Egyetemi doktori (PhD) értekezés tézisei

Webontológiák felhasználási lehetőségei

Possible Uses of Web Ontologies

Jeszenszky Péter

Témavezető: Dr. Bognár Katalin

Debreceni Egyetem
Természettudományi Doktori Tanács
Informatikai Tudományok Doktori Iskola

Debrecen, 2010

Tartalomjegyzék

Bevezetés

Jelen doktori értekezés a szerző a szemantikus webhez kapcsolódó új eredményeit tárgyalja, amelyek itt négy tézisben kerülnek összefoglalásra. Az 1. tézis részletes kifejtését az értekezés Modellezés című I. része tartalmazza (1. fejezet), a 2. és a 3. tézisét az RDF kinyerés című II. rész (2. és 3. fejezetek), végül a 4. tézisét a Csomagkezelés című III. rész.

1. tézis

Listák modellezése az OWL-ben

1.1. Problémafelvetés

Bármilyen meglepően hangzik, annak ellenére, hogy listaszerű szerkezetek kezelése szinte minden programozási nyelvben megvalósítható, az OWL-ben problémás a használatuk.

Noha kézenfekvő választásnak tűnhet az RDF kollekciók [10] használata OWL ontológiákban listák modellezéséhez, ez az út bizonyos esetekben nem elfogadható. Az OWL 1 [18] és OWL 2 [16] kedvező kiszámíthatósági tulajdonságokkal rendelkező OWL 1 DL és OWL 2 DL alnyelve ugyanis kizárólag magának az ontológiának az ábrázolásához teszi lehetővé a kollekció szókészlet alkalmazását, modellezéshez nem.

[14] egy az OWL DL határain belül maradó olyan általános megoldást ismertet, amely alkalmas egyedeket tartalmazó tipizált listák ábrázolásához. Fogyatékossága azonban, hogy nem támogatja a literálokat. A problémát alapvetően az jelenti, hogy sem az OWL 1 DL, sem pedig az OWL 2 DL nem tekinti egyedeknek a literálokat.

1.2. Eredmények

A szerző egy olyan általános tervezési mintát dolgozott ki, amely lehetővé teszi listaszerű szerkezetek problémamentes használatát OWL ontológiákban. A konstrukció egyaránt alkalmas kizárólag egyedeket, kizárólag literálokat, valamint literálokat és egyedeket vegyesen tartalmazó listákhoz, támogatja továbbá tipizált listák létrehozását és az elemek számának korlátozását. A gyakorlatban alkalmazhatóságot szem előtt tartva a listák modellezése szigorúan az OWL 1 DL (egyben az OWL 2 DL) keretei között történik. A szerző által adott megoldás Boris Motik az OWL 2 számára javasolt ötletének

kidolgozása a [14]-ben vázolt konstrukció alapján.[1]

A megvalósítás egyszerű és rugalmas. Pontosan meghatározott a készen adott listaosztályokból új listaosztályok létrehozásának módja, a szabályok betartása esetén az alosztályok megfelelően illeszkednek az osztályhierarchiába.

A szerző kifejlesztett egy olyan Java programot is, amely a listaszerkezetek használatához automatikusan előállítja a megfelelő OWL konstrukciókat.

1.3. Összefoglalás

Új, általános megoldást sikerült megadni listák OWL DL-ben modellezéséhez. A konstrukció lehetővé teszi kizárólag egyedeket, kizárólag literálokat, valamint egyedeket és literálokat vegyesen tartalmazó listák leírását. Támogatja tipizált listák kezeléséhez alkalmas további listosztályok definiálását, megadva ennek pontos módját. Kellően egyszerű és rugalmas, amely alkalmassá teszi a gyakorlati felhasználásokra.

[1] Boris Motik felvetését lásd az [19] címen elérhető levelezési lista üzenetben és az ennek kapcsán kibontakozott párbeszédben. Motik az OWL 2 szókészletének bővítését javasolta listák ábrázolásához alkalmas osztályokkal és tulajdonságokkal, amely javaslat végül nem került adaptálásra a szabványban.

2. tézis

RDF kinyerő konverziós programok

2.1. Problémafelvetés

A szemantikus web megvalósulásának előfeltétele az információk RDF-ben rendelkezésre állása az alkalmazások számára. Választ kell adnunk tehát arra a kérdésre, hogy miként juthatnak hozzá az alkalmazások RDF adatokhoz.

Elméletileg tetszőleges erőforráshoz társítható egy annak leírását RDF-ben szolgáltató másik erőforrás. Például HTML és XML dokumentumokhoz a társítás megvalósításához szabványos megoldás létezik. Az RDF gráfok XML szintaxisát (RDF/XML) definiáló [11] szabvány egy alkalmas mechanizmust ad metaadatokat hordozó külső RDF/XML dokumentumok HTML és XML dokumentumokhoz kapcsolásához. Noha ez egy egyszerű és kézenfekvő megoldás, meglehetősen rugalmatlan és kényelmetlen, éppen ezért nem is túl népszerű és elterjedt.

A W3C RDFa szabványa [8, 9] egy egyszerű és elegáns megoldást ad RDF kijelentések XHTML dokumentumokba beágyazásához. A beágyazás implicit módon, speciális XML attribútumok felhasználásával történik.

Ugyancsak W3C szabvány a GRDDL [13], amely lehetővé teszi RDF hármasok kinyerésére szolgáló transzformációk hozzárendelését XML dokumentumokhoz. URI-k szolgálnak a transzformációk azonosítására, amelyek megvalósítása a gyakorlatban többnyire XSLT stíluslapokkal történik, noha a szabvány nem zárja ki az egyéb megoldásokat (például szkriptek, programok alkalmazását).

Az utóbbi két megoldás kizárólag weboldalakhoz és XML dokumentumokhoz használhatók, azonban a weben számtalan egyéb fajta erőforrás fordul elő. Sok formátumhoz rendelkezésre állnak olyan konverziós eszközök, me-

lyek metaadatok RDF hármasok formájában történő kinyerésére szolgálnak. A különböző erőforrásokból RDF kijelentéseket kinyerni tudó konverziós eszközökre gyakran használják az **RDFizer** kifejezést, amely eredetileg egy SIMILE[1] alprojekt fedőneve. Ezeknek az eszközöknek a segítségével egy csapásra létező erőforrások serege válhat szemantikus web alkalmazások által is kiaknázható információforrássá.

2.2. Eredmények

Az RDFizers projekt [5] keretében olyan konverziós eszközök egy heterogén gyűjteménye érhető el, amelyeket a projekt személyzetének tagjai és külső közreműködők fejlesztettek. Nincs semmiféle megkötés a megvalósításra, például annak programozási nyelvére vagy az eszköz használatának módjára, az eszközök egyetlen közös jellemzője az, hogy valamilyen fajta erőforrásokból RDF kijelentéseket állítanak elő valamilyen alkalmas formában.

Az RDFizers projekthez hozzájárulva a szerző az alábbi állományformátumokhoz fejlesztett ki RDF kinyerő programokat:

- A BitTorrent protokoll [1] által használt metainfo állományok (közismert nevükön `.torrent` állományok)
- A számos operációs rendszer csomagkezelésének alapját képező RPM Package Manager által használt RPM csomagformátum [15]

A formátumokhoz értelemszerűen alkalmas RDF szókészletek is készültek. A megvalósítás Java nyelven történt, a programok szabad és nyílt forrású szoftverként állnak rendelkezésre a szerző weblapján [2].

Különböző RDF kinyerő eszközök használatának egységesítéséhez a szerző egy Java keretrendszert is kidolgozott.

2.3. Összefoglalás

A szerző olyan konverziós eszközöket fejlesztett ki, amelyek a szemantikát tükröző RDF kijelentéseket képesek kinyerni BitTorrent metainfo és RPM állományokból. A programok szabad és nyílt forrású szoftverként érhetők el az RDFizers projekt [5] vagy közvetlenül a szerző weblapjáról [2].

[1] A SIMILE [6] a MIT Computer Science and Artificial Intelligence Laboratory (MIT CSAIL) és a MIT Libraries közös, a nyílt forrás iránt elkötelezett projektje, amelynek keretében több a szemantikus webhez kapcsolódó fejlesztés is folyik.

3. tézis

Új böngészőfunkció fejlesztése

3.1. Problémafelvetés

Az Extensible Metadata Platform (XMP) [7] az Adobe Systems RDF-alapú metaadat keretrendszere erőforrások leírásához. Erőforrásként tekinthető egy állomány, vagy annak egy olyan része, amely egy feldolgozó alkalmazás számára jelentéssel bírhat, és amely a formátum szempontjából az állományszerkezet egy logikai komponense. Az XMP egy olyan adatmodellt definiál, amelynek ábrázolásához az RDF XML szintaxisának (RDF/XML) [11] egy részhalmazát használja.

Olyan szabványos metaadat szókészleteket biztosít továbbá, amelyeket a legkülönfélébb alkalmazások használhatnak erőforrások – például digitális képek, hang- és videó állományok – leírására.

Kulcsfontosságú jellemzője, hogy lehetővé teszi metaadatok beágyazását állományokba úgynevezett XMP csomagok formájában. Számos elterjedt állományformátumhoz – például AVI, JPEG, PDF, MPEG, PNG, PostScript, TIFF – pontosan meghatározza a beágyazás fizikai megvalósítását is.

Az XMP legnagyobb előnye abban rejlik, hogy szabványos és állományformátum-független módját adja digitális képek és egyéb erőforrások metaadatokkal annotálásának, és hogy a beágyazás révén a metaadatok együtt utaznak az állományokkal azok átvitele során. Sok lehetőséget rejt szemantikus web alkalmazások számára is, hiszen az állományokat hasznosan kiaknázható metaadat-forrásokká teszi.

Az XMP az Adobe számára stratégiai fontosságú, gyakorlatilag valamennyi terméke (például az Adobe InDesign, Adobe Photoshop és Adobe Reader) támogatja. Nyílt szabványról lévén szó, sok további alkalmazás nyújt XMP támogatást. Sem a hagyományos, sem a kísérleti szemantikus web böngészők nem fordítottak azonban eddig megfelelő figyelmet erre az

ígéretes technológiára.

3.2. Eredmények

A szerző egy olyan új funkciót fejlesztett ki a Firefox böngészőprogramhoz, amellyel egy weboldalról elérhető erőforrásokból lehet kinyerni az azokba beágyazott XMP metaadatokat. A művelet képekre vagy az oldalon hiperhivatkozások célpontjaként megadott erőforrásokra értelmezett. A metaadatok kinyerhetők állományonként, de lehetőség van valamennyi erőforrás egy menetben feldolgoztatására is.

A új böngészőfunkció a Firefox böngészőhöz rendelkezésre álló szabad és nyílt forrású Piggy Bank [3] böngésző kiterjesztésen alapul. A Piggy Bank a szemantikus web böngészők egyik úttörője. Szemantikus web technológiákat és olyan innovatív megoldásokat alkalmaz, amelyek a böngészésnek egy újfajta élményét adják.

A szerző valójában a Piggy Bank kiterjesztéshez implementálta az új funkciót, képessé téve azt XMP kinyerésre. A kinyerés végrehajtása a szerző által biztosított XMP kinyerő webszolgáltatással történik. A webszolgáltatás mindössze néhány formátum kezelésére képes, azonban a kliensek számára transzparens módon teszi lehetővé további formátumokat kezelő kinyerők hozzáadását.

A kinyerés után az XMP csomagok a Piggy Bank felhasználói felületén manipulálhatók.

3.3. Összefoglalás

A szerző egy mai napig újszerű és egyedülálló böngészőfunkciót valósított meg, amely lehetővé teszi a Firefox böngészőprogramban XMP metaadatok kinyerését és böngészését.

4. tézis

Csomag metaadatok publikálása „kapcsolt adatokként”

4.1. Problémafelvetés

A szoftvercsomag kifejezés egységnyi telepíthető szoftvert jelent. Egy szoftvercsomag általában egyetlen olyan archív állomány formájában adott, amely a telepítendő számítógépes szoftvert tartalmazza.

A modern Linux-disztribúciókat és Unix-szerű operációs rendszereket sok száz vagy több ezer csomag alkotja. A csomagok kezelését ezekben a rendszerekben egy úgynevezett csomagkezelő rendszer valósítja meg. A csomagkezelő rendszer egy olyan alkalmazás, amely támogatást biztosít csomagok automatikus és egységes módon történő telepítéséhez, valamint olyan további kapcsolódó feladatokhoz, mint például a csomagok eltávolítása és frissítése. A csomagkezelő rendszerek általában egy adott csomagformátumot használnak.

Nem csupán az operációs rendszerek élvezhetik szoftvercsomagok előnyeit, hanem akár alkalmazói programok is felhasználhatják őket új funkciók hozzáadásához. Egy példa a szabad és nyílt forrású R statisztikai környezet [4], amely saját csomagformátummal és csomagkezelő rendszerrel rendelkezik.

Noha sok csomagkezelési megoldás létezik, a csomagok egy közös jellemzője, hogy sok metaadatot hordoznak, mint például a csomag teljes neve, verziószáma, leírása, a tartalmazott szoftver licence és a függőségek listája.

Mivel a szerző egyaránt lelkes híve a Linux-módra történő csomagkezelésnek és a szemantikus webnek, elég nyilvánvaló volt számára a következő feladat: tegyünk elérhetővé szoftvercsomag metaadatokat szemantikus web alkalmazások számára is!

4.2. Eredmények

A Linked Data („kapcsolt adatok") [12] kifejezés RDF adatok közzétételének egy olyan módját jelenti, amely hivatkozás-feloldható URI-k [17] használatán alapul. Esetünkben ez lehetővé teszi a felhasználók és alkalmazások számára, hogy a böngészés élményét adó módon függőségeket és egyéb kapcsolatokat reprezentáló RDF linkek követésével navigáljanak a csomagok között.

A munka részeként a szerző olyan eszközöket fejlesztett ki, amelyek csomagokból metaadatokat nyernek ki és RDF-be alakítják az információkat. Jelenleg az RPM [15] és Debian csomagok [22] támogatottak, valamint az R csomagok [21]. Az eszközök funkcionalitása elérhető egy REST-stílusú webszolgáltatásként [20] is, amely egy csomag URI-ját kapja meg és RDF-ben adja vissza a metaadatokat.

A fenti eszközök speciális RDF szókészleteket használnak csomag metaadatok ábrázolásához. A munka eredményeként több OWL webontológia készült, amelyek a támogatott csomagformátumokhoz definiálják a szókészleteket.

A fentebb tárgyalt programokat egyetlen állományhoz lehet használni, nem „kapcsolt adatokat" állítanak elő, nem alkalmasak csomagok közötti függőségek és egyéb kapcsolatok vizsgálatához. Célunk eléréséhez a csomagokat egy kontextusban kell tekinteni. Esetünkben a kontextus egy úgynevezett tároló. A csomagtárolók olyan helyek, ahol csomagok összessége áll rendelkezésre telepítésre. Általában hálózaton keresztül érik el őket a csomagkezelő rendszerek, de rendelkezésre állhatnak CD/DVD lemezeken is. A fenti három csomagformátumon alapuló csomagkezelő rendszerek mindegyike támogatja tárolók használatát.

A szerző megoldása jelenleg R csomagokat tartalmazó CRAN-stílusú tárolókhoz [21] biztosít Linked Data nézetet. A Linked Data publikálás a tároló adatbázisán alapul, amelyből konverzióval egy RDF gráf jön létre. A csomag metaadatok egy SPARQL végponton keresztül nyerhetők ki, de akár egy közönséges webböngészőben is megtekinthetők.

4.3. Összefoglalás

A szerző olyan programokat fejlesztett ki, amelyek szoftvercsomag metaadatokat tesznek elérhetővé szemantikus web alkalmazások számára. Olyan OWL ontológiák készültek több elterjedten használt csomagformátumhoz, amelyek alkalmasak szoftvercsomag tárolók metaadatainak „kapcsolt adatokként" történő publikálásához. A szerző megmutatta, hogyan valósítható meg a gyakorlatban ilyen Linked Data szolgáltatás. A munka fő eredménye

az, hogy rámutat arra, hogy értékes Linked Data adathalmazokká lehet változtatni a csomagtárolókat.

Tudományos közlemények

Az értekezés témájához kapcsolódó közlemények

Referált közlemények

1. Péter Jeszenszky: *Adding XMP support to Firefox*, Acta Cybernetica, 18 (2): pp. 257–274, 2007.

Konferenciakiadványban megjelent dolgozatok

1. Péter Jeszenszky: *Browsing the Semantic Web*, Proceedings of the 7^{th} International Conference on Applied Informatics, Volume 2, pp. 237–245, 2007.

Konferencia előadások

1. Jeszenszky Péter: *Oktatási webontológia fejlesztése és lehetséges alkalmazásai*, Informatika a Felsőoktatásban 2005, Debrecen, 2005. augusztus 24–26.

2. Jeszenszky Péter: *XML és szemantikus web az oktatásban*, Informatika a Felsőoktatásban 2005, Debrecen, 2005. augusztus 24–26.

3. Péter Jeszenszky: *Web Ontology for Software Package Management*, 8th International Conference on Applied Informatics, Eger, January 28–31 2007.

Meghívott előadások

1. Jeszenszky Péter: *Böngészés a szemantikus weben*, 7. Gyires Béla Informatikai Nap, Debrecen, 2006. december 15.

2. Jeszenszky Péter: *A Web jövője?*, 2. Debreceni Szabad Szoftver Este, Debrecen, 2008. október 8.

3. Jeszenszky Péter: *Szoftvercsomagok modellezése webontológiákkal*, 13. Gyires Béla Informatikai Nap, Debrecen, 2010. december 17.

Szoftverek

1. Firefox XMP Support.

2. OWLListUtils.
URL `http://www.inf.unideb.hu/~jeszy/OWLListUtils`.

3. RDFizers. URL `http://www.inf.unideb.hu/~jeszy/rdfizers/`.

További közlemények

Referált közlemények

1. Péter Antal, Norbert Bátfai, István Fazekas, Péter Jeszenszky. *The mobiDIÁK Educational Portal*: Journal of Universal Computer Science, Volume 12, Number 9, pp. 1118–1127, 2006.

2. Péter Jeszenszky: *Teaching XML*, Teaching Mathematics and Computer Science, Volume V, Issue II, pp. 317–335, 2007.

Konferenciakiadványban megjelent dolgozatok

1. István Bencze, Balázs Fark, László Hatala, Péter Jeszenszky: *Server side PDF generation based on LaTeX templates*, TUGboat Volume 27, Number 1, EuroTeX 2006 Proceedings, pp. 51–56, 2006.

2. Bátfai Norbert, Jeszenszky Péter, Bartha Csaba, Gilányi Attila, Széll Sándor, Szimeonov György, Vaskó Gábor, Terdik György: *Műholdas helymeghatározás alkalmazása a labdajátékokban*, Az elmélet és a gyakorlat találkozása a térinformatikában, Térinformatikai Konferencia 2010, Debrecen, 223–231. oldal.

Konferencia előadások

1. Jeszenszky Péter: *Keresőalgoritmusok hatékony implementálása Java nyelven*, Informatika a Felsőoktatásban 2008, Debrecen, 2008. augusztus 27–29.

2. Jeszenszky Péter: *XML oktatás az új típusú programtervező informatikus B.Sc. képzésben*, Informatika a Felsőoktatásban 2008, Debrecen, 2008. augusztus 27–29.

3. Jeszenszky Péter: *Szoftveres szemléltetés a mesterséges intelligencia oktatásban*, Multimédia az Oktatásban 2009, Debrecen, 2009. június 24–25.

Meghívott előadások

1. Jeszenszky Péter: *Matematika a számítógéppel*, FSF.hu Roadshow, Debrecen, 2006. április 3.

2. Jeszenszky Péter: *Az R statisztikai és grafikai környezet*, Magyar Aktuárius Társaság (MAT) – Őszi Iskola, 2008. november 14–15.

3. Jeszenszky Péter: *Az R nyelv*, 4. Debreceni Szabad Szoftver Este, Debrecen, 2009. február 12.

Szoftverek

1. Mesterséges intelligencia keretrendszer Java-hoz (keresőalgoritmusok, kétszemélyes játékok).
   URL `http://www.inf.unideb.hu/~jeszy/mestint`.

Irodalomjegyzék

[1] Bittorrent Protocol Specification v1.0. URL http://wiki.theory.org/BitTorrentSpecification.

[2] RDFizers developed by Peter Jeszenszky. URL http://www.inf.unideb.hu/~jeszy/rdfizers/.

[3] Piggy Bank. URL http://simile.mit.edu/piggy-bank/.

[4] The R Project for Statistical Computing. URL http://www.r-project.org/.

[5] RDFizers. URL http://simile.mit.edu/wiki/RDFizers.

[6] SIMILE Project. URL http://simile.mit.edu/.

[7] Adobe XMP: Adding intelligence to media. URL http://www.adobe.com/products/xmp/.

[8] Ben Adida and Mark Birbeck. RDFa Primer. W3C Recommendation, 2008. URL http://www.w3.org/TR/xhtml-rdfa-primer/.

[9] Ben Adida, Mark Birbeck, Shane McCarron, and Steven Pemberton. RDFa in XHTML: Syntax and Processing. W3C Recommendation, 2008. URL http://www.w3.org/TR/rdfa-syntax/.

[10] Frank Manola and. Eric Miller. W3C Recommendation, 2004. URL http://www.w3.org/TR/rdf-primer/.

[11] Dave Beckett. RDF/XML Syntax Specification (Revised). W3C Recommendation, 2004. URL http://www.w3.org/TR/rdf-syntax-grammar/.

[12] Tim Berners-Lee. Linked Data, 2006. URL http://www.w3.org/DesignIssues/LinkedData.html.

[13] Dan Connolly. Gleaning Resource Descriptions from Dialects of Languages (GRDDL). W3C Recommendation, 2007. URL http://www.w3.org/TR/grddl/.

[14] Nick Drummond, Alan Rector, Robert Stevens, Georgina Moulton, Matthew Horridge, Hai H. Wang, and Julian Seidenberg. Putting OWL in Order: Patterns for Sequences in OWL. In *OWL: Experiences and Directions*, 2006. URL http://www.webont.org/owled/2006/acceptedLong/submission_12.pdf.

[15] Eric Foster-Johnson. RPM Guide, 2005. URL http://rpm5.org/docs/rpm-guide.pdf.

[16] W3C OWL Working Group. OWL 2 Web Ontology Language Document Overview. W3C Recommendation, 2009. URL http://www.w3.org/TR/owl-overview/.

[17] Ian Jacobs and Norman Walsh. Architecture of the World Wide Web, Volume One. W3C Recommendation, 2004. URL http://www.w3.org/TR/webarch/.

[18] Deborah L. McGuinness and Frank van Harmelen. OWL Web Ontology Language Overview. W3C Recommendation, 2004. URL http://www.w3.org/TR/owl-features/.

[19] Boris Motik. A proposal for ISSUE-104 (built-in vocabulary), 2008. URL http://lists.w3.org/Archives/Public/public-owl-wg/2008Jun/0070.html.

[20] Leonard Richardson and Sam Ruby. *RESTful Web Services*. O'Reilly Media, 2007. ISBN 978-0-596-80168-7.

[21] R Development Core Team. *Writing R Extensions*, 2010. URL http://cran.r-project.org/doc/manuals/R-exts.html. version 2.11.1.

[22] The Debian Policy Mailing List. *Debian Policy Manual*, 2010. URL http://www.debian.org/doc/debian-policy/. version 3.8.4.0.

Introduction

This doctoral dissertation discusses new results by the author related to the Semantic Web, that are summarized in four theses here. The detailed discussion of Thesis 1 can be found in Part I: Modelling of the dissertation (Chapter 1), Thesis 2 and 3 in Part II: RDF Extraction (Chapters 2 and 3), and Thesis 4 in Part III: Package Management.

Thesis 1

Modelling Lists in OWL

1.1 Problem Proposal

Although it may sound surprising, despite the fact that the handling of list-like structures can be implemented in almost all programming languages, their use in OWL is problematic.

Whereas the use of RDF collections for modelling lists in OWL may seem evident, this way is inappropriate in some cases. The OWL 1 DL and OWL 2 DL sublanguages of OWL 1 and OWL 2, that have favourable computational properties, allow the use of the collection vocabulary only for the representation of the ontology itself, and not for modelling.

[14] describes a general solution remaining strictly within the limits of OWL that is suitable for representing typed lists containing individuals. Its deficiency is the lack of support for literals. The problem is essentially that neither OWL 1 DL nor OWL 2 DL treats literals as individuals.

1.2 Results

The author has developed a general design pattern that makes it possible to use list-like structures in OWL without any problems. The construct is equally applicable for lists that contain only individuals or literals, or a mixture of both. It also supports the creation of typed lists and constraining the number of elements. Bearing in mind practical applicability, the modelling of lists is carried out strictly within OWL 1 DL (and OWL 2 DL). The solution provided by the author is a development of the idea proposed by Boris Motik for OWL 2, based on the construct outlined in [14].[1]

[1]For Boris Motik's proposal see the mailing list message that can be found at location [19], and also the discussion started by the message. Motik suggested extending the

The implementation is simple and flexible. The method of creating new list classes from existing list classes is clearly defined, compliance with the rules guarantees that the subclasses adequately fit in the class hierarchy.

The author has also developed a Java program that automatically creates the appropriate OWL constructs for using the list structures.

1.3 Summary

A new and general solution for modelling lists in OWL DL has been succesfully been provided. The construct makes it possible to describe lists containing only individuals or literals, or a mixture of both. It provides support for defining additional list classes suitable for handling typed lists giving its precise method.

vocabulary of OWL 2 with classes and properties suitable for representing lists, which proposal was not finally adapted into the standard eventually.

Thesis 2

RDF Extractor Conversion Programs

2.1 Problem Proposal

A necessary prerequisite for the realisation of the Semantic Web is the availability of information to applications in RDF. Therefore, we must answer the question of how applications may obtain RDF data.

Theoretically, any resource may be associated with another resource that provides a description of the former in RDF. For example, a standard solution exists for HTML and XML documents to implement the association. The [11] standard that defines the XML syntax for RDF graphs (RDF/XML) provides a suitable mechanism to link external RDF/XML documents carrying metadata to HTML and XML documents. Although it is simple and obvious, but also a rather inflexible and cumbersome solution, and therefore is not too popular and widely used.

The RDFa standard [8, 9] from W3C provides a simple and elegant solution for embedding RDF statements into XHTML documents. The embedding is done implicitly using special XML attributes.

GRDDL [13] is also a W3C standard, that makes it possible to associate RDF triple extractor transformations with XML documents. URIs are used to identify transformations, which, in practice, are usually implementented by XSLT stylesheets, although the standard does not preclude other solutions (e.g. the use of scripts or programs).

The latter two solutions can only be used for web pages and XML documents, however many other kinds of resources can be found on the web. Conversion tools are available for many formats, that can extract metadata in the form of RDF triples. Conversion tools that can extract RDF state-

ments from various resources are often referred to as **RDFizers**, which was originally the code name for a SIMILE subproject.[1] Using these tools a number of existing resources may become available to Semantic Web applications as information sources that can be exploited effectively.

2.2 Result

The RDFizers project [5] offers a heterogeneous collection of conversion tools, each of which was developed by members of the project staff or external contributors. No constraints are imposed on the implementation (e.g. programming language, method of use), their only common feature is that they all produce RDF statements in a suitable form processing some kind of resources.

Contributing to the RDFizers project the author has developed RDF extractor programs for the following file formats:

- Metainfo files used by the BitTorrent protocol [1] (commonly known as `.torrent` files)
- The RPM package format [15] used by the RPM Package Manager, on which many operating systems' package management is based.

It is obvious that suitable RDF vocabularies have also been developed for the formats. The implementation was done in Java, the programs are available as free and open source software at the authors web page [2].

The author has also developed a Java framework to standardize the use of different RDF extractor tools.

2.3 Summary

The author has developed conversion tools that can extract RDF statements from files, that reflect the underlying semantics. The programs are available from the web page of the RDFizers project [5] or directly from the authors web page [2] as free and open source software.

[1] SIMIE [6] is a joint project of MIT Computer Science and Artificial Intelligence Laboratory (MIT CSAIL) and MIT Libraries committed to open source, in which several Semantic Web releated development are in progress.

Thesis 3

Developing a New Browser Feature

3.1 Problem Proposal

Extensible Metadata Platform (XMP) [7] is Adobe Systems' RDF-based metadata framework for describing resources. A resource may be a file or a portion of it that may be meaningful to a processing application in itself, and that is also a distinct logical component of the file structure. XMP defines a data model that can be represented using a subset of the XML syntax of RDF (RDF/XML) [11].

Furthermore, it provides standard metadata vocabularies that can be used by various applications for describing resources (e.g. digital images, audio and video files).

Its key feature is the support for embedding metadata into application files in the form of so-called XMP packets. The physical implementation of embedding is also clearly defined for many popular file formats, such as AVI, JPEG, PDF, MPEG, PNG, PostScript, and TIFF.

The most important advantage of XMP is that it provides a file format independent way to annotate digital images and other resources with metadata, and metadata is transmitted together with the files during the transfer by means of the embedding. XMP offers many opportunities for Semantic Web applications also, because it turns files into metadata sources that can be usefully exploited.

XMP has strategic importance for Adobe, practically all of its products supports it (e.g. Adobe InDesign, Adobe Photoshop, Adobe Reader). Since it is an open standard, many other applications provide XMP support. However, neither traditional web browsers, nor experimental Semantic Web

browsers have given appropriate attention to this promising technology until now.

3.2 Results

The author has developed a new feature for the Firefox web browser, the purpose of which is to extract embedded XMP metadata from resources available on a web page. The operation can be performed on images or resources that appear as hyperlink targets on the page. Metadata can be extracted on a file-by-file basis, but it is also possible to process all resources in one turn.

The new browser feature is based on the free and open source Piggy Bank [3] browser extension. Piggy Bank is one of the pioneers of Semantic Web browsers. It employs Semantic Web technologies and such innovative solutions that provide a novel browsing experience.

Actually, the author has implemented the new feature for Piggy Bank, making it capable of extracting XMP metadata. An XMP extractor web service provided by the author is used for performing the extraction. The web service can handle only few file formats, however, extractors handling other formats can be added transparently to the clients.

XMP packets can be manipulated by Piggy Bank after the extraction is performed.

3.3 Summary

The author has developed a browser function enabling the extraction and browsing of XMP metadata in the Firefox web browser, which is novel and unique to this day.

Thesis 4

Publishing Package Metadata as Linked Data

4.1 Problem Proposal

The term software package refers to a unit of distributable and installable software. A software package is usually provided as a single archive file that contains computer software to be installed.

Modern Linux distributions and Unix-like operating systems consist of hundreds or thousands of software packages. The handling of software packages on these systems is carried out by a package management system. A package management system is an application that provides support to install packages automatically and in a uniform manner, and to other related tasks, such as to remove and update installed packages. A package management system usually uses a specific package format.

Not only operating systems can benefit from the advantages of software packages, even application software may use them to add new functionality. An example is the free and open source R statistical computing environment [4] that has its own software package format and package management system.

Although many package management solutions exist, a common characteristic of packages is that they all carry a lot of metadata, such as full name, version number, description, license of the contained software, and the list of its dependencies.

Since the author is an enthusiast of both the Linux way to package management and the Semantic Web, he found the following task to be a fairly evident one: make software package metadata to be also available to Semantic Web applications!

4.2 Results

The term Linked Data [12] refers to a style of publishing RDF on the Web, that is based on the use of dereferenceable URIs [17]. In our case, this allows users and applications to navigate between packages following RDF links that represent dependencies and other relationships, giving people the experience of browsing.

As part of the work the author has developed tools to retrieve metadata from packages and to transform this information to RDF. Currently, RPM [15] and Debian packages [22] are supported, as well as R packages [21]. The functionality of these tools is also available as a RESTful web service [20] that accepts a package URI and returns metadata in RDF.

The above tools use specialized RDF vocabularies to represent package metadata. As a result of this work several OWL web ontologies are available to define these vocabularies for the supported package formats.

The above discussed utilities operate on single files and they do not produce Linked Data and are not suitable to explore dependencies and other relationships between packages. In order to achieve our goal, packages must be considered in context. In our case, that context is a so-called repository. Software repositories are locations where collections of packages are available for installation. They are usually accessed by package management systems over network, but can also be available on CD/DVD. Package management systems based on any of the above three formats support the use of repositories.

The author's solution provides Linked Data views of CRAN-style repositories [21] holding R packages. Linked Data publishing in a repository is based on its package database that is converted into an RDF graph. Package metadata can be retrieved via a SPARQL endpoint, but can also be viewed in an ordinary web browser.

4.3 Summary

The author has developed computer programs that make package metadata available to Semantic Web applications. OWL ontologies are available for several widely used package formats, that are also suitable for publishing repository metadata as Linked Data. The author showed how such a Linked Data service can be implemented in practice. The main contribution of the work is that it shows that package repositories can be turned into valuable Linked Data sets.

List of Publications

Publications in Dissertation Topics

Reviewed Papers

1. Péter Jeszenszky: *Adding XMP support to Firefox*, Acta Cybernetica, 18 (2): pp. 257–274, 2007.

Conference Proceedings

1. Péter Jeszenszky: *Browsing the Semantic Web*, Proceedings of the 7^{th} International Conference on Applied Informatics, Volume 2, pp. 237–245, 2007.

Conference Talks

1. Jeszenszky Péter: *Oktatási webontológia fejlesztése és lehetséges alkalmazásai*, Informatika a Felsőoktatásban 2005, Debrecen, 2005. augusztus 24–26.

2. Jeszenszky Péter: *XML és szemantikus web az oktatásban*, Informatika a Felsőoktatásban 2005, Debrecen, 2005. augusztus 24–26.

3. Péter Jeszenszky: *Web Ontology for Software Package Management*, 8th International Conference on Applied Informatics, Eger, January 28–31 2007.

Invited Talks

1. Jeszenszky Péter: *Böngészés a szemantikus weben*, 7. Gyires Béla Informatikai Nap, Debrecen, 2006. december 15.

2. Jeszenszky Péter: *A Web jövője?*, 2. Debreceni Szabad Szoftver Este, Debrecen, 2008. október 8.

3. Jeszenszky Péter: *Szoftvercsomagok modellezése webontológiákkal*, 13. Gyires Béla Informatikai Nap, Debrecen, 2010. december 17.

Software

1. Firefox XMP Support.

2. OWLListUtils.
URL `http://www.inf.unideb.hu/~jeszy/OWLListUtils`.

3. RDFizers. URL `http://www.inf.unideb.hu/~jeszy/rdfizers/`.

Other Publications

Reviewed Papers

1. Péter Antal, Norbert Bátfai, István Fazekas, Péter Jeszenszky. *The mobiDIÁK Educational Portal*: Journal of Universal Computer Science, Volume 12, Number 9, pp. 1118–1127, 2006.

2. Péter Jeszenszky: *Teaching XML*, Teaching Mathematics and Computer Science, Volume V, Issue II, pp. 317–335, 2007.

Conference Proceedings

1. István Bencze, Balázs Fark, László Hatala, Péter Jeszenszky: *Server side PDF generation based on LaTeX templates*, TUGboat Volume 27, Number 1, EuroTeX 2006 Proceedings, pp. 51–56, 2006.

2. Bátfai Norbert, Jeszenszky Péter, Bartha Csaba, Gilányi Attila, Széll Sándor, Szimeonov György, Vaskó Gábor, Terdik György: *Műholdas helymeghatározás alkalmazása a labdajátékokban*, Az elmélet és a gyakorlat találkozása a térinformatikában, Térinformatikai Konferencia 2010, Debrecen, 223–231. oldal.

Conference Talks

1. Jeszenszky Péter: *Keresőalgoritmusok hatékony implementálása Java nyelven*, Informatika a Felsőoktatásban 2008, Debrecen, 2008. augusztus 27–29.

2. Jeszenszky Péter: *XML oktatás az új típusú programtervező informatikus B.Sc. képzésben*, Informatika a Felsőoktatásban 2008, Debrecen, 2008. augusztus 27–29.

3. Jeszenszky Péter: *Szoftveres szemléltetés a mesterséges intelligencia oktatásban*, Multimédia az Oktatásban 2009, Debrecen, 2009. június 24–25.

Invited Talks

1. Jeszenszky Péter: *Matematika a számítógéppel*, FSF.hu Roadshow, Debrecen, 2006. április 3.

2. Jeszenszky Péter: *Az R statisztikai és grafikai környezet*, Magyar Aktuárius Társaság (MAT) – Őszi Iskola, 2008. november 14–15.

3. Jeszenszky Péter: *Az R nyelv*, 4. Debreceni Szabad Szoftver Este, Debrecen, 2009. február 12.

Software

1. Artificial intelligence framework for Java (search algorithms, two-player games). URL `http://www.inf.unideb.hu/~jeszy/mestint`.

Bibliography

[1] Bittorrent Protocol Specification v1.0. URL http://wiki.theory.org/BitTorrentSpecification.

[2] RDFizers developed by Peter Jeszenszky. URL http://www.inf.unideb.hu/~jeszy/rdfizers/.

[3] Piggy Bank. URL http://simile.mit.edu/piggy-bank/.

[4] The R Project for Statistical Computing. URL http://www.r-project.org/.

[5] RDFizers. URL http://simile.mit.edu/wiki/RDFizers.

[6] SIMILE Project. URL http://simile.mit.edu/.

[7] Adobe XMP: Adding intelligence to media. URL http://www.adobe.com/products/xmp/.

[8] Ben Adida and Mark Birbeck. RDFa Primer. W3C Recommendation, 2008. URL http://www.w3.org/TR/xhtml-rdfa-primer/.

[9] Ben Adida, Mark Birbeck, Shane McCarron, and Steven Pemberton. RDFa in XHTML: Syntax and Processing. W3C Recommendation, 2008. URL http://www.w3.org/TR/rdfa-syntax/.

[10] Frank Manola and. Eric Miller. W3C Recommendation, 2004. URL http://www.w3.org/TR/rdf-primer/.

[11] Dave Beckett. RDF/XML Syntax Specification (Revised). W3C Recommendation, 2004. URL http://www.w3.org/TR/rdf-syntax-grammar/.

[12] Tim Berners-Lee. Linked Data, 2006. URL http://www.w3.org/DesignIssues/LinkedData.html.

[13] Dan Connolly. Gleaning Resource Descriptions from Dialects of Languages (GRDDL). W3C Recommendation, 2007. URL http://www.w3.org/TR/grddl/.

[14] Nick Drummond, Alan Rector, Robert Stevens, Georgina Moulton, Matthew Horridge, Hai H. Wang, and Julian Seidenberg. Putting OWL in Order: Patterns for Sequences in OWL. In *OWL: Experiences and Directions*, 2006. URL http://www.webont.org/owled/2006/acceptedLong/submission_12.pdf.

[15] Eric Foster-Johnson. RPM Guide, 2005. URL http://rpm5.org/docs/rpm-guide.pdf.

[16] W3C OWL Working Group. OWL 2 Web Ontology Language Document Overview. W3C Recommendation, 2009. URL http://www.w3.org/TR/owl-overview/.

[17] Ian Jacobs and Norman Walsh. Architecture of the World Wide Web, Volume One. W3C Recommendation, 2004. URL http://www.w3.org/TR/webarch/.

[18] Deborah L. McGuinness and Frank van Harmelen. OWL Web Ontology Language Overview. W3C Recommendation, 2004. URL http://www.w3.org/TR/owl-features/.

[19] Boris Motik. A proposal for ISSUE-104 (built-in vocabulary), 2008. URL http://lists.w3.org/Archives/Public/public-owl-wg/2008Jun/0070.html.

[20] Leonard Richardson and Sam Ruby. *RESTful Web Services*. O'Reilly Media, 2007. ISBN 978-0-596-80168-7.

[21] R Development Core Team. *Writing R Extensions*, 2010. URL http://cran.r-project.org/doc/manuals/R-exts.html. version 2.11.1.

[22] The Debian Policy Mailing List. *Debian Policy Manual*, 2010. URL http://www.debian.org/doc/debian-policy/. version 3.8.4.0.