu úložiště datového skladu.
2. Komerční systémy realizující datové sklady jsou velmi sofistikované systémy nabízející mnohem širší funkcionalitu, než to, co používáme my. Přestože náš systém je mnohem jednodušší a menší, z principu se jedná o datový sklad, neboť nabízí read–only data připravená již ve tvaru potřebném v programu, který by jinak byl zpomalen prováděním dotazů nad XML databází.

3.3 XSLT překlady v teorii a v praxi

Jazyk XSLT slouží k překladu XML dokumentů do jiného tvaru a je dnes jedním z nejrozšířenějších jazyků rodiny XML. Nejčastěji se XSLT používá pro jednoduchý převod XML dokumentu do HTML stránky, kterou pak vidí uživatel. V našem případě a „v naší době“, konkrétně na začátku roku 2002, byl problém v tom, že tehdy ještě neexistovala žádná implementace XSLT verze 1.0 (ani pro HTTP server, ani pro www prohlížeče) a XSLT verze před 1.0 neumožňovaly vkládat libovolný HTML strom do XML stromu bez toho, abychom v transformační šabloně vyjmenovali všechny použité HTML tagy (což je samozřejmě nesplnitelné; jde o syntaktické omezení – podrobnější diskuze by byla nad rámec této práce).

Kvůli zmíněným problémům náš systém obsahuje vlastní jednoduchý XML parser a místo XSLT překladu skládá výstupní HTML soubor pomocí vlastního formátovacího algoritmu. Transformaci podle XSLT verze 1.0 podporuje až Internet Explorer od verze 6.0, proto ačkoliv veřejná verze našeho systému tento prvek nepoužívá, je v něm od počátku zavedena (jako sekundární) i možnost posílat na klienta místo HTML přímo XML soubor s odkazem na

příslušnou XSLT šablonu. Překlad do výsledného HTML tvaru by pak provedl přímo Internet Explorer 6.0, uživatelé ostatních prohlížečů by však neviděli nic.[3]

3.4 Fulltextové vyhledávání

Databáze článků z časopisu (stejně jako databáze článků ze sborníků konference) pochopitelně obsahuje velké množství textu, který je možné použít pro fulltextové vyhledávání. Tuto funkci považujeme za primární způsob hledání v dokumentech – možná je tento názor poněkud ovlivněn častým používáním Google a jiných vyhledávačů na internetu, avšak je více než zřetelné, že v minulosti rozšířené vyhledávání v textech pomocí věcných rejstříků je již dávno přežité, navíc vytvořit kvalitní věcný rejstřík je i extrémně pracné (pro důkaz viz článek [4]).

Databáze sborníků TSW fulltextové vyhledávání neobsahuje. Většina dokumentů je přítomna ve formě naskenovaných obrázků, které byly navíc nevhodně převedeny do 1bitové jasové hloubky (tj. černá na bílé, žádné mezistupně – lze se domnívat, že to bylo kvůli úspoře místa), což způsobilo, že tyto dokumenty jsou navíc prakticky nečitelné. Přitom sborníky samy jsou kvalitně natisknuté a mohly by být zpracovány systémem OCR velmi dobře. Dokumenty z několika posledních ročníků TSW jsou přítomny ve formě PDF souborů vytvořených převodem z Wordu, ovšem ani v nich není zavedena funkce fulltextového vyhledávání. (Přitom by to nemuselo být složité, viz diskuzi v sekci 2.1.)

Falstaff fulltextové vyhledávání může podporovat poměrně snadno díky tomu, že veškeré texty jsou ve formátu XML (do kterého pochopitelně musely být ručně převedeny). Samotná implementace vyhledávání je opět postavena na vlastním systému – ten je velmi jednoduchý a používá booleovský model vyhledávání slov v XML souborech. Pro zrychlení vyhledávání tohoto typu je možno sestavit term-dokument incidenční matici, pomocí které lze vyhledávání provádět velmi snadno. Samotná matice nemusí být nijak rozsáhlá, vzhledem k tomu, že počet dokumentů z časopisu typu Falstaff nebo konference typu TSW je poměrně malý (maximálně stovky, což nelze považovat za velký počet). Díky rychlosti dnešních počítačů pak při tomto počtu dokumentů i jednoduché implementace vyhledávání fungují uspokojivě.

Zajímavou alternativou k vlastnoručnímu řešení vyhledávání může být použití standardního webového vyhledávače. Např. Google může být snadno (a zdarma) použit pro vyhledávání v doméně jediného webového sídla. Tímto způsobem tedy outsourcujeme problém fulltextového vyhledávání na Google a sami řešíme pouze GUI. (Problematika offline vyhledávání bude diskutována níže.)

3.5 Databáze dokumentů není redakční systém

Pro software zajišťující běh www serverů publikujících články se ujal název „redakční systém“. V době přípravy databází Falstaffu a TSW nebyly ještě redakční systémy tolik rozšířené, v podstatě se o nich moc nevědělo a bylo běžné, že každý si tvořil softwarový systém na míru. Dnes se můžeme pokusit o zpětnou analýzu toho, zda bychom mohli použitím běžného redakčního systému (některé jsou dostupné zcela zdarma, např. phpRS [6]) dosáhnout lepšího výsledku, nebo alespoň dosáhnout stejného výsledku v kratším čase.

Ačkoliv to na první pohled může vypadat jinak, smysl a hlavní účel redakčního systému je jiný, než co potřebujeme pro účely databáze časopisu nebo sborníků. Redakční systém totiž zajišťuje zejména rychlé zveřejnění obsahu, proto je vhodný spíše pro informační servery (iDnes, iHNed) nebo jiná webová sídla, která dbají na časté aktualizace obsahu a rychlý kontakt s návštěvníky, případně i zpětnou vazbu. Lze je tedy najít na mnoha zájmových

[3] Nejnovější prohlížeče, jako Mozilla Firefox 1.0, dnes již XSLT 1.0 podporují. Na začátku roku 2002 však existovala jen ranná verze Internet Exploreru 6.0, natož aby už byl Firefox nebo dokonce XSLT podpora v něm.

stránkách, firemních stránkách nebo třeba i na stránkách univerzit. Pro naše účely se tedy redakční systém příliš nehodí, avšak jistě by bylo možné nějaký dostupný software tohoto typu lehce upravit („přešít na naši míru“) a použít. Zajímavým prvkem redakčních systémů, který zatím v databázi Falstaffu i TSW chybí, je třeba možnost čtenářských komentářů a diskuzí u jednotlivých publikovaných článků.

4 Srovnání Falstaffu a TSW

Databáze článků z časopisu Falstaff a databáze sborníků TSW mají srovnatelný počet dokumentů (stovky) i střední dobu mezi aktualizacemi (minimálně půl roku), jsou tedy dobře srovnatelné.

4.1 Rozdíly v řešení

Dvě porovnávané databáze se liší téměř ve všech ohledech. Jmenovitě:

Databáze TSW je postavena na síťové databázi a Cobolu, ze které je programově velmi složitým způsobem vytvořeno statické webové sídlo. Dokumenty jsou uchovávány primárně jako obrázky, bez sekundární textové formy, což znemožňuje fulltextové vyhledávání. Grafické rozhraní je velmi expresivní – razantní ostře barevné schéma může uživatele rušit. Statická podoba sídla je vhodná na umístění na CD, které je pak možno prohlížet na každém počítači s obyčejným WWW prohlížečem.

Databáze Falstaffu je postavena na moderních technologiích. Základem je XML databáze, ze které je výstup sestavován dynamicky pomocí PHP skriptů; je možno použít i XSLT transformace na straně klienta. Dokumenty jsou uchovávány výhradně v čistokrevné XML podobě. Forma časopisu nepožaduje řešení vzorců a obrázky u dokumentů jsou řešeny jako doplňkové soubory na serveru (obrázků je velmi málo).

Specializace na online řešení umožnila v případě Falstaffu snížit množství kódu. Celý systém, včetně pomocného datového skladu a algoritmů pro fulltext vyhledávání, byl implementován během jediného pracovního dne. Offline verzi lze pak snadno získat pomocí „web downloaderů“ – programů, které umějí projít webové sídlo a udělat lokální kopii. Tato kopie je pak čistokrevným statickým sídlem a je možné ji umístit na CD. Offline verzi je možno doplnit o fulltextové vyhledávání pomocí přídavného vyhledávácího programu (téhož, který je na serveru) – ukázku úspěšné implementace takového vyhledávání je možno najít např. na CD časopisu Vesmír [6].

4.2 Shodné rysy

Za shodné rysy diskutovaných databází můžeme považovat použití XML pro datové zdroje (i když způsob a úroveň použití je dosti rozdílná) a CCS stylů pro definici vzhledu HTML stránek (zde je však opět obrovský rozdíl mezi jemným a nenápadným vzhledem Falstaffu a ostrým barevným efektům TSW).

4.3 Věcné rozdíly

Při srovnávání obou databází je možno vidět i několik rysů, ve kterých se sborníky a časopis liší z podstaty věci (tj. bez ohledu na konkrétní implementaci). U odborných článků TSW se podařilo sestavit věcný rejstřík a je to z povahy těchto dokumentů docela pochopitelné. Naopak u literárního Falstaffu, kde drtivá většina dokumentů má smyšlený obsah, není možné podobný rejstřík jednoduše sestavit.

Falstaff naopak umožňuje rozlišovat články podle žánru či formy díla, a tak je možné filtrovat a vyhledávat dokumenty jako „povídka“, „báseň“, „sci-fi“ apod. Konference TSW typy příspěvků nerozlišuje.

5 Závěrečné shrnutí

V dnešní době mohutného rozvoje internetu je možno nalézt databáze podobné těm, o kterých byla řeč v tomto příspěvku, na mnoha místech. Jde tedy o téma velmi aktuální. Jak bylo ukázáno v příspěvku, základním kamenem úspěchu je použití XML a dalších moderních technologií, díky kterým se dříve složitá práce s „velkým množstvím“ dat a dokumentů mění na velmi snadnou práci s „několika málo“ dokumenty (přitom matematicky je počet dokumentů stále stejný).

V článku [3] se můžeme dočíst mj. to, že databáze sborníků TSW byla tvořena postupně třikrát v různých verzích a různým způsobem, neboť předchozí pokusy byly z různých důvodů neúspěšné. Náš projekt Digitalizace časopisu Falstaff byl naopak realizován velmi rychle a skončil jednoznačným úspěchem – stanovených cílů bylo dosaženo. Největším „nepřítelem“ našeho projektu nakonec nebylo technické zpracování, ale autorské právo. Dle platného zákona je totiž nutno při takovém opětovném publikování uměleckého díla, kterým každý takový dokument–článek jistě je, mít výslovné svolení každého autora. Při snaze kontaktovat autory literárních děl z Falstaffu jsme v mnoha případech obdrželi naopak výslovný nesouhlas s odůvodněním, že tato díla již nevyjadřují současné názory a/nebo postoje autora a jejich zveřejnění si autoři tedy nepřejí. V případě databáze sborníků TSW autoři o svolení ani žádáni nebyli. Aneb: Jsou problémy, které ani XML nevyřeší…

Literatura

1. Barashev, D., Krátký, M., Skopal, T. Modern Approaches to Indexing XML Data. In *Transactions of VŠB–TU Ostrava*, Computer Science and Mathematics Series, Ostrava, Czech Republic, Volume 2, 2003, ISBN 80-248-0455-7, ISSN 1213-4279.
2. Čevela, V. 30 let informací, inspirace a interakce – Tvorba softwaru a Programování Ostrava. Ve sborníku *Tvorba softwaru 2004.* Tanger, Ostrava, 2004. ISBN 80-85988-96-8. http://honor.fi.muni.cz/tsw/2004/025.pdf
3. Čevela, V. XML a XWEB jako nástroje pro tvorbu webového sídla s velkým množstvím křížových odkazů. Ve sborníku *Tvorba softwaru 2004.* Tanger, Ostrava, 2004. ISBN 80-85988-96-8.
   http://honor.fi.muni.cz/tsw/2004/028.pdf
4. Chaudhri, A. B., Rashid, A. , Zicari, R. *XML Data Management: Native XML and XML–Enabled Database Systems.* Addison Wesley Professional, 2003.
5. Kay, R. Sidebar: Sorting the Cards – the Way Indexing Used to Be. In *Computerworld*, 2004. http://www.computerworld.com/databasetopics/data/story/0,10801,96366,00.html?from=story_packa gečeská verze vyšla v Computerworldu číslo 8/2005.
6. *Vesmír 1994–2003*. CD s deseti ročníky časopisu. Vesmír, Praha, 2004.
   http://www.vesmir.cz/
7. iDnes (zpravodajský server MF Dnes) – http://www.idnes.cz/
8. iHNed (zpravodajský server Hospodářských novin) – http:/www.ihned.cz/
9. phpRS – http://www.supersvet.cz/phprs/vlastnosti.php
10. Xan – http://www.keprt.cz/povidky/
11. W3 Consortium. *Extensible Markup Language (XML) 1.0.*
    http://www.w3.org/TR/REC-xml/