

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SKUTEČNĚ OFF-LINE ČTENÍ RSS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ERIK DRESTO

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SKUTEČNĚ OFF-LINE ČTENÍ RSS

REALLY OFF-LINE RSS READING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ERIK DRESTO

VEDOUCÍ PRÁCE
SUPERVISOR

Doc. RNDr. Ph.D. PAVEL SMRŽ

BRNO 2009

Abstrakt

Práce se zabývá problematikou čtení *RSS* článků bez internetového připojení. Cílem bylo získat informace z oblasti čtení *RSS* článků, seznámit se s formátem *RSS*, poznat jeho strukturu a historii. Dále s dostupnými *RSS* čtečkami, zhodnotit jejich výhody a nevýhody z hlediska jednoduchosti rozšíření, aktualizace a možnosti převodu na mobilní zařízení. V práci se dále nachází stručný popis technologií, které se využívají při práci s *RSS* kanály. Jsou vysvětlené důvody zavedení formátu *RSS* pro publikaci článků na internetu, následně je vypracován návrh aplikace pro čtení *RSS* článků bez internetového připojení a popis implementace.

Výsledkem práce je vytvořená aplikace *OfflineRSS Reader* naprogramována v jazyku *Java*.

Abstract

This work deals with the problem of reading *RSS* articles without active internet connection. The goal of this work was to obtain information about ways of reading *RSS* articles, to get knowledge about *RSS* format, get to know its structure and history. Also to find existing solutions of *RSS* readers, review their advantages and disadvantages from point of expandability, actualization, and possibility to back port on mobile devices. This work also contains brief description of technologies, which are used in work with *RSS* channels. The reasons of using *RSS* for reading online articles are given in this work, thereafter the scheme of the application for reading *RSS* articles without active internet connection, and finally description of its implementation is worked out in it.

The result of this work is an application called *OfflineRSS Reader* created in *Java* programming language.

Klíčová slova

RSS, offline čtení *RSS*, *XML*, *OPML*, *Atom*

Keywords

RSS, offline *RSS* reader, *XML*, *OPML*, *Atom*

Citace

Erik Dresto: Skutočne off-line čítanie *RSS*, bakalárská práca, Brno, FIT VUT v Brně, 2009

Skutečně off-line čtení RSS

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Doc. RNDr. Ph.D. Pavla Smrža.

Uvedl jsem všechny literární prameny a publikace, z kterých jsem čerpal.

.....
Erik Dresto

20. 5. 2009

Poděkování

Chtěl bych poděkovat vedoucímu bakalářské práce Doc. RNDr. Ph.D. Pavlovi Smržovi za jeho odbornou pomoc a cenné rady při řešení této práce.

© Erik Dresto, 2009

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
1.1 Organizácia kapitol.....	3
2 Teoretický úvod.....	5
2.1 Značkovacie jazyky.....	5
2.1.1 Úvod.....	5
2.1.2 História.....	5
2.2 Jazyk XML.....	6
2.2.1 Úvod.....	6
2.2.2 Použitie.....	7
2.3 Formát RSS.....	7
2.3.1 Úvod.....	7
2.3.2 História.....	8
2.3.3 Použitie.....	9
2.3.4 Štruktúra dokumentu.....	10
2.4 Atom.....	12
2.5 OPML.....	12
3 Dostupné riešenia.....	14
3.1 Úvod.....	14
3.2 Spolupráca webových a desktopových aplikácií v rámci off-line režimu.....	15
3.3 Webové aplikácie.....	16
3.3.1 Google Reader.....	16
3.3.2 Windows Live.....	16
3.3.3 Tiny Tiny RSS.....	17
3.4 Desktopové aplikácie.....	17
3.4.1 Mozilla Thunderbird.....	17
3.4.2 RSSOwl.....	18
3.4.3 FeedDemon.....	18
3.5 Rozšírenia desktopových aplikácií.....	19
3.5.1 Feedview.....	19
3.5.2 News Fox.....	19
3.5.3 Sage.....	20
4 Návrh aplikácie.....	21

5	Implementácia.....	23
5.1	Objektový model	23
5.2	Čítačka XML súborov	23
5.3	RSS modul.....	24
5.4	Off-line sťahovač.....	24
5.5	Testovanie.....	25
6	Záver	27
	Literatúra	28
	Zoznam príloh.....	29

1 Úvod

V súčasnosti si nevieme predstaviť život bez informačných technológií, zasahujú do všetkých oblastí našich záujmov. V rámci internetu vznikla potreba zverejňovať informácie o dátach podávaných na rôznych internetových portáloch, či už články, blogy alebo rôzne iné často aktualizujúce sa internetové obsahy. Z počiatku sa hľadal nový formát, ktorý by umožňoval výmenu dát medzi internetovými portálmi, aby mohli rýchlo a efektívne navzájom odkazovať na novo zverejnené dáta. Hlavné charakteristiky hľadaného formátu boli univerzálnosť, otvorenosť, prenositeľnosť a jednoduchosť. Vzniklo niekoľko formátov pre tieto účely, ale buď neboli otvorené, alebo sa nerozšírili a používali sa v rámci malých skupín portálov pre vnútornú komunikáciu. K zmene došlo s príchodom prvej verzie formátu *RSS (RDF Site Summary)*, ktorý vychádzal z prvého konceptu formátu *RDF* a splňoval všetky charakteristiky hľadaného formátu. K veľkému rozšíreniu došlo až v druhej verzii tohto formátu, ktorý bol premenovaný na *RSS (Rich Site Summary)*. V priebehu ďalšieho vývoja sa opäť zmenilo vysvetlenie skratky *RSS (Really Simple Syndication)* a štruktúra tohto formátu. *RSS* ako textový formát je pre človeka dobre čitateľný, nakoľko využíva *XML* formát pre reprezentáciu dát. Obsahuje základné informácie o zverejnenom článku, jeho názov, stručný popis a odkaz, kde si užívateľ môže daný článok prečítať.

Tento dokument sa zaoberá problematikou čítania článkov uvedených v *RSS* dokumente bez aktívneho pripojenia k internetu. Ako vstupné dáta uvažuje pripojenie k internetu a *RSS* dokument, ktorý je pomocou aplikácie spracovaný tak, aby bolo možné zobrazit' obsah článkov špecifikovaných v *RSS* dokumente v podobe *HTML* stránok bez aktívneho pripojenia k internetu.

Po prečítaní práce by mal byť čitateľ schopný rozumieť problematike *RSS* dokumentov, poznať ich štruktúru, vyčleniť jednotlivé dáta pre jeden článok a pracovať s aplikáciou *OfflineRSS Reader*, ktorá vznikla ako výsledok tejto práce.

1.1 Organizácia kapitol

V druhej kapitole sa budeme venovať teoretickému úvodu do problému, ktorý je rozdelený na päť podkapitol. Prvou časťou je úvod do značkovacích jazykov, kde spomenieme hlavne históriu, aby sme pochopili, ako a prečo vznikli ďalšie jazyky ako výsledok vývoja. Druhou časťou je jazyk *XML*, ktorý patrí do rodiny značkovacích jazykov, treťou časťou je formát *RSS*, ktorý je považovaný za štandardizovaný *XML* dokument. Ako štvrtý si predstavíme konkurenčný štandard *Atom*. Posledná časť je venovaná formátu *OPML*, ktorý sa používa ako prostriedok pre publikáciu *RSS* kanálov.

V tretej kapitole sa budeme zaoberať dostupnými riešeniami *RSS* čítačiek, porovnáme ich výhody a nevýhody, jednoduchosť rozšírenia, aktualizácie a možnosti prevodu na mobilné zariadenia.

Štvrtá kapitola je venovaná návrhu aplikácie, ktorá je výsledkom tejto práce. Vysvetlíme, prečo sme zvolili implementáciu nového programu, očakávané funkcie a výsledky.

Piata kapitola popisuje implementáciu navrhnutej aplikácie v programovacom jazyku *Java*, spôsob implementácie zaujímavých častí, riešenie problémov a testovanie. Túto kapitolu dopĺňa Príloha 2. Programová dokumentácia programu *Offline RSS Reader*.

Záverom zhodnotíme výsledky práce a vytvorenej aplikácie, porovnáme ju s riešeniami spomínanými v tretej kapitole a uvedieme niektoré rozšírenia, ktoré môžu zvýšiť presadenie navrhnutej aplikácie.

V prílohách nájdeme užívateľský manuál pre vytvorenú aplikáciu, programovú dokumentáciu vo forme *JavaDoc*, plagát prezentujúci prácu a CD s programom samotným.

2 Teoretický úvod

2.1 Značkovacie jazyky

2.1.1 Úvod

Značkovacie jazyky sú formálne definované jazyky, ktoré kombinujú text a informácie o texte. Tieto informácie o texte sú v podobe značiek a udávajú inštrukcie pre spracovanie daného textu. Typickým rysom značkovacích jazykov sú znaky so špeciálnym významom, ktoré oddeľujú normálny text od značiek. Značky môžeme rozdeliť na logické, ktoré definujú štruktúru textu a na vizuálne, ktoré definujú formát. Toto delenie nespomíname náhodou, v minulosti a aj v súčasnosti hrá vo vývoji a použiteľnosti značkovacích jazykov významnú rolu.

Využitie značkovacích jazykov je veľmi rôznorodé, napriek tomu majú jednu spoločnú črtu, a to je dátová nezávislosť, čoho dôsledkom je dlhá životnosť. Typickou vlastnosťou značkovacích jazykov je zjednodušenie spracovania textu a vyhľadávania v ňom.

2.1.2 História

Počiatky vývoja značkovacích jazykov začali v šesťdesiatych rokoch 20. storočia, ako dôsledok dopytu pre publikačný priemysel. Termín značka je odvodený od tradičného označovania rukopisu, kde boli popísané inštrukcie pre tlač daného rukopisu. V minulosti túto úlohu vykonávali ľudia, tzv. značkovači, označovali aký druh písma, veľkosť, alebo štýl zvoliť pre dané časti rukopisu. Takto označovaný dokument predali ďalej na sadzbu a tým bol dokument pripravený na tlač. Táto technika bola veľmi časovo náročná, hľadalo sa riešenie v podobe automatizácie celého procesu.

Myšlienka značkovacích jazykov bola prvýkrát prezentovaná *Williamom W. Tunnicliffeom* na konferencii v roku 1967, pod názvom všeobecné kódovanie. Podstatou jeho myšlienky bolo použiť v texte nejaké všeobecné príkazy, ktoré sa neviazali ku konkrétnej sade riadiacich úloh. V podstate sú tieto obecné príkazy priamymi predkami značiek a značkovacích jazykov. V sedemdesiatych rokoch viedol vývoj štandardu *GenCode* pre publikačný priemysel, ale toto sa ukázalo ako slepá ulička a vývoj tohto štandardu skončil. Napriek tomu veľa poznatkov z tohto projektu si našlo neskôr uplatnenie a dokázali, že rôzne dokumenty pre rôzne oblasti vyžadujú rôzne možnosti značkovania.

Korene značkovacích jazykov, tak ako ich poznáme dnes, sa viažu k aktivitám firmy *IBM*, pre ktorú pracoval *Charles F. Goldfarb*. Tento muž je v súčasnosti považovaný za otca značkovacích jazykov. V roku 1969 začal pracovať na projekte, ktorého cieľom bolo vytvoriť informačný systém pre právnu kanceláriu. Mal sa skladať z troch častí, a to z aplikácie pre editáciu textu, vyhľadávacieho

systému a programu generujúci výstupný dokument pre tlač. Pri práci na tomto projekte narážal na problém s nekompatibilitou a zistil, že každá časť projektu vyžaduje iný druh procedurálneho značkovania. A tak pre účely tohto projektu vznikol nový jazyk, ktorý výhodne zovšeobecňoval značkovanie. V roku 1973 bol tento jazyk, ktorý nazval *GML (Generalized Markup Language)*, hotový a pripravený na prvú implementáciu.

GenCode v podobe jednoduchého značkovania sa nepresadil, napriek tomu *GML* budovalo koncept formálne definovaného typu dokumentu (*DTD*) s explicitne stanovenou štruktúrou a tvarom elementov. Princípy jazyka *GML* sa osvedčili a na jeho základoch sa začal vytvárať nový metajazyk, ktorý by umožňoval definovanie vlastných značiek. Štandardizačná organizácia *ANSI* a združenie *CGA* spojili svoje sily a v rámci projektu *ODA (Open Document Architecture)* mali za cieľ poskytnúť štandardnú architektúru pre vytváranie, výmenu, uchovávanie a spracovanie rôznorodých elektronických dokumentov nezávislých na platforme. Výsledkom ich snahy bol špecifikačný jazyk *SGML*.

SGML (Standard Generalized Markup Language) vznikol v roku 1986 ako štandard, ktorý definuje spôsob zostavenia množiny značiek vhodných pre popis a tvorbu štruktúrovaných dokumentov, nezávislých na platforme a aplikáciách. Tento metajazyk zaznamenal veľký úspech, ale zaplatil krutú daň za svoju všeobecnosť. Príliš veľká zložitosť mala za následok veľké náklady detailnej implementácie, tým bolo presadenie tohto metajazyka problematické a presadil sa len vo veľkých spoločnostiach.

2.2 Jazyk XML

2.2.1 Úvod

XML (eXtensible Markup Language) je potomkom rodiny jazykov *SGML*, bol vyvinutý a štandardizovaný konzorciom *W3C*, hlavným účelom je špecifikácia pre písanie vlastných značkových jazykov. Je považovaný za rozšíriteľný jazyk, pretože umožňuje užívateľom definovať vlastné typy značiek. V súčasnej dobe sa jedná o jeden z najdôležitejších formátov pre výmenu dát štruktúrovaným spôsobom. Popisuje dáta nezávisle na ich platforme, čím sa stáva veľmi silným prostriedkom pre zaistenie prenositeľnosti dát. Je to otvorený formát, do ktorého môžeme dopĺňať značky podľa potreby, z pohľadu kompatibility je možné dáta transformovať na iný formát. Značky tohto jazyka majú pevnú gramatiku, ktorú je nutné striktné dodržiavať.

2.2.2 Použitie

Hlavnou výhodou *XML* dokumentov je, že informácie v nich uložené sú použiteľné viacerými typmi aplikácií, nakoľko je tento formát univerzálny. Je možné zdieľať informácie medzi programami bez nutnosti akejkoľvek konverzie. *XML* dokumenty môžeme rozdeliť do dvoch veľkých skupín.

- Dátovo orientované dokumenty - hlavným účelom je prenos dát medzi aplikáciami. Typické pre tieto dokumenty je používanie veľkého množstva rôznych značiek. Obsahujú štruktúrovanú informáciu, ktorá má podobu čísel alebo krátkych znakov. Takéto dokumenty sú v súčasnosti využívané ako konfiguračné súbory pre rôzne aplikácie.
- Textovo orientované dokumenty - typické tým, že obsahujú menej značiek a väčšiu časť zaberá samostatná informácia. Značky obaľujú reťazce, ktoré predstavujú vety. Tieto dokumenty nachádzajú uplatnenie v prípade kníh, internetových stránok alebo programových dokumentáciách.

2.3 Formát RSS

2.3.1 Úvod

Formát *RSS* vychádza z jazyka *XML*, bol vyvinutý pre účely výmeny dát medzi internetovými portálmi, aby mohli rýchlo a efektívne navzájom odkazovať na novo publikované dáta. V dnešnej dobe sa táto technológia rozšírila natoľko, že umožňuje užívateľom internetu prihlásiť sa k odberu dát, ktoré publikuje portál. Tieto dáta sú publikované pomocou *RSS* kanálu, čo je vlastne *RSS* dokument, ktorý obsahuje informácie o novo zverejnených dátach. Je typické, že tieto *RSS* dokumenty často menia svoj obsah a sú pridávané nové informácie. Štruktúru a vlastnosti tohto formátu si popíšeme v kapitole 2.3.4 Štruktúra dokumentu.

Užívateľ potrebuje určitý druh softwaru, aby sa mohol prihlásiť k odberu dát. Takéto aplikácie sa nazývajú *RSS* čítačky, šetria čas a prenesené dáta, nemusíme pravidelne a ručne kontrolovať naše obľúbené internetové portály, pri použití čítačky budeme upozornení na pridanie novej udalosti. V súčasnosti je ponuka veľká a je možnosť si vybrať takú aplikáciu, ktorá bude vyhovovať každému. Či už človek hľadá program s veľkým množstvom nastavení a prepracovaným prostredím alebo veľmi jednoduchý program, ktorý poskytuje základné informácie. Existujú voľne šíriteľné druhy takýchto čítačiek alebo profesionálne komerčné riešenia pre náročných užívateľov. *RSS* sa v súčasnosti rozšíril natoľko, že jednoduché čítačky *RSS* dokumentov sú súčasťou internetových prehliadačov.

2.3.2 História

Pred formátom *RSS* existovalo pár podobných formátov, ktorým sa nepodarilo rozšíriť a zanikli alebo sa využívajú len pre účely jednej aplikácie, pre ktorú boli vymyslené. Ich hlavným problémom bola uzavretosť, poprípade neboli vymyslené pre univerzálne použitie. Ako prvý predchodca formátu *RSS* vznikol firmou *Apple* navrhnutý takzvaný *MCF* (*Meta Content Framework*), ktorý špecifikoval štruktúru dát o internetových portáloch, poprípade iných dátach. Po prerušení vývoja tohto projektu, hlavný predstaviteľ *Ramanathan V. Guha* odišiel z firmy *Apple* a začal pracovať pre *NetScape*.

Guha sa rozhodol v projekte pokračovať a preformovať pôvodné *MCF* do *XML* formátu. V roku 1997 predal špecifikácie tohto formátu medzinárodnej štandardizačnej organizácii *W3C*. Výsledným formátom, ktorý vznikol kombináciou *XML* a *MCF* bol formát *RDF* (*Resource Description Framework*). Jedná sa o dátový model popísaný metadátami. Tento formát sa stal univerzálnou metódou pre konceptuálne modelovanie.

Ďalším odlišným formátom sa stal *CDF* (*Channel Definition Format*), vyvinutý firmou *Microsoft* pre internetový prehliadač *Internet Explorer 4*. Tento formát sa nikdy nepresadil, najmä z dôvodov veľkosti prenášaných dát, v čase tohto formátu väčšina ľudí používala vytáčané pripojenie k internetu. Postupom času tento formát zanikol.

Prvá verzia formátu *RSS* neskôr označovaná ako 0.9 (*RDF Site Summary*) vytvorená firmou *NetScape*, ktorá bola používaná na ich internetovom portály. Tento formát zaujal verejnosť svojím veľkým potenciálom, čo viedlo k mnohým diskusiám, veľa nápadom a pripomienkam ako tento formát zlepšiť. Ako odpoveď na tieto diskusie bol vytvorený prototyp novej verzie formátu, označovaný ako *RSS 0.91* (*Rich Site Summary*), jeho autorom bol *Dan Libby*. Hlavným rozdielom bolo zjednodušenie formátu. Boli spracované nové časti, ktorých autorom bol *David Winer*. Tieto nové prvky prevzal zo svojho internetového blogu *Scripting News*, kde bol tento formát publikovaný.

V roku 2001 firma *AOL* kúpila *NetScape*, ktorý sa stal dcérskou spoločnosťou. V rámci celkovej zmeny štruktúry firmy sa rozhodli zmeniť vzhľad a vlastnosti stránok tohto úspešného internetového portálu. V rámci týchto zmien došlo k odstráneniu podpory *RSS* formátu. Výsledkom bolo, že tento formát ostal bez vlastníka práve v čase, keď sa stával populárnym a používaný veľkým počtom užívateľov.

Mnohí nadšenci nechceli, aby tento formát zanikol. Tak vznikla vývojárska skupina nazývaná *RSS-DEV*. V rovnakom čase *David Winer* uverejnil upravenú verziu formátu pod označením *RSS 0.91* pre internetový portál *Userland*. Nakoľko tento formát využívali vo svojich produktoch, tak označoval firmu *Userland Software* za vlastníka tohto formátu. Ani jedna strana nemala žiadne oficiálne právo na toto meno alebo formát a neustále hádky o tom, komu tento formát patrí, vyústili k rozdeleniu vývoja medzi tieto dve skupiny.

Skupina *RSS-DEV* začala s vývojom *RSS* verzie 1.0 založenej na verzii 0.9. Bola založená na špecifikácii *RDF*, ale bola viac modulárna.

Ako odpoveď na tento vývoj zverejnil *David Winer* *RSS* verzie 0.92. Malý počet zmien a predpokladane formát kompatibilný s *RSS* 0.91. Neskôr pokračoval s uverejnením návrhu verzie 0.93, ktorý bol takmer identický s verziou 0.92 a následne nato vydal návrh verzie 0.94, kde vrátil späť všetky zmeny uskutočnené v 0.93, jediný rozdiel bolo pridanie nového elementu.

V roku 2002 *Winer* zverejnil finálnu podobu verzie 0.92, ktorá je v súčasnosti známa pod označením *RSS 2.0 (Really Simple Syndication)* a používa sa dodnes. Do tejto špecifikácie sa nedostal ani nový element pridaný vo verzii 0.94, ale umožňoval pridávať vlastné elementy. Táto verzia formátu sa začala šíriť po internete a je v súčasnosti považovaná za štandard. V roku 2003 *Winer* pripísal vlastníctvo tohto formátu organizácii *Userland Software*.

2.3.3 Použitie

RSS dokument je dátovo orientovaný *XML* dokument, ktorého hlavným účelom je výmena dát medzi aplikáciami. Celý formát *RSS* je založený na myšlienke uverejňovania zmien na webovom portáli v špeciálnom formáte. Všeobecne môžeme povedať, že autor internetového portálu vytvorí jeden alebo viac *XML* súborov, na ktoré umiestni odkaz na stránkach. Následne do týchto súborov v *RSS* formáte uloží informácie napríklad o publikovaných článkoch a pravidelne ich aktualizuje. Užívateľské programy sťahujú a analyzujú tieto *XML* súbory a upozornia užívateľa na novo zverejnené články. Z toho vyplýva, že použitie môžeme rozdeliť na dve skupiny, a to z pohľadu administrátora internetového portálu a z pohľadu užívateľských aplikácií.

Z pohľadu administrátora sa jedná o pravidelné aktualizovanie tohto súboru, či už pridávanie novo publikovaných informácií alebo odstraňovanie starých a neaktuálnych dát. Je nutné dodržiavať štruktúru tohto formátu, lebo inak môže dôjsť k nečitateľnosti týchto súborov užívateľskými aplikáciami. Môže zvoliť jednu z verzií *RSS* formátu, ale túto skutočnosť musí špecifikovať v hlavičke. V prípade, že úspešne vytvoril *XML* súbor, umiestni odkaz na svoje stránky, poprípade môže odkaz na tento *XML* dokument špecifikovať v metadátach internetového portálu. Tým končí práca administrátora na strane webového portálu a *RSS* kanál je pripravený pre používanie užívateľských aplikácií.

Každý užívateľ si môže zvoliť, ku ktorému *RSS* kanálu sa prihlási a pomocou akej aplikácie bude sledovať jeho obsah. Každá aplikácia funguje na inom princípe, ale v podstate robia to isté, takže popíšeme všeobecné riešenie. Užívateľ ak pri prehliadaní internetového portálu narazí na odkaz pre *RSS* kanál, tak tento odkaz použije ako vstup pre jeho aplikáciu. Tá sa pripojí na tento kanál a stiahne obsah *XML* súboru a analyzuje ho. Ako výstup poskytne užívateľovi zoznam publikovaných článkov. Informácie o danom článku, ktoré poskytne závisia na verzii *RSS* formátu, ktorý bol použitý

pre vytvorenie daného RSS kanálu. Ako povinné parametre sú názov daného článku, stručný popis článku a odkaz, kde sa nachádza celý obsah článku. Vo verzii RSS 2.0 je povinný aspoň jeden z dvojice parametrov názov článku alebo popis. Ďalej môžeme nájsť rôzne voliteľné parametre, ktoré si popíšeme v nasledujúcej kapitole.

2.3.4 Štruktúra dokumentu

Formát RSS vychádza z jazyka XML, a preto je pre tento dokument špecifický obsah v štruktúrovanej forme. Pre dokumenty založené na jazyku XML je typické, že sa skladajú z textového obsahu a textových značiek (*tagov*), je možné definovať svoje vlastné značky, ktoré určujú štruktúru dát. Syntaktické pravidlá týchto dokumentov sú veľmi jednoduché a tiež veľmi presné.

Pre dokumenty RSS je typická logická štruktúra, ktorá rozdeľuje dokument do pomenovaných jednotiek a podjednotiek, nazývaných elementy. Jednotlivé elementy tohto formátu a ich parametre sa postupným vývojom menili. Hlavnú štruktúru celého dokumentu tvoria tri elementy, ktoré si postupne popíšeme, ich význam, syntax a zmeny štruktúry jednotlivých jednotiek počas vývoja formátu RSS. Zameriame sa hlavne na povinné elementy a ich vlastnosti.

Prvým a najdôležitejším elementom je hlavička súboru, v ktorom špecifikujeme verziu XML, RSS a kódovanie, ktoré sme použili pre tento dokument. Táto informácia je dôležitá pre užívateľské aplikácie, ktoré sa snažia analyzovať daný dokument. Pre prvú verziu RSS formátu 0.9 bol typický len jeden element, ktorý obsahoval verziu XML jazyka, pre ostatné verzie bol tento element rozšírený o typ použitého kódovania dokumentu. Od verzie 0.91 sa stal povinným aj druhý element, ktorý popisoval verziu zvoleného RSS formátu. Hlavička súčasných dokumentov RSS verzie 2.0 má napríklad takýto tvar:

```
<?xml version="1.0" encoding="windows-1250"?>
<rss version="2.0">
...
</rss>
```

Druhým elementom je *Channel*, ktorý obsahuje informácie typické pre daný RSS kanál a je podjednotkou elementu RSS. Pre všetky verzie RSS formátu sú typicky povinné elementy ako *title*, ktorý obsahuje názov RSS kanálu, ďalej je to element *description*, ktorý obsahuje stručný popis kanálu, napríklad aký druh informácií je publikovaný cez tento kanál, nasleduje element *link*, ktorý obsahuje adresu internetového portálu, pre ktorý bol tento dokument vytvorený. Vo verzii 0.91 sa vyskytol ako povinný nový element *language*, ktorý popisoval jazyk daného kanálu, ale v ďalších verziách sa stal nepovinným. Ďalším typicky používaným elementom, ktorý bol pridaný vo verzii 0.91 ako nepovinný, je *pubDate*, ktorý obsahuje časové informácie poslednej aktualizácie. Tento element obsahuje ešte pár nepovinných elementov, ktoré sa často nevyskytujú v RSS dokumentoch. Element *Channel* súčasných dokumentov verzie 2.0 má napríklad takýto tvar:

```

<channel>
  <title>DSL.sk</title>
  <link>http://www.dsl.sk/</link>
  <description>Digitálny Svet pod Lupou.</description>
  <language>sk</language>
  <pubDate>Tue, 10 Mar 2009 11:05:00 GMT</pubDate>
</channel>

```

Posledným elementom je *item*, ktorý obsahuje informácie o novo publikovanej položke a je podjednotkou elementu *Channel*. Pre tento element je typické, že sa v dokumente vyskytuje viac krát, pre každú publikáciu nová položka. Do verzie 0.92 bol element *title*, ktorý obsahuje názov danej publikácie, povinný, ale neskôr sa stal voliteľným. Vo verzii 0.91 bol predstavený nepovinný element *description*, ktorý obsahoval stručný popis danej publikácie. Vo verzii 0.92, kde boli tieto dva elementy nepovinné, dochádzalo k problémom s identifikáciou jednotlivých publikácií, tak vo verzii 2.0 bola špecifikovaná podmienka, že tieto elementy sú nepovinné, ale vždy musí byť zadaný aspoň jeden z nich. Do verzie 0.92 bol pridaný ďalší element, nazývaný *link*, povinný, ale s príchodom tejto verzie sa stal nepovinným. To platí aj pre verziu 2.0. V špecifikácii verzie 2.0 boli pridané ďalšie voliteľné elementy ako *author*, ktorý obsahoval meno autora, *comments*, ktorý obsahoval stručné komentáre a posledným bol *pubDate*, ktorý špecifikoval dátum a čas publikovania článku. Formát súčasného elementu *title* môže vyzeráť takto:

```

<item>
  <title>Samsung začal dodávať 1.5 TB disk</title>
  <link>http://www.dsl.sk/article.php?article=7150</link>
  <description>Samsung začal dodávať 1.5 TB disk</description>
  <pubDate>Mon, 09 Mar 2009 08:00:00 GMT</pubDate>
</item>

```

Hlavné spomínané elementy hlavička súboru, *Channel* a *item* sú vo všetkých verziách RSS formátu povinné a majú nasledujúcu štruktúru:

```

<?xml version="1.0" encoding="windows-1250"?>
<rss version="2.0">
  <channel>
    <title>DSL.sk</title>
    <link>http://www.dsl.sk/</link>
    <description>Digitálny Svet pod Lupou.</description>
    <language>sk</language>
    <pubDate>Tue, 10 Mar 2009 11:05:00 GMT</pubDate>
    <item>
      <title>Samsung začal dodávať 1.5 TB disk</title>
      <link>http://www.dsl.sk/article.php?article=7150</link>
      <description>Samsung začal dodávať 1.5 TB disk</description>
      <pubDate>Mon, 09 Mar 2009 08:00:00 GMT</pubDate>
    </item>
  </channel>
</rss>

```

Toto je príklad správne naformátovaného XML dokumentu písaného formátom RSS verzie 2.0, ktorý by bola užívateľská aplikácia schopná analyzovať a upozorniť užívateľa na jednu novú publikáciu, poskytnúť základné informácie o danom RSS kanále a odkaz, kde si môže prečítať daný článok.

2.4 Atom

Začiatkom 21. storočia mal *RSS* formát dominantné postavenie medzi metódami pre sledovanie zmien webového obsahu. Skupina užívateľov, ktorá chcela prispieť a vylepšiť tento formát, narazila na problém, že tento formát verzie 2.0 bol zmrazený *Harvardskou univerzitou* a prehlásili, že nemôže dôjsť k zmenám tohto formátu a ďalší vývoj musí byť vedený pod iným menom. V roku 2003 *Sam Ruby* rozpútal diskusiu medzi komunitou ľudí, ktorí boli ochotní sa zapojiť do vývoja nového formátu a nakoniec sformulovali ciele, ktoré by mal spĺňať nový formát. Medzi tieto ciele patrili hlavne otvorenosť, možnosť implementácie a rozšíriteľnosti hocikým, jasne a presne špecifikovateľný.

Onedlho vydali verziu 0.3, ktorá sa začala šíriť najmä vďaka spoločnosti *Google*, ktorá využila tento formát vo svojich aplikáciách. Po ďalšom vývoji bola vydaná verzia 1.0, ktorá je aktuálna dodnes. Táto verzia bola schválená ako štandard organizáciou *IETF (Internet Engineering Task Force)*. Rozdiely *Atomu* od *RSS* môžeme zhrnúť do hlavných štyroch bodov.

- Najhlavnejším rozdielom je obsahový model. Formát *RSS* obsahuje čistý text alebo odkaz na *HTML* dokument, ale bez možnosti zistiť, o ktorú variantu sa jedná. Na druhej strane *Atom* ponúka možnosť explicitne označiť o aký druh informácie sa jedná.
- Rozdielny formát zápisu dátumov.
- Možnosť špecifikovať jazyk pre každú publikáciu a nie len jednotný jazyk pre celý kanál.
- Väčšia modularita.

2.5 OPML

OPML (Outline Processor Markup Language) je *XML* formát pre výmenu súvisiacich informácií medzi aplikáciami tak, aby štruktúra týchto dát reprezentovala ich vzťahy a nebol závislý na cieľovom operačnom systéme. Špecifikácia štandardu popisuje hierarchickú skladbu elementov v podobe stromovej štruktúry, obsahujúcej distribuované informácie. V súvislosti s *RSS* sa tento štandard využíva ako prostriedok pre publikáciu zoznamu *RSS* kanálov rozdelených do skupín podľa zamerania alebo iných kritérií. Takýto súbor môže užívateľ importovať do *RSS* čítačky, čím sa prihlási k odberu daných kanálov.

Štruktúra dokumentu sa skladá z štyroch hlavných elementov.

- Element *opml* ohraničuje rozsah dokumentu a špecifikuje verziu použitého štandardu. Obsahuje dve podjednotky *head* a *body*.
- Jednotka *head* obsahuje metadáta popisujúce dokument, môže obsahovať voliteľné podjednotky ako napríklad názov dokumentu, dátum vytvorenia, dátum poslednej modifikácie a tak ďalej.

- Element *body* obsahuje publikované dáta v podobe podjednotiek *outline*, ktorých počet nie je štandardom limitovaný.
- Jednotka *outline* môže obsahovať ľubovoľný počet atribútov, ktoré popisujú distribuované dáta. V súvislosti s *RSS* sú typické atribúty *text* a *xmlUrl*. Tento element môže obsahovať ďalšie podjednotky typu *outline*.

Hlavným cieľom tohto formátu je poskytnúť prehľadný, rozširiteľný dokument, ktorý je schopný reprezentovať rôzne druhy dát tak, aby si zachoval jednotnú štruktúru a nebol závislý na cieľovom operačnom systéme.

3 Dostupné riešenia

3.1 Úvod

RSS čítačka je užívateľská aplikácia alebo webové rozhranie programu, zamerané na čítanie informácií publikovaných pomocou formátu RSS na rôznych internetových portáloch. Hlavnou úlohou čítačiek je znížiť čas a úsilie, ktoré musí užívateľ vynaložiť pravidelným kontrolovaním webového portálu. Tieto aplikácie pravidelne kontrolujú RSS kanály, ku ktorým sa užívateľ prihlásil a v prípade, že aplikácia zistí nové publikácie, upozorní užívateľa. Možnosti prihlasovania k RSS kanálom a spôsoby notifikácie koncového užívateľa sa líšia pre každý program. V súčasnosti je k dispozícii veľké množstvo RSS čítačiek, každá sa snaží byť unikátna a poskytovať užívateľom funkcie, ktoré iné riešenia neobsahujú.

Podľa umiestnenia cieľovej aplikácie môžeme rozdeliť RSS čítačky do troch skupín:

- **Webové aplikácie** – pre tento druh aplikácii je typické, že sú umiestnené na internetových stránkach a sú určené pre osobné použitie. Hlavné výhody plynú z toho, že táto aplikácia je poskytovaná určitou internetovou stránkou, na ktorej sa užívateľ zaregistruje a všetky zapísané RSS kanály sú uložené na vzdialenom počítači. To znamená, že nepotrebuje mať u seba nainštalovanú aplikáciu, ale môže si skontrolovať nové publikácie z rôznych počítačov bez toho, aby musel modifikovať stávajúce nastavenia. Jedinou podmienkou je pripojenie k internetu.
- **Desktopové aplikácie** – tieto programy sú špecifické tým, že vyžadujú mať lokálne nainštalovanú aplikáciu, ktorá poskytuje grafické rozhranie, cez ktoré sa užívateľ môže prihlásiť k odberu publikácii pre daný RSS kanál. Typicky toto grafické rozhranie je kvalitnejšie a viac interaktívne ako u webových aplikácii. Výhodou oproti webovým aplikáciám je možnosť pracovať v pozadí bez interakcie s užívateľom a automaticky, pravidelne kontrolovať nové publikácie. V prípade, že sú zverejnené nové dáta, môže rôznymi spôsobmi upozorniť užívateľa o danej skutočnosti a ten si môže vybrať, čo chce vykonať.
- **Rozšírenia stávajúcich desktopových aplikácii** – množstvo desktopových aplikácii umožňuje užívateľovi rozšírenie pomocou nových knižníc alebo zásuvných modulov. Výhodou oproti užívateľským aplikáciám je, že nemusíme inštalovať nové programy, ale stačí rozšíriť stávajúce riešenia o nové doplnky. Správanie takýchto rozšírení sa potom zhoduje s riešením pomocou desktopovej aplikácie.

Ďalej si uvedieme pre každú cieľovú skupinu aplikácie príklady takýchto riešení. Popíšeme si hlavné výhody a nevýhody z hľadiska rozšíriteľnosti, jednoduchosti aktualizácie, prepracovanosti grafického rozhrania, prívetivosť programov k užívateľom, možnosti prevodu na mobilné zariadenia a možnosti prehliadania publikácii v off-line režime.

3.2 Spolupráca webových a desktopových aplikácii v rámci off-line režimu

Skôr ako si popíšeme jednotlivé riešenia RSS čítačiek, vysvetlíme si, aké sú ich možnosti použitia bez aktívneho internetového pripojenia z obecného hľadiska. Jedná sa hlavne o webové aplikácie, ktoré off-line režim nepodporujú vôbec alebo nie je na dostačujúcej úrovni. Desktopové programy majú výhodu v tom, že program samotný je nainštalovaný na cieľovom počítači a môže si ukladať svoje dáta na pevný disk.

Najlepšie riešenie off-line režimu pre webovú aplikáciu je, ak daný internetový portál poskytuje podporný program, ktorý je nainštalovaný na cieľovom počítači. Následne komunikuje s webovou aplikáciou a ukladá potrebné dáta tak, aby nebolo potrebné internetové pripojenie počas prezerania aktualizovaných RSS kanálov. Ďalej môže takýto program zaznamenávať články, ktoré si užívateľ pozrel alebo označil ako dôležité a keď bude aktívne internetové pripojenie tak sa zosynchronizuje s webovým programom. Na takomto koncepte pracuje RSS čítačka Google Reader, ktorá pre funkčný off-line režim vyžaduje aplikáciu Google Gears.

Ďalšie veľmi často používané riešenie je využitie *cache* internetového prehliadača. *Cache* je vyrovnávacia pamäť, ktorá si ukladá navštívené webové stránky na pevný disk. Následne je možné takéto stránky prehliadať aj bez aktívneho internetového pripojenia. Toto je komplexné riešenie, ktoré platí pre všetky webové aplikácie.

Menej používané je riešenie v podobe využitia internetových sťahovačov, ktoré poskytujú rôzne spôsoby realizácie off-line režimu. Typické pre webovú aplikáciu je, že obsahuje zoznam zapísaných RSS kanálov a zoznam všetkých odkazov na publikované články. Internetový sťahovač použije takúto stránku ako zdroj dát pre sťahovanie. Nastavíme hĺbku sťahovania, do ktorej sa bude zanorovať, typicky je to minimálne 1 a následne získame kópiu dát, ktorú môžeme prehliadať bez internetového pripojenia. Ďalším možným riešením je, ak webová aplikácia umožňuje export zoznamu adres všetkých článkov, ktorý použijeme ako parameter pre internetový sťahovač.

Desktopové aplikácie majú off-line režim vyriešený typicky pomocou lokálnej kópie RSS kanálu, ktorá obsahuje zoznam internetových odkazov na publikované články. Buď priamo obsahujú v sebe internetový sťahovač, ktorý je schopný vytvoriť lokálnu kópiu odkazovaných stránok, alebo umožňujú export zoznamu článkov.

3.3 Webové aplikácie

3.3.1 Google Reader

Google Reader (<https://www.google.com/reader>) patrí medzi najznámejšie RSS čítačky, ktoré nepotrebujú lokálnu inštaláciu klienta. Potrebná je len bezplatná registrácia, ktorá vytvorí užívateľský účet pre prístup k uloženým dátam.

Charakteristické vlastnosti sú:

- Jednoduchosť grafického rozhrania neuberá na funkčnosti. Umožňuje zvoliť rôzne druhy zobrazovania.
- Import, export zapísaných RSS kanálov vo formáte *OPML*.
- Možnosť používania klávesových skratiek.
- Po nainštalovaní rozšírenia je možné využívať jednoduchý off-line mód.
- Nakoľko nepotrebuje klienta na cieľovom počítači, je prenositeľnosť na mobilné zariadenia bezproblémová. Dokonca je vyriešená aj podpora pre zariadenia, ktoré nemajú dostatočné vybavenie pre zobrazovanie dynamických internetových stránok, na ktorých sa zobrazuje vo WAP podobe.

Pre túto RSS čítačku môžem vyzdvihnúť hlavne možnosť importu a exportu RSS kanálov, jednoduchosť a priateľnosť užívateľského rozhrania, prijateľnú rýchlosť aktualizácií. Jediná nevýhoda je off-line mód, pretože nedokáže stiahnuť celé články, ktoré sú odkazované v RSS súbore. Sťahuje len zoznam dostupných článkov a ich stručný popis, ak tento popis obsahuje odkazy na nejaké obrázky alebo iné dokumenty, tak v off-line móde sa nebudú zobrazovať.

3.3.2 Windows Live

Windows Live (<http://my.live.com/>) je celá séria webových aplikácií, ktorých súčasťou je aj RSS čítačka. Pre prístup k tejto internetovej aplikácii nepotrebujeme ani registráciu.

Charakteristické vlastnosti sú:

- Prepracované dynamické grafické rozhranie, s veľkým množstvom nastavení zobrazenia.
- Import a export zapísaných RSS kanálov pomocou formátu *OPML*.
- Možnosť vyhľadávania RSS kanálov a ich následný odber.

Vyzdvihnúť môžem hlavne veľmi dobre spracované grafické rozhranie. Možnosť meniť farby celého rozhrania, používať rôzne druhy zobrazovania RSS kanálov, vytvárať si vlastné stránky v rámci rozhrania a sprehľadniť si zoskupovanie zapísaných RSS kanálov. Jednoduché pridávanie nových RSS kanálov pomocou vyhľadávania. Možnosť prevodu na mobilné zariadenia je tiež riešená

pomocou optimalizovaných stránok. Hlavnou nevýhodou je absencia off-line módu a o niečo pomalšia rýchlosť aktualizácie.

3.3.3 Tiny Tiny RSS

Tiny Tiny RSS (<http://tt-rss.org/>) je na rozdiel od predchádzajúcich spomínaných RSS čítačiek otvorené riešenie, ktoré je možné umiestniť na vlastné internetové stránky. Pre prístup nie je potrebná registrácia a je možné hneď začať pracovať s aplikáciou.

Charakteristické vlastnosti sú:

- Používanie klávesových skratiek.
- Možnosť zmeny vzhľadu celého grafického rozhrania.
- Otvorený zdrojový kód pod licenciou GPL.

Ako hlavnú výhodu môžem považovať otvorenosť zdrojového kódu a možnosť upravenia výslednej aplikácie presne na mieru. Vzhľadom nato aplikácia platí veľkú daň oproti komerčným riešeniam, lebo vývoj je pomalý a nie je toľko kreatívny, neprináša veľa nových funkcií. Medzi nevýhody patrí hlavne absencia importu a exportu RSS kanálov, off-line módu a možnosti nastavení pre cieľových užívateľov. Možnosť prevodu na mobilné zariadenia nie je riešená. Zariadenia, ktoré podporujú zobrazovanie klasických webových stránok, by nemali mať problém so zobrazením.

3.4 Desktopové aplikácie

3.4.1 Mozilla Thunderbird

Hlavné zameranie tejto aplikácie s otvoreným kódom je emailový klient, ale obsahuje aj kvalitnú RSS čítačku. Vzhľadom nato, že čítanie RSS kanálov nie je hlavnou náplňou tohto programu, tak tu nájdeme veľké množstvo nastavení a jednoduché prepracované užívateľské rozhranie.

Hlavné charakteristiky sú:

- Otvorenosť a podpora viacerých operačných systémov.
- Veľké množstvo nastavení a funkcií.
- Podpora automatického kontrolovania aktualizácií a prípadného upozornenia na nový obsah.
- Jednoduchý off-line mód.

Hlavnou výhodou tejto aplikácie môžem označiť otvorenosť a prenositeľnosť. Ďalej musím spomenúť, že aplikácia nie je zameraná len na čítanie RSS kanálov, ale aj veľa ďalších protokolov a z toho vyplýva univerzálnosť. Rýchlosť aktualizácie je veľmi dobrá a množstvo nastavení, či už automatické aktualizovanie po nastavenom časovom intervale alebo pre každý RSS kanál zvlášť nastaviť čas aktualizácie. Program je možné minimalizovať na lištu, kde beží v pozadí a v prípade

nových udalostí upozorní užívateľa. Medzi nevýhody patrí veľmi jednoduchý off-line mód, ktorý sťahuje len nadpisy a stručný popis publikácii a takisto ako v prípade Google Reader odkazy v rámci popisov sú v off-line móde nedostupné. V prípade prevodu na mobilné zariadenia je podpora veľmi slabá, jedine zariadenia s Windows Mobile sú schopné spustiť túto aplikáciu.

3.4.2 RSSOwl

Aplikácia je zameraná na čítanie RSS kanálov a poskytuje bohaté možnosti exportu dát a nastavení. Je dodávaná s veľkým počtom dopredu zapísaných RSS kanálov a samozrejme je možné pridávať ďalšie nové.

Charakteristickými vlastnosťami sú:

- Rozšíriteľnosť a veľké možnosti nastavenia.
- Import a export RSS kanálov a takisto aj export publikácii do rôznych formátov.
- Jednoduché a zároveň prepracované užívateľské rozhranie.
- Otvorenosť a podpora viacerých operačných systémov.

Hlavnými kladnými prvkami tejto aplikácie sú veľké možnosti nastavení spracovania jednotlivých RSS kanálov a možnosti upozornenia na jednotlivé nové udalosti. Jednoduché grafické rozhranie, ktoré poskytuje veľké možnosti nastavení a spôsobov usporiadania jednotlivých kanálov a publikácii. Za záporné vlastnosti môžem opäť považovať off-line mód, ktorý funguje na princípe ako predchádzajúce riešenia a možnosti prevodu na mobilné zariadenia neobsahuje žiadne.

3.4.3 FeedDemon

Program je bezplatný a zameraný na prezeranie RSS kanálov a umožňuje užívateľom základné nastavenia. Existuje aj komerčná verzia, ktorá obsahuje pokročilé nastavenia. Užívateľské rozhranie je dosť jednoduché a vzhľadovo vyzerá dosť zastaralo, napriek tomu obsahuje všetky základné funkcie, ktoré bežný užívateľ potrebuje.

Charakteristické vlastnosti sú:

- Import a export uložených RSS kanálov.
- Jednoduchý off-line mód s možnosťou výberu miesta uloženia.
- Len základné nastavenia a možnosti spracovania.

Medzi výhody tejto aplikácie môžem zaradiť jednoduchosť a rýchlosť užívateľského rozhrania. Na druhej strane užívateľské rozhranie neosloví užívateľa na prvý pohľad, lebo neobsahuje animované prvky ako konkurenčné produkty. Aplikácia funguje len pod operačným systémom Windows a prenositeľnosť na mobilné zariadenia nie je doriešená. Off-line mód je podobný konkurenčným riešeniam, ale obsahuje možnosť výberu umiestnenia uložených dát.

3.5 Rozšírenia desktopových aplikácií

3.5.1 Feedview

Patrí k rozšíreniu internetového prehliadača Mozilla FireFox. Momentálne bol vývoj tohto rozšírenia zastavený z dôvodu pridania podpory čítania *RSS* kanálov v rámci internetového prehliadača. Patrí medzi prvé a klasické rozšírenia, ktoré umožňovali sledovanie *RSS* kanálov v podobe záložiek. Každý kanál mal svoju záložku, ktorá obsahovala informácie o zverejnených publikáciách v podobe odkazov na dané téma.

Charakteristické vlastnosti sú:

- Otvorenosť a rozšíriteľnosť.
- Jednoduché nastavenia a pridávanie nových *RSS* kanálov.
- Rýchlosť aktualizácie.

Medzi hlavné výhody patrí rýchlosť aktualizácie, nakoľko sa sťahujú len nadpisy publikácií a adresy odkazov. Ďalej je to integrácia do internetového prehliadača, kde stačí si daný *RSS* kanál pridať do záložiek a automaticky bude spracovávaný. Ako nevýhody môžeme považovať absenciu pokročilých nastavení alebo off-line mód. Prenositeľnosť na mobilné zariadenia záleží hlavne na prenositeľnosti internetového prehliadača.

3.5.2 News Fox

Radí sa tiež k rozšíreniam aplikácie Mozilla FireFox, ale na rozdiel od predchádzajúceho riešenia sa stále vyvíja a obsahuje aj vlastné užívateľské rozhranie. Umožňuje užívateľom základné možnosti nastavení a je zobrazovaný ako internetová stránka.

Medzi charakteristické vlastnosti patrí:

- Vlastné užívateľské rozhranie v podobe internetovej stránky s množstvom nastavení.
- Integrácia so záložkami, ktoré obsahujú *RSS* kanály.
- Rýchlosť aktualizácie
- Možnosti upozornenia na nové neprečítané publikácie.

Ako hlavné výhody môžeme spomenúť dobrú integráciu v rámci internetového prehliadača a rýchlosť aktualizácie. Jednoduché a užívateľsky príjemné menu umožňuje základné nastavenia, ktoré budú dostačujúce pre nenáročných užívateľov. Z negatívnej stránky to je opäť prenositeľnosť, ktorá závisí na internetovom prehliadači a jednoduchý off-line mód v podobe stručného popisu publikácie.

3.5.3 Sage

Opäť sa jedná o rozšírenie aplikácie Mozilla FireFox, ktoré je integrované v podobe postranného panelu, ktorý obsahuje základné a najpoužívanejšie možnosti. Prehľad RSS kanálu je zobrazovaný v podobe internetovej stránky ale tieto informácie môžeme nájsť aj na postrannom paneli.

Charakteristické vlastnosti aplikácie sú:

- Integrácia v podobe postranného panelu.
- Rýchlosť aktualizácie.
- Základné možnosti nastavení.

Medzi pozitívne vlastnosti tohto rozšírenia môžem zaradiť jednoduché a užívateľsky príjemné grafické rozhranie v podobe postranného panelu a rýchlosť aktualizácie. Ako negatíva môžem spomenúť prenositeľnosť, ktorá závisí na internetovom prehliadači, veľmi malé možnosti nastavení a absencia off-line módu.

4 Návrh aplikácie

Prvý návrh aplikácie bol veľmi jednoduchý, nakoľko som si popozeral stávajúce riešenia RSS čítačiek a poznal som ich výhody a nevýhody. Uvažoval som nad dokončením off-line režimu pre existujúcu RSS čítačku, ale nenašiel som vyhovujúcu aplikáciu, v ktorej by som mohol pokračovať. Hlavným problémom bola uzavretosť zdrojového kódu alebo zložitosť samotného riešenia programu. Nakoniec som sa rozhodol vytvoriť si vlastnú aplikáciu, ktorá by obsahovala všetky zaujímavé funkcie stávajúcich riešení a obsahovala prepracovaný off-line režim s veľkým počtom nastavení.

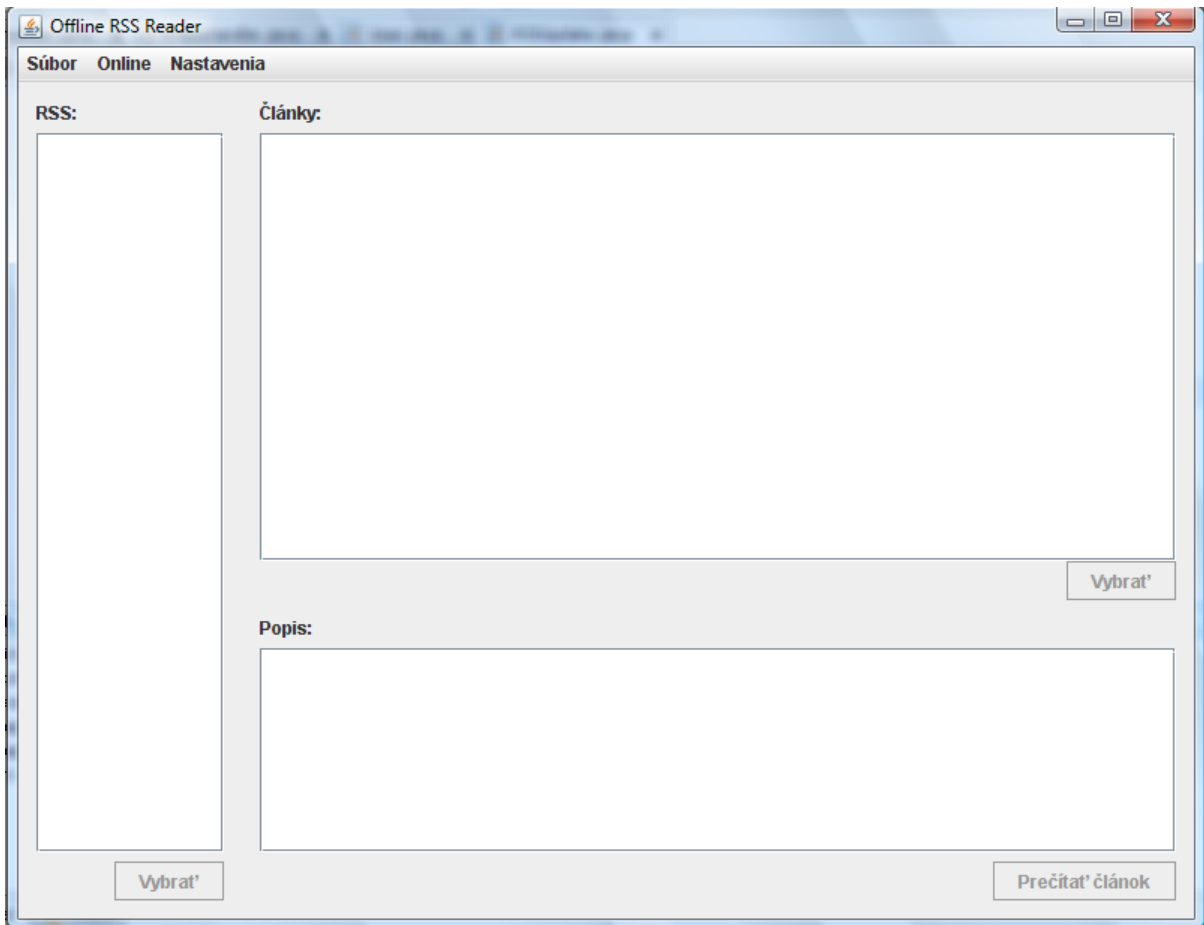
Úmyslom bolo vytvoriť jednoduchý užívateľsky prívetivý program, ktorý umožňuje prácu bez aktívneho internetového pripojenia. Otvorenosť zdrojového kódu bola samozrejmosťou a posledná otázka bola, aký zvoliť programovací jazyk pre riešenie výslednej aplikácie. Nakoniec som sa rozhodol pre *Javu* a to z niekoľkých dôvodov. Jedná sa o programovací jazyk, pre ktorý je typický objektovo orientovaný prístup, ktorý v spojení so správnou implementáciou zaručuje jednoduchú údržbu a modifikovateľnosť zdrojových kódov. Ďalšou veľmi dôležitou vlastnosťou tohto jazyka je prenositeľnosť. Či už sa jedná o rôzne operačné systémy alebo zariadenia. V rámci mojej aplikácie som predpokladal s použitím systému *Windows* a *Linux*, ale rozšírenie podpory pre ďalšie by nebol problém.

V tomto bode návrhu som vedel, čo a v akom programovacom jazyku chcem vytvoriť. Nasledovalo hľadanie najvhodnejšieho spôsobu realizácie výslednej aplikácie. Ako prvý významný problém môžem považovať spôsob uloženia dát na pevnom disku, ktorý bol spojený s off-line režimom. Počet ukladaných súborov vo výslednej aplikácii bude obrovský a môže dochádzať ku kolízii mien a umiestnenia. Problém konfliktných názvov som vyriešil použitím hashovacej funkcie, ktorá preloží *URL* adresu ukladaného súboru na číslo. Pre každý RSS kanál bude vytvorený vlastný priečinok, ktorý bude obsahovať sťahované dáta, čím sa vyriešil problém s umiestnením.

Následne bol vypracovaný prvý návrh aplikácie, ktorý zahŕňal koncept grafického rozhrania. Jednalo sa o jednoduché rozdelenie obrazovky na jednotlivé časti, ktoré budú obsahovať informácie o RSS kanáloch a publikovaných článkoch. Súčasťou bol aj logovací systém, ktorý zaznamenával všetky vykonané operácie a generoval prípadné chybové správy. Veľmi sa osvedčil vo fáze ladenia programu. Posledným významným prvkom bol integrovaný prehliadač *HTML* stránok, ktorý nie je obsiahnutý vo výslednej aplikácii, kde bol nahradený použitím predvoleného internetového prehliadača na cieľovom počítači.

Prvý návrh neobsahoval najpodstatnejšie časti aplikácie. Chýbalo navrhnuť modul RSS čítačky a spôsob spracovania RSS kanálu pre potreby prezerania publikovaných článkov bez aktívneho internetového pripojenia. Jednoduchšou časťou bolo vytvorenie RSS čítačky, ktorá spracuje *XML* súbor napísaný podľa štandardu RSS. Výstupom bude zoznam všetkých publikovaných článkov

pre daný kanál. Poslednou časťou návrhu bol modul pre off-line režim, ktorého cieľom bolo zo zoznamu *URL* adres vytvoriť na pevnom disku lokálnu kópiu odkazovaných stránok. Myšlienka veľmi jednoduchá, ale v konečnom dôsledku sa jednalo o najťažšiu časť celej aplikácie.



Obrázok 1 Prvý návrh grafického rozhrania

5 Implementácia

V tejto kapitole si popíšeme objektový model výslednej aplikácie, detailnejšie rozoberieme niektoré zaujímavé implementované časti, objasníme spôsob testovania a overovania správnej funkčnosti programu. Ako programovací jazyk som zvolil *Javu* vo verzii 1.6.0 a pre programovanie som používal vývojové prostredie *NetBeans*.

5.1 Objektový model

Z vypracovaného návrhu vznikol objektový model aplikácie, ktorý po menších úpravách získal finálnu podobu. Výsledná štruktúra aplikácie pozostáva z nasledujúcich balíčkov:

- *HTML* – zaoberá sa sťahovaním jednotlivých webových stránok s rôznymi nastaveniami, veľmi dôležitá časť pre off-line režim
- *Log* – logovací systém aplikácie
- *Main* – obsahuje metódy spúšťané po štarte a hlavného okna programu
- *RSS* – modul pre prácu s *RSS* kanálmi na abstraktnej úrovni
- *Settings* – pracuje s nastaveniami aplikácie
- *XML* – balík pre prácu s *XML* súbormi

Každý balík obsahuje jednotlivé triedy, ktoré implementujú požadované metódy. Detailnejšie si popíšeme časti, ktoré sa podieľajú na spracovaní *RSS* kanálov.

5.2 Čítačka XML súborov

Balík *XML* obsahuje triedy, ktoré sa zaoberajú spracovaním a vytváraním *XML* súborov. Pre uľahčenie práce som využil knižnicu *dom4j* (www.dom4j.org). S touto knižnicou som mohol pracovať s *XML* súbormi na abstraktnej úrovni, čo veľmi urýchlilo implementáciu. Jednotlivé triedy implementujú metódy pre prácu s jedným typom *XML* súborov.

Trieda *Settings* obsahuje metódy, ktoré spracovávajú konfiguračný súbor aplikácie. Triedy *ImportRSS* a *ExportRSS* sa používajú pre spracovanie importovaných a vytváranie exportovaných súborov vo formáte *OPML*. *XML* súbor vytvorený triedou *RSSList* reprezentuje zoznam všetkých zapísaných *RSS* kanálov a pre každý obsahuje dáta využívané aplikáciou. Trieda *RSSHeader* generuje pre každý kanál *XML* dokument, ktorý obsahuje informácie o publikovaných článkoch. Poslednou, ale najdôležitejšia triedou je *RSSReader*, ktorá reprezentuje čítačku *RSS* súborov. Kontroluje dáta, či zodpovedajú podporovanému štandardu. Získava zoznam nových článkov, ktoré boli publikované daným kanálom.

5.3 RSS modul

Balík *RSS* obsahuje logiku celého spracovania jednotlivých kanálov. Obsahuje triedu *RSSUpdate*, ktorá koordinuje celú aktualizáciu. Pozostáva zo štyroch častí, ktoré na seba nadväzujú, ale tento priebeh môže byť prerušený užívateľom.

Aktualizácia začína kontrolou daného *RSS* kanálu, či obsahuje nejaké nové dáta. Porovná sa dátum poslednej modifikácie s dátumom poslednej aktualizácie a ak sa líšia tak môžeme pokračovať ďalej. Element obsahujúci informácie o poslednej úprave *RSS* súboru je nepovinný a v prípade, že nebol špecifikovaný, tak budeme predpokladať, že bol daný kanál modifikovaný.

V ďalšom kroku sa prevedie prevod dát publikovaných *RSS* kanálom na vnútornú reprezentáciu aplikácie a uložia sa informácie o jednotlivých článkoch. V tomto bode aktualizácie už je možné prezerat' publikované dáta.

Nasleduje nepovinný krok aktualizácie, ktorý sťahuje webové stránky špecifikované v *RSS* dokumente. Nastaviť, alebo úplne vypnúť túto časť je možné v nastaveniach, ktoré sú prístupné z hlavného okna aplikácie.

Poslednou časťou aktualizácie je vytvorenie súboru, ktorý reprezentuje všetky články publikované daným kanálom v podobe webovej stránky. Obsahuje odkazy na dáta jednotlivých dokumentov v on-line podobe a v prípade sťahovania lokálnej kópie aj v off-line podobe.

Druhou triedou, ktorú tento balík obsahuje, je *RSSHandler*, ktorý slúži pre abstrakciu práce s jednotlivými *RSS* kanálmi a obsahuje všetky potrebné metódy využívané aplikáciou.

5.4 Off-line sťahovač

Hlavnou úlohou balíku *HTML*, je vytvoriť lokálnu kópiu webovej stránky, tak aby si zachovala svoj vzhľad a obsah. Pre prácu s *HTML* kódom som použil knižnicu *Jericho HTML Parser* (<http://jericho.htmlparser.net/>), ktorá zjednodušuje prehľadávanie jednotlivých tagov a umožňuje rozpoznávanie kódovania webovej stránky. Základný princíp je stiahnuť stránku, ktorá je predávaná ako parameter, a podľa nastavení jej jednotlivé elementy.

Trieda *HTMLHandler* reprezentuje rozhranie pre navrhnutý sťahovač. *RSS* modul postupne využíva poskytované metódy a získava všetky potrebné dáta. Najzaujímavejšou je časť pre sťahovanie dát pre jednotlivé dokumenty publikované *RSS* kanálom. Ako prvé si získa zoznam všetkých článkov a vyberie tie, ktoré neobsahujú lokálnu kópiu dát. Postupne pre každý začne preberanie za pomoci triedy *HTMLContent*. Následne označí článok za úspešne uložený a sprístupní jeho obsah z hlavného okna aplikácie.

Sťahovač samotný je reprezentovaný triedou *HTMLContent*, ktorá obsahuje jednotlivé metódy pre získavanie internetových objektov. Obsahuje dva spôsoby pre preberanie jednotlivých

dokumentov. Jednoduchší prístup využíva bytový prenos a priamo ukladá získavané dáta na pevný disk. Tento spôsob sa využíva na sťahovanie všetkých súborov, ktoré nevyžadujú analýzu jednotlivých tagov. Zložitejší prístup postupne prechádza sťahovaný objekt a podľa jednotlivých nastavení sťahuje požadované elementy. Ako prvé sa stiahne samotný text *HTML* dokumentu a dochádza k analýze jeho kódovania. Následne ak sa odkazuje na externý súbor s *CSS* štýlom, tak dôjde k prepisu jeho obsahu do hlavičky. Ďalej sa vytvorí zoznam všetkých obrázkov, ktoré daný dokument obsahuje a postupne sa budú sťahovať pomocou bytového prenosu. Následne sa vytvorí zoznam všetkých odkazovaných dokumentov a dochádza k ich rozdeleniu podľa prípon. Opäť dôjde k sťahovaniu jednotlivých súborov. Pre každý sťahovaný objekt sa v originálnom *HTML* dokumente zmení cesta, ktorá odkazuje na lokálnu kópiu požadovaného objektu. Ako posledné sa uloží takto modifikovaný zdrojový kód stránky na pevný disk a tým sa ukončí tento proces.

5.5 Testovanie

V tejto kapitole popíšem významné testy, ktoré som aplikoval počas vývoja aplikácie a ich výsledky. Snažil som sa odladiť program tak, aby bol schopný včas identifikovať nesprávne dáta na vstupe a upozorniť užívateľa chybovou správou, poprípade zaznamenať informácie o vzniknutej chybe do logovacieho systému.

Po vytvorení prvého návrhu aplikácie som testoval hlavne grafické rozhranie a možnosti logovacieho systému. Funkčnosť grafického rozhrania bola na dobrej úrovni, ale ovládanie bolo zbytočne zložité. Takisto ovládanie aplikácie klávesnicou nebolo takmer možné. Logovací systém fungoval podľa očakávaní a s menšími úpravami ostal vo svojej podobe až do finálnej verzie programu. Na základe výsledkov tohto testu som vypracoval druhý návrh grafického rozhrania, ktorý bol užívateľsky intuitívnejší a umožňoval jednoduché ovládanie klávesnicou.

Po dokončení implementácie čítačky *RSS* súborov nasledovalo testovanie v podobe rôznych formátov súborov. Jednalo sa o podporované verzie štandardu *RSS* a ich chybné modifikované verzie. Cieľom bolo zistiť správnosť rozpoznávania jednotlivých publikovaných článkov a identifikovanie nepodporovaných formátov. Výsledok testu potvrdil správnosť implementácie modulu *RSS* čítačky, takisto odhalil možnosť neočakávaného ukončenia aplikácie v prípade chybného vstupu.

Posledný implementovaný modul bol off-line sťahovač, ktorý vyžadoval postupné a dôkladné testovanie, ktoré som rozdelil do troch častí. Prvá fáza testovala rýchlosť sťahovania samotného. Ako druhé som analyzoval správnosť vytvorených dát a poslednou časťou bol záťažový test. Po každom výsledku došlo k ladeniu a opätovnému testovaniu. Rýchlosť sťahovania bol najväčší problém, lebo množstvo prenášaných dát bolo obrovské a práca s nimi na textovej úrovni veľmi pomalá. Nakoniec som tento problém vyriešil pomocou knižnice *Jericho HTML Parser*, ktorá umožňuje rýchle prehľadávanie *HTML* dokumentov. Nasledovalo overenie správnosti sťahovaných dát. V tejto časti

som nenarazil na zásadnejšie problémy, všetky odkazy boli správne upravené a požadované dáta uložené na pevnom disku. Posledná časť bol záťažový test, ktorý mal otestovať sťahovač pri veľkom počte *RSS* kanálov a preberaní všetkých elementov. Pre tento test som využil *OPML* dokument obsahujúci zoznam sto *RSS* kanálov. Aktualizácia trvala 58 minút, na pevný disk sa uložilo 4,1GB dát a 11 zdrojov sa nepodarilo úspešne dokončiť. 8 z dôvodu nepodporovaného formátu, 2 nedodržiavali štandard verzie špecifikovanej v dokumente a 1 sa čiastočne dokončil, ale niektoré publikované články obsahovali neplatné dáta.

6 Záver

Počas práce na tomto projekte som sa zoznámil s rôznymi formami distribúcie článkov na internete. S formátom *RSS* som sa stretol už predtým, ale nepoznal som jeho vývoj, štruktúru, výhody, nevýhody a ďalšie charakteristické vlastnosti. Následne som si naštudoval veľké množstvo internetových zdrojov, ktoré sa zaoberajú touto tematikou a vyskúšal dostupné *RSS* čítačky, ktoré som vzájomne porovnával a hodnotil.

Cieľom tejto práce bolo vytvoriť aplikáciu, ktorá bude umožňovať čítanie článkov distribuovaných pomocou protokolu *RSS* bez aktívneho internetového pripojenia. Ako najdôležitejšiu časť považujem návrh programu, lebo v prípade chybnnej alebo neúplnej špecifikácie požiadaviek, mohlo dôjsť k vytvoreniu nefunkčnej aplikácie. Zdokonalil som sa v objektovom návrhu, ktorý mi určite uľahčil implementáciu a ušetril množstvo času. Výsledný kód aplikácie som sa snažil písať prehľadne, efektívne a využívať rôzne možnosti komentárov. Zaručuje to jednoduchú orientáciu v zdrojových kódoch a následnú modifikáciu.

Napriek tomu, že je súčasné riešenie prakticky kompletne a vyhovuje všetkým požadovaným vlastnostiam, bolo by možné na aplikácii ďalej pracovať. Spomeniem niektoré možnosti vylepšenia programu, nad ktorými som uvažoval, ale nemal som potrebný čas pre ich implementáciu. Rozšírenie podporovaných formátov o *RDF* a *Atom*. Zlepšiť spoluprácu s internetovým prehliadačom, poprípade pokúsiť sa o priamu integráciu. Vytvoriť iné návrhy užívateľského rozhrania a umožniť používanie skinov. Vytvoriť stratégie sťahovania jednotlivých *RSS* kanálov a ich článkov, a tým urýchliť aktualizovanie aplikácie. A mnoho ďalších rozšírení, ktoré obsahujú konkurenčné produkty.

Práca bola pre mňa veľkým prínosom v oblasti programovania a získal som aj bohaté teoretické znalosti z oblasti distribúcie internetových publikácií. Zoznámil som sa s technikou jednotlivých formátov a ich využitím. Taktiež som sa zdokonalil v objektovom návrhu, vyskúšal som si niektoré koncepcie objektového programovania v praxi. Následne som sa naučil implementovať samostatnú aplikáciu na základe špecifikovaných požiadaviek a vytvoriť inštaláciu pre rôzne operačné systémy. Nakoniec som vytvoril programovú dokumentáciu a užívateľský manuál.

Literatúra

- [1] Wikipedia, the free encyclopedia. *Markup Language* [online]. [cit. 2009-04-29].
Dostupný z WWW: < http://en.wikipedia.org/wiki/Markup_language >.
- [2] Wikipedia, the free encyclopedia. *OPML* [online]. [cit. 2009-05-03].
Dostupný z WWW: < <http://en.wikipedia.org/wiki/OPML> >.
- [3] OPML. *OPML Specification* [online]. [cit. 2009-04-28].
Dostupný z WWW: < <http://www.opml.org/spec2>>.
- [4] Wikipedia, the free encyclopedia. *RSS* [online]. [cit. 2009-05-01].
Dostupný z WWW: < <http://en.wikipedia.org/wiki/RSS> >.
- [5] RSS.NET. *RSS Version Comparison* [online]. [cit. 2009-05-02].
Dostupný z WWW: < http://rss-net.sourceforge.net/documents/version_comparison.html >.
- [6] Sun Microsystems. *The Java Tutorials* [online]. [cit. 2009-04-13].
Dostupný z WWW: < <http://java.sun.com/docs/books/tutorial/>>.
- [7] Wikipedia, the free encyclopedia. *XML* [online]. [cit. 2009-04-28].
Dostupný z WWW: < <http://en.wikipedia.org/wiki/XML> >.
- [8] Ayers, D., Watt, A. *Beginning RSS and Atom Programming*, London: Wrox, 2005, ISBN 978-0-7645-7916-5.
- [9] Bradley, N. *XML: Kompletní průvodce*, Praha: Grada Publishing, 2000, ISBN 80-7169-949-7.
- [10] Hammersley, B. *Content Syndication with RSS*, Sebastopol: O'Reilly, 2003, ISBN 0-596-00383-8.
- [11] Herout, P. *Java a XML*, České Budějovice: Kopp, 2007, ISBN 978-80-7232-307-4.
- [12] Herout, P. *Java grafické uživatelské prostředí a čeština*, České Budějovice: Kopp, 2001, ISBN 80-7232-237-0.
- [13] Herout, P. *Učebnice jazyka Java*, České Budějovice: Kopp, 2000, ISBN 80-7232-115-3.
- [14] Pecinovský, R. *Myslíme objektově v jazyku Java 5.0*, Praha: Grada Publishing, 2004, ISBN 80-247-0941-4.

Zoznam príloh

Príloha 1. Uživatelský manuál aplikácie *Offline RSS Reader*

Príloha 2. Programová dokumentácia programu *Offline RSS Reader*

Príloha 3. CD so zdrojovými kódmi a aplikáciou *Offline RSS Reader*

Príloha 4. Plagát prezentujúci prácu

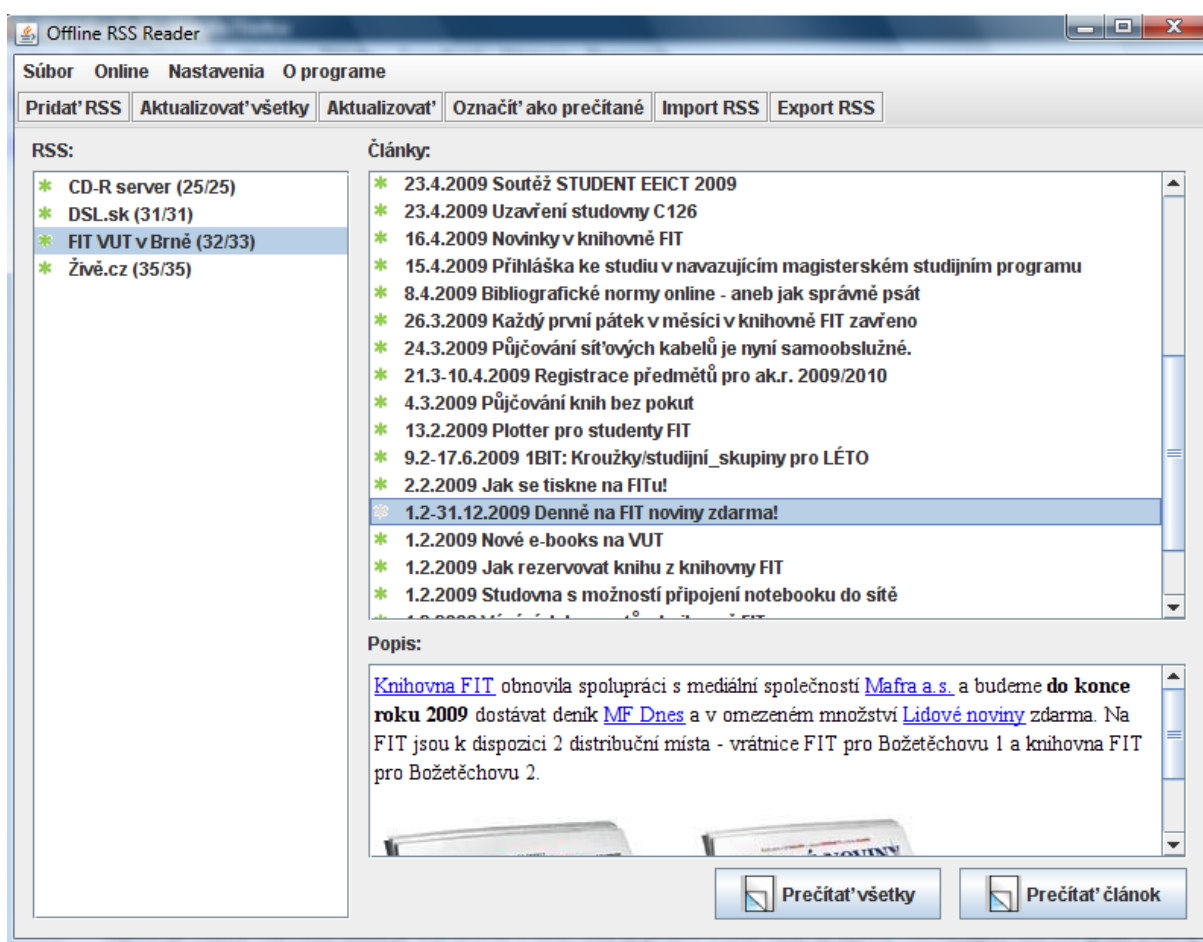
Príloha 1. Užívateľský manuál aplikácie

Offline RSS Reader

Prehľad

Meno *Offline RSS Reader* vzniklo spojením anglických slov, ktoré sú typické pre túto aplikáciu. *Offline* charakterizujúci prepracovaný režim práce bez aktívneho internetového pripojenia a *RSS Reader* vo význame čítačky RSS súborov.

Hlavným cieľom aplikácie je byť užívateľsky prívetivá, stať sa prostriedkom pre každodenné získavanie informácií z internetu a umožňovať užívateľom prácu v off-line režime.



Obrázok 2 Hlavné okno aplikácie

Charakteristické vlastnosti

- Plná podpora RSS formátu vo verzii 2.0.
- Prepracovaný režim sťahovania RSS kanálov, ktorý umožňuje ukladanie publikovaných článkov lokálne na pevný disk.
- Import a export RSS kanálov pomocou štandardu OPML.

Systemové požiadavky

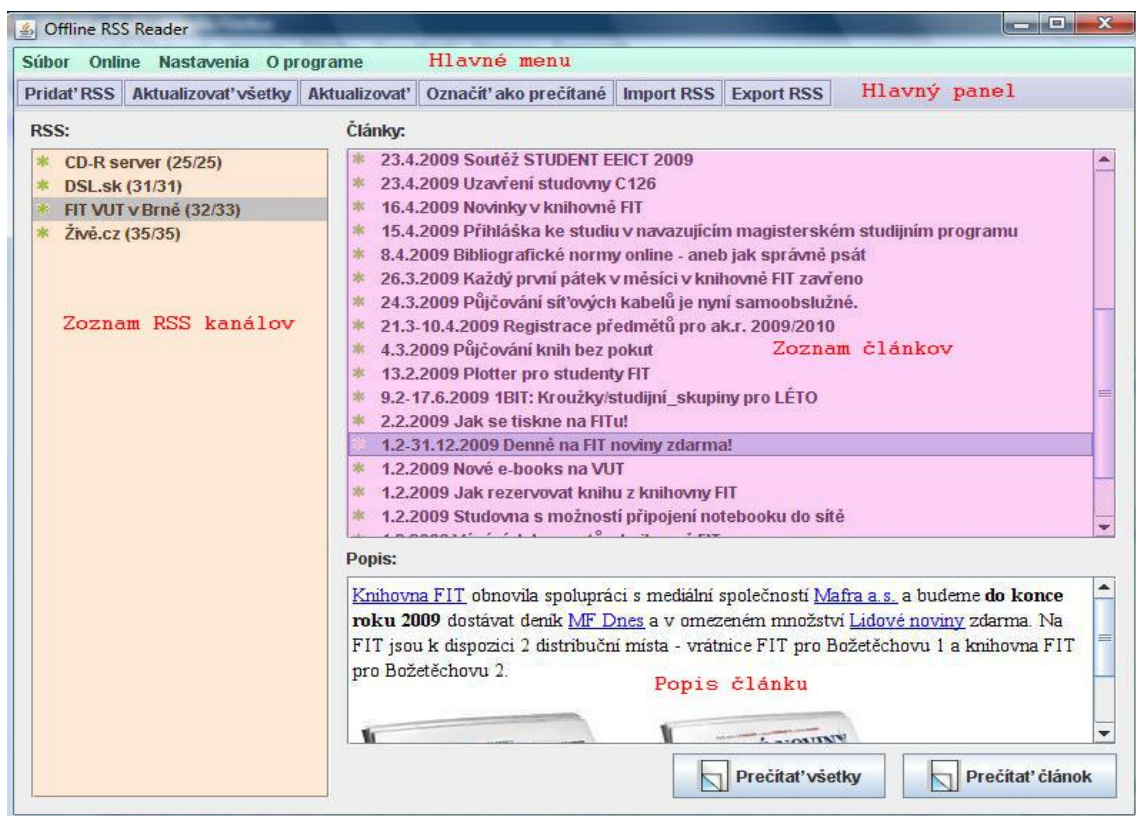
Doporučené požiadavky na cieľový systém:

Operačný systém	Windows, Linux
Processor	minimálne 300MHz
Pamäť RAM	minimálne 128MB
Pevný disk	minimálne 75MB voľného miesta

Popis jednotlivých funkcií

Offline RSS Reader poskytuje veľký počet funkcií, ktoré môžete využiť pri práci s jednotlivými RSS kanálmi. Postupne si ich jednotlivo popíšeme a ukážeme na jednoduchých príkladoch.

Grafické rozhranie aplikácie sa skladá z piatich základných častí, ktoré budú v ďalších častiach textu referované nasledujúcimi názvami:



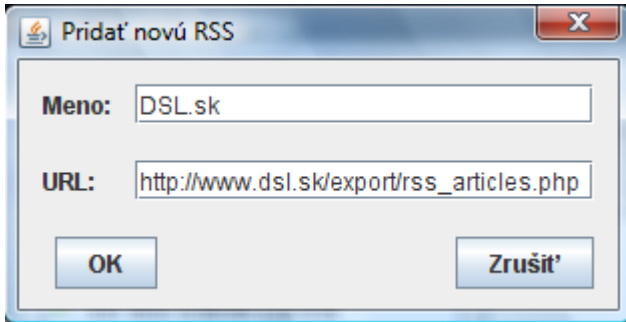
Obrázok 3 Základné časti grafického rozhrania

Prihlásenie k RSS kanálu:

Slúži na prihlásenie k odberu článkov, ktoré sú publikované RSS kanálom.

Inštrukcie:

- Z hlavného menu vyberte ponuku Súbor -> Pridať RSS.
- Zadajte požadované meno RSS kanálu do poľa Meno.
- Zadajte URL RSS kanálu do poľa URL.
- Potvrďte tlačidlom OK.



Obrázok 4 Pridávanie novej RSS

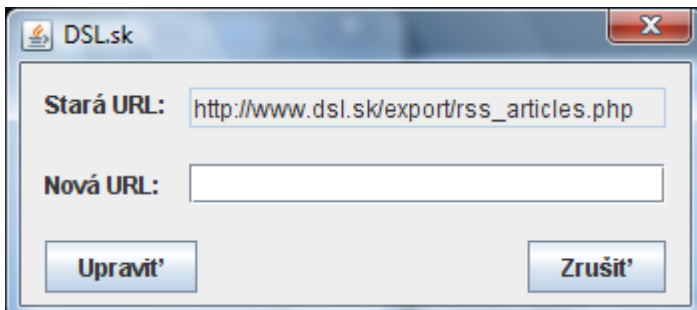
Dialóg pre pridávanie nových RSS kanálov môžete nájsť v hlavnom paneli pod názvom Pridať RSS.

Modifikovanie zapísaných RSS kanálov:

Umožňuje zmodifikovať URL pre vytvorený RSS kanál.

Inštrukcie:

- V hlavnom menu zvolte ponuku Súbor -> Upraviť RSS.
- Vyberte RSS kanál, ktorý chcete modifikovať a potvrďte svoj výber pomocou tlačidla Upraviť.
- Vyplňte pole Nová URL s novou adresou RSS kanálu.
- Potvrďte tlačidlom Upraviť.



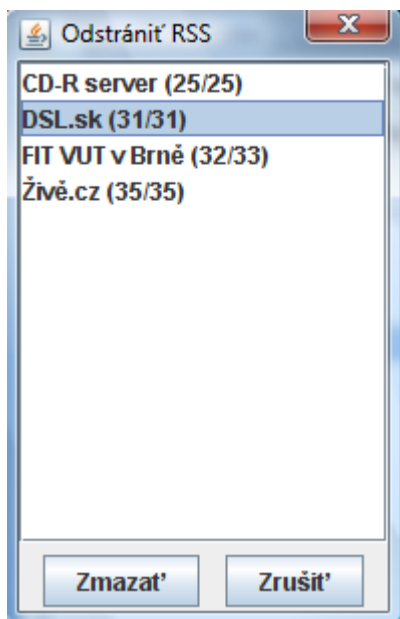
Obrázok 5 Modifikácia RSS

Zrušiť odber článkov z RSS kanálu:

Vymazáva všetky články, ktoré boli pre daný RSS kanál sťahované a vymaže tento zdroj zo zoznamu RSS kanálov.

Inštrukcie:

- Zvoľte v hlavnom menu ponuku Súbor -> Odstrániť RSS.
- Vyberte požadovaný RSS kanál.
- Potvrďte tlačidlom Zmazať.



Obrázok 6 Odstraňovanie RSS

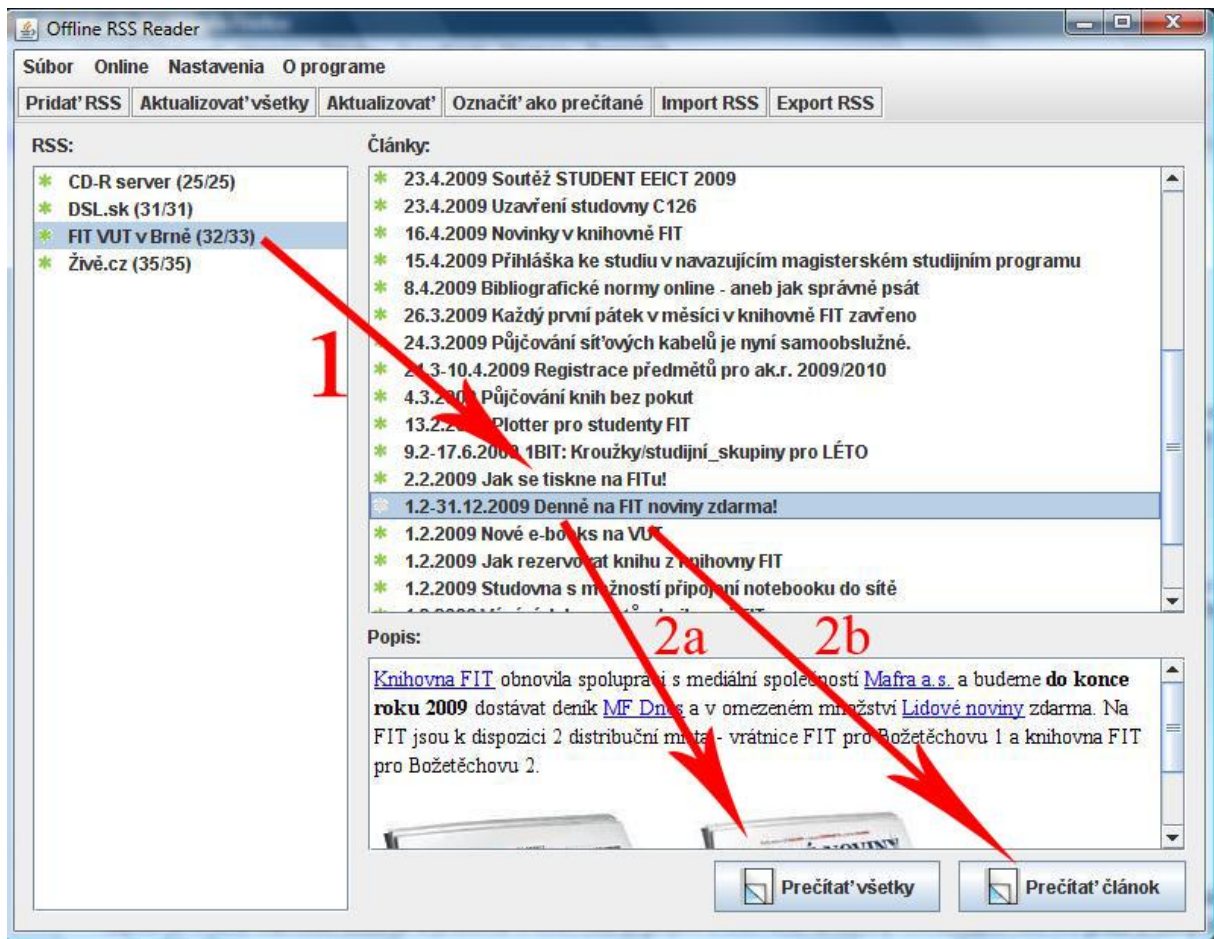
Iný spôsob zmazania je kliknúť pravým tlačidlom myši na položku zo zoznamu RSS kanálov a vybrať z ponuky Zmazať RSS.

Čítanie článkov:

Krátky popis k jednotlivým článkom je možné zobrazit' priamo v popise článku. Pre čítanie celého článku aplikácia využíva predvolený internetový prehliadač, v prípade Linuxu nadefinovaný.

Inštrukcie:

- Zo zoznamu RSS kanálov vyberte požadovanú položku, zobrazia sa všetky publikované články.
- Zo zoznamu článkov vyberte požadovaný článok, zobrazí sa popis daného článku.
- Tlačidlom Prečítať článok sa otvorí predvolený internetový prehliadač a zobrazí celý text daného článku.



Obrázok 7 Postupnosť pri zobrazovaní článkov

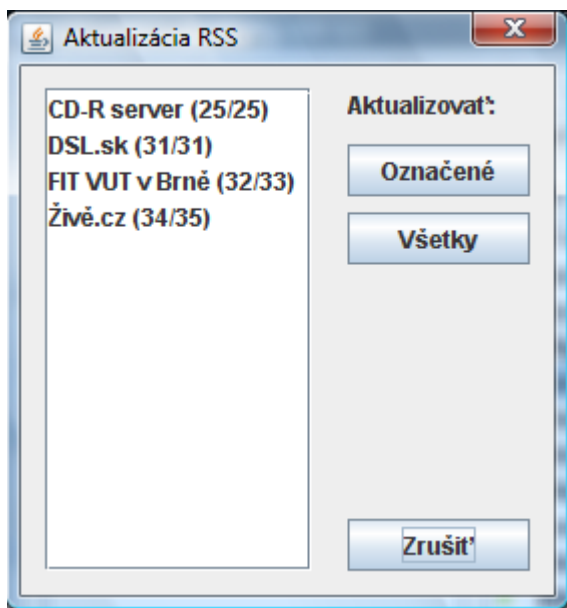
Tlačidlo **Prečítať všetky články**, ktoré je aktívne po vybratí RSS kanálu, slúži na zobrazenie zoznamu všetkých článkov v predvolenom internetovom prehliadači, z ktorého sú prístupné odkazy na každý dokument.

Aktualizácia RSS kanálov:

Kontroluje RSS kanály a sťahuje nové publikované články.

Inštrukcie:

- V hlavnom menu vyberte položku Online -> Aktualizácia RSS.
- Zvoľte požadovaný kanál(y).
- Výber potvrdíte tlačidlom **Označené** alebo pre aktualizáciu všetkých kanálov použite tlačidlo **Všetky**.



Obrázok 8 Aktualizácia RSS

S rovnakým výsledkom je možné využiť tlačidlá Aktualizovať z hlavného panelu alebo z kontextovej ponuky zoznamu RSS kanálov.

Import a export RSS kanálov:

Pre import a export RSS kanálov sa používa formát *OPML*, ktorý je univerzálnym riešením na prenášanie informácií medzi aplikáciami. Takýto dokument obsahuje zoznam názvov a adries kanálov.

Inštrukcie:

- Z hlavného panelu použite tlačidlo import alebo export.
- Zvoľte importovaný súbor alebo vytvorte nový pre exportovanie.
- Potvrďte tlačidlom Otvoriť alebo Uložiť.



Obrázok 9 Tlačidlá pre import a export RSS

Nastavenia aktualizácie RSS kanálov:

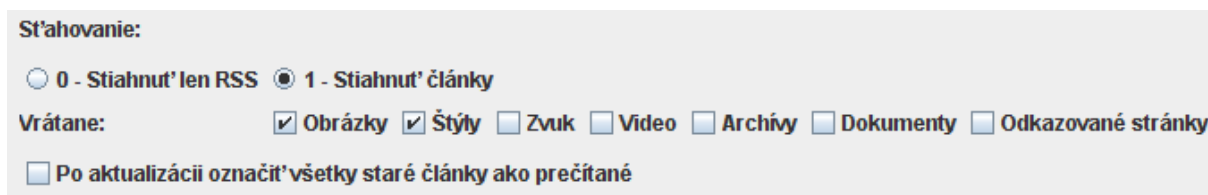
Poskytujú všetky možnosti nastavení pre sťahovanie dát publikovaných RSS kanálom.

Inštrukcie:

- V hlavnom menu zvoľte ponuku Nastavenia -> Nastavenia.
- V časti Sťahovanie vyberte požadované možnosti.
- Potvrďte tlačidlom Uložiť.

V prípade výberu možnosti 0 – Stiahnuť len RSS sa lokálne uloží len zoznam publikovaných článkov a ich stručný popis, celé texty článkov budú uložené formou odkazov na internetové stránky.

Po zvolení možnosti 1 – Stiahnuť články sa budú sťahovať odkazované webové stránky a ich obsah podľa vybratých nastavení. CSS štýly, obrázky, zvukové nahrávky, videá, archívy, dokumenty alebo aj odkazované stránky. Posledné spomínané nastavenie môže byť časovo náročné.



The image shows a settings dialog box for RSS feeds. It has a title bar 'Sťahovanie:' and two radio buttons: '0 - Stiahnuť len RSS' (unselected) and '1 - Stiahnuť články' (selected). Below this is a section 'Vrátane:' with several checkboxes: 'Obrázky' (checked), 'Štýly' (checked), 'Zvuk' (unchecked), 'Video' (unchecked), 'Archívy' (unchecked), 'Dokumenty' (unchecked), and 'Odkazované stránky' (unchecked). At the bottom, there is a checkbox 'Po aktualizácii označiť všetky staré články ako prečítané' (unchecked).

Obrázok 10 Možnosti nastavenia sťahovania RSS kanálov

Notifikácie

Každý RSS kanál a článok má svoj status, ktorý je reprezentovaný farebnou kvetinkou pred názvom.

- * RSS kanál alebo článok je v poriadku a má lokálne uložené dáta
- * Článok je v poriadku, ale dáta nie sú uložené lokálne a sú v podobe URL odkazov
- * RSS kanál alebo článok je prečítaný a neobsahuje neprečítané dáta
- * RSS kanál alebo článok je v chybe, nepodarilo sa stiahnuť požadované dáta

Vzniknuté chyby pri práci s aplikáciou sú užívateľovi zobrazené v podobe varovných dialógov. Podrobnejšie informácie o vzniknutej chybe nájdeme v logovacom systéme programu, ktorý je prístupný z hlavného menu pomocou ponuky Nastavenia -> Zobrazit' Log.

Príloha 2. Programová dokumentácia

programu *Offline RSS Reader*

Programová dokumentácia vo forme *JavaDoc* je umiestnená na CD, ktoré je dodávané s touto technickou prácou ako Príloha 3. CD so zdrojovými kódmi a aplikáciou *Offline RSS Reader*.

Príloha 3. CD so zdrojovými kódmi a aplikáciou *Offline RSS Reader*

Priložené CD obsahuje nasledujúce dáta:

- Adresár */Example* – obsahuje príklad súboru vo formáte *OPML*, ktorý obsahuje zoznam *RSS* kanálov a je ho možné importovať do aplikácie.
- Adresár */Install* – inštalačné balíčky aplikácie pre Windows a Linux.
- Adresár */JavaDoc* – programová dokumentácia vo forme *JavaDoc*.
- Adresár */Plagát* – plagát prezentujúci prácu v rôznych formátoch.
- Adresár */Source* – zdrojové kódy aplikácie a súbory pre import projektu do vývojového prostredia *NetBeans*.
- *PDF* súbor *technická_správa.pdf* – úplné znenie technickej správy vo formáte *PDF*.

Príloha 4. Plagát prezentujúci prácu



Vysoké Učení Technické v Brně
Fakulta Informačních Technologii
Ústav Počítačové Grafiky a Multimédií
Akademický rok 2008/2009

Offline RSS Reader

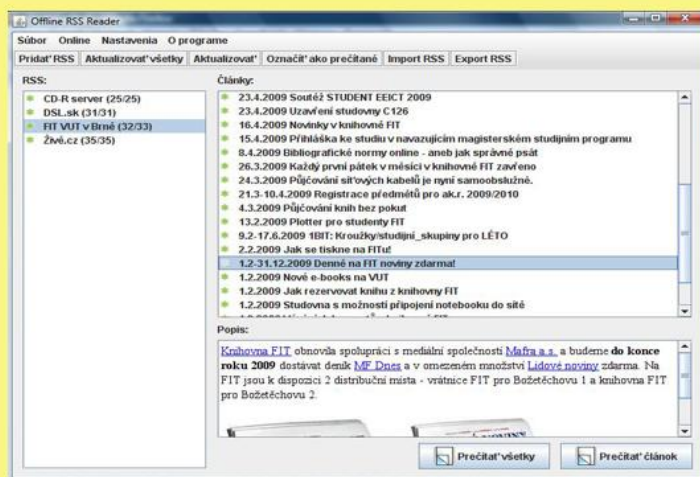
Autor: Erik Dresto

Prehľad:

Offline RSS Reader je RSS čítačka s prepracovaným off-line režimom, ktorá podporuje RSS formát vo verzii 2.0. Aplikácia je vytvorená v programovacom jazyku Java, ktorý zabezpečuje prenositeľnosť a správu pamäte. Automaticky kontroluje zapísané RSS kanály a vytvára lokálnu kópiu dát publikovaných daným kanálom. Off-line sťahovač podporuje pokročilé nastavenia, ktoré umožňujú podrobne nastaviť požadované prenášané dáta.

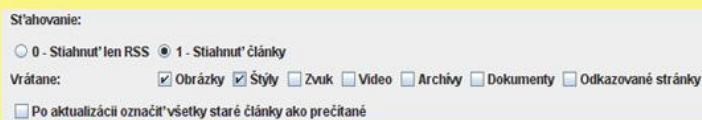
Užívateľské rozhranie:

Prívetivé užívateľské rozhranie, ktoré je možné jednoducho a rýchlo ovládať aj pomocou klávesnice.



Off-line sťahovač:

Podrobné nastavenia prenášania publikovaných dát.



Podpora OPML:

Možnosť importovania zoznamu RSS kanálov pomocou formátu OPML. Takisto je možné exportovať zoznam zapísaných kanálov.

Import RSS Export RSS

Notifikácie:

Každý RSS kanál a uložený článok má svoju stavovú ikonu, ktorá reprezentuje aktuálny stav.

- Dáta uložené lokálne *
- Dáta odkazované online *
- Prečítané dáta *
- Chybový stav *