Raúl García-Castro
Dieter Fensel
Grigoris Antoniou (Eds.)

LNCS 7117

The Semantic Web: ESWC 2011 Workshops

ESWC 2011 Workshops,
Heraklion, Greece, May 2011
Revised Selected Papers

Springer

Lecture Notes in Computer Science 7117

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Raúl García-Castro
Dieter Fensel   Grigoris Antoniou (Eds.)

The Semantic Web: ESWC 2011 Workshops

ESWC 2011 Workshops,
Heraklion, Greece, May 29-30, 2011
Revised Selected Papers

Springer

Volume Editors

Raúl García-Castro
Polytechnical University of Madrid
Campus de Montegancedo s/n, Boadilla del Monte, Madrid, Spain
E-mail: rgarcia@fi.upm.es

Dieter Fensel
University of Innsbruck
Technikerstr. 21a, 6020 Innsbruck, Austria
E-mail: dieter.fensel@sti2.at

Grigoris Antoniou
FORTH-ICS and University of Crete
71110 Heraklion, Crete, Greece
E-mail: antoniou@ics.forth.gr

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-25952-4 e-ISBN 978-3-642-25953-1
DOI 10.1007/978-3-642-25953-1
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: Applied for

CR Subject Classification (1998): H.4, H.3.3, C.2, H.5, J.4, I.2.4, D.2

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This volume contains a selection of the best papers from the workshops that were held at the 8th Extended Semantic Web Conference (ESWC 2011), held during May 29–30, 2011, in Heraklion, Greece. ESWC conferences present the latest results in research and applications of Semantic Web technologies, and the workshops co-located with them are distinguished meeting points for discussing ongoing work and the latest ideas in the Semantic Web field.

The Workshop Chairs selected 11 workshops, focusing on specific research issues related to the Semantic Web. The workshops were organized by international experts in the respective fields and each workshop set up an international Program Committee that carefully selected the workshop contributions.

- First International Workshop on eLearning Approaches for the Linked Data Age (Linked Learning 2011)
- First Workshop on High-Performance Computing for the Semantic Web (HPCSW 2011)
- Third International Workshop on Inductive Reasoning and Machine Learning for the Semantic Web (IRMLES 2011)
- First Workshop on Making Sense of Microposts (#MSM2011)
- First Workshop on Ontology and Semantic Web for Manufacturing (OSEMA 2011)
- First Workshop on Question Answering over Linked Data (QALD-1)
- 4th International Workshop on REsource Discovery (RED 2011)
- 6th International Workshop on Semantic Business Process Management (BPM 2011)
- First Workshop on Semantic Publications (SePublica 2011)
- First Workshop on Semantics in Governance and Policy Modelling (SGPM 2011)
- First International Workshop on User Profile Data on the Social Semantic Web (UWeb 2011)

The workshop selection reflects the international and interdisciplinary scope of the ESWC conferences, covering:

- Advances in Semantic Web technologies through cross-fertilization with other fields: parallel data processing and high-performance computing (HPCSW 2011), knowledge discovery and machine learning (IRMLES 2011), question answering (QALD-1), and resource discovery (RED 2011)
- Consolidation of the social Semantic Web through micropost (#MSM2011) and user data (UWeb 2011) exploitation.

- Application of Semantic Web technologies to other fields: eLearning (Linked Learning 2011), Business Process Modelling (BPM 2011), and Governance and Policy Modelling (SGPM 2011)
- Industrial application of Semantic Web technologies: manufacturing (OS-EMA 2011) and publishing (SePublica 2011).

From the 75 papers that were accepted for these workshops, a selection has been included in this volume. Each workshop Organizing Committee evaluated the papers accepted in their workshop to propose those to be included in this volume. Then, the authors of each proposed paper improved their papers, taking into account the comments and feedback obtained during the conference. Finally, the ESWC Workshop, Program Committee and Conference Chairs made the final decision on inclusion, based on the paper improvements from the workshop version. As a result, 22 papers were selected for this volume.

Special thanks go to all the workshop organizers and their respective Program Committees, who contributed to making ESWC 2011 a real success. We would also like to thank the ESWC 2011 Organizing Committee and specially the local organizers and the Program Chairs for supporting the day-to-day operation of the workshops and the selection of papers, respectively.

September 2011

Raúl García-Castro
Dieter Fensel
Grigoris Antoniou

Organization

Organizing Committee

General Chair

Grigoris Antoniou	FORTH-ICS and University of Crete, Greece

Program Chairs

Marko Grobelnik	Jozef Stefan Institute, Slovenia
Elena Simperl	Karlsruhe Institute of Technology, Germany

News from Front Coordinators

Lyndon Nixon	STI International, Austria
Alexander Wahler	STI International, Austria

Poster and Demo Chairs

Bijan Parsia	University of Manchester, UK
Dimitris Plexousakis	FORTH-ICS and University of Crete, Greece

Workshop Chairs

Raúl García-Castro	Universidad Politécnica de Madrid, Spain
Dieter Fensel	University of Innsbruck, Austria

Tutorials Chair

Manolis Koubarakis	University of Athens, Greece

PhD Symposium Chairs

Jeff Pan	University of Aberdeen, UK
Pieter De Leenheer	VU Amsterdam, The Netherlands

Semantic Technologies Coordinators

Matthew Rowe	The Open University, UK
Sofia Angeletou	The Open University, UK

Proceedings Chair

Antonis Bikakis	University of Luxembourg, Luxembourg

Sponsorship Chair

Anna Fensel	FTW, Austria

Publicity Chair

Lejla Ibralic-Halilovic	STI, Austria

Panel Chairs

John Domingue	The Open University, UK
Asunción Gómez-Pérez	Universidad Politécnica de Madrid, Spain

Treasurer

Alexander Wahler	STI International, Austria

Local Organization and Conference Administration	STI International, Austria

First International Workshop on eLearning Approaches for the Linked Data Age

Organizing Committee

Stefan Dietze	The Open University, UK
Mathieu d'Aquin	The Open University, UK
Dragan Gasevic	Athabasca University and Simon Fraser University, Canada
Miguel-Angel Sicilia	University of Alcalá, Spain

Program Committee

Lora Aroyo
Soeren Auer
Panagiotis Bamidis
Charalampos Bratsas
Dan Brickley
Vania Dimitrova
John Domingue
Nikolas Dovrolis
Marek Hatala
Jelena Jovanovic
Eleni Kaldoudi
Tomi Kauppinen
Carsten Kessler
Effie Lai-Chong Law
Nikos Manouselis
Dave Millard
Evangelia Mitsopoulou
Wolfgang Nejdl
Mikael Nilsson
Carlos Pedrinaci
Davide Taibi
Vlad Tanasescu
Fridolin Wild
Martin Wolpers
Hong Qing Yu

First Workshop on High-Performance Computing for the Semantic Web

Organizing Committee

Jesse Weaver	Rensselaer Polytechnic Institute, Troy, NY, USA
Spyros Kotoulas	Vrije Universiteit Amsterdam, The Netherlands
Jacopo Urbani	Vrije Universiteit Amsterdam, The Netherlands
Eric Goodman	Sandia National Laboratories, Albuquerque, NM, USA
David Mizell	Cray, Inc., Seattle, WA, USA

Program Committee

Aidan Hogan
Alexey Cheptsov
Axel Polleres
Cliff Joslyn
David Haglin
David Mizell
Eric Goodman
Gregory Todd Williams
Jacopo Urbani
Jans Aasman
Jason Maassen
Jesse Weaver
Matthias Assel
Robert Adolph
Sinan al-Saffar
Spyros Kotoulas

Third International Workshop on Inductive Reasoning and Machine Learning for the Semantic Web

Organizing Committee

Claudia d'Amato	University of Bari, Italy
Nicola Fanizzi	University of Bari, Italy
Blaz Fortuna	Jozef Stefan Institute, Slovenia
Agnieszka Ławrynowicz	Poznan University of Technology, Poland
Vojtěch Svátek	University of Economics, Prague, Czech Republic

Program Committee

Bettina Berendt
Stephan Bloehdorn
Ross D. King
Jens Lehmann
Thomas Lukasiewicz
Matthias Nickles
Achim Rettinger
Thomas Scharrenbach
Baris Sertkaya
Steffen Staab
Volker Tresp
Joaquin Vanschoren

First Workshop on Making Sense of Microposts

Organizing Committee

Matthew Rowe	KMi, The Open University, UK
Milan Stankovic	Hypios/University Paris-Sorbonne, France
Aba-Sah Dadzie	University of Sheffield, UK
Mariann Hardey	University of Durham, UK

Program Committee

Amparo-Elizabeth Cano
Alexandre Passant
Andres Garcia-Silva
Bernhard Schandl
Claudia Wagner
Danica Radovanovic
David Beer
Elena Simperl
Eric T. Meyer
Fabien Gandon
Guillaume Ereteo
Harald Sack
Harith Alani
Jelena Jovanovic
Jennifer Jones
John Breslin
Jon Hickman
Mischa Tuffield
Oscar Corcho
Pablo Mendes
Philipe Laublet
Sofia Angeletou
Raphael Troncy
Robert Jaeschke
Shenghui Wang
Uldis Bojars
Victoria Uren
Yves Raimond
Ziqi Zhang

First Workshop on Ontology and Semantic Web for Manufacturing

Organizing Committee

Alexander García Castro	University of Bremen, Germany
Lutz Schröder	German Research Center for Artificial Intelligence, Germany
Carlos Toro	Vicomtech Research Centre / Donostia-San Sebastian, Spain
Luis Enrique Ramos García	University of Bremen, Germany

Program Committee

Aristeidis Matsokis
Aziz Bouras
David Baxter
Dong Yang
Grubic Tonci
John Bateman

Jürgen Angele
Kristina Shea
Oliver Eck
Parisa Ghoudous
Richard Gil Herrera
Sylvere Krima
Yuh-Jen Chen

First Workshop on Question Answering over Linked Data

Organizing Committee

Christina Unger	CITEC, Bielefeld University, Germany
Philipp Cimiano	CITEC, Bielefeld University, Germany
Vanessa Lopez	KMi, Open University, UK
Enrico Motta	KMi, Open University, UK

Program Committee

Sören Auer
Abraham Bernstein
Chris Bizer
Kalina Bontcheva
Johan Bos
Tim Finin
Jorge Gracia
Gregory Grefenstette
Iryna Gurevych
Andreas Harth
Tom Heath
Bernardo Magnini
Michael Minock
Dan Moldovan
Günter Neumann
David Palfrey
Maarten de Rijke
Sebastian Rudolph
Victoria Uren
Duc Thanh Tran
Haofen Wang
Gerhard Weikum

4th International Workshop on REsource Discovery

Organizing Committee

María-Esther Vidal	Universidad Simón Bolívar, Caracas, Venezuela
Edna Ruckhaus	Universidad Simón Bolívar, Caracas, Venezuela
Zoé Lacroix	Arizona State University, USA

Program Committee

Mohammad Alrifai
José Luis Ambite
Yudith Cardinale
Vassilis Christophides
Oscar Corcho
Joyce El Haddad
Alberto Fernández
Manolis Gergatsoulis
Marlene Goncalves
Andreas Harth

H.V. Jagadish
Günter Ladwig
Maria Maleshkova
Kunal Patel
Marta Rukoz
Fatiha Saïs
Miguel-Angel Sicilia
Sherif Sakr
Hala Skaf-Moli
Dimitrios Skoutas
Maciej Zaremba

6th International Workshop on Semantic Business Process Management

Organizing Committee

Nenad Stojanovic	FZI, University of Karlsruhe, Germany
Barry Norton	Institute AIFB, Karlsruhe Institute of Technology, Germany

Program Committee

Agata Filipowska
Ana Karla Alves de Medeiros
Andreas Abecker
Carlos Pedrinaci
Darko Anicic
Florian Lautenbacher
Jan Mendling
John Domingue
Jorge Cardoso
Juhnyoung Lee
Ljiljana Stojanovic
Liliana Cabral
Markus Nüttgens
Michel Klein
Oliver Thomas
Oscar Corcho
Roxana Belecheanu
Tommaso Di Noia
Witold Abramowicz
York Sure
Yuxiao Zha

First Workshop on Semantic Publications

Organizing Committee

Alexander García Castro	University of Bremen, Germany
Christoph Lange	Jacobs University Bremen, Germany
Anita de Waard	Elsevier, The Netherlands
Evan Sandhaus	The New York Times Company, USA

Program Committee

Christopher Baker
Paolo Ciccarese
Tim Clark
Oscar Corcho
Joseph Corneli
Stéphane Corlosquet
Michael Dreusicke
Henrik Eriksson
Leyla Jael García Castro
Benjamin Good
Tudor Groza
Michael Kohlhase
Sebastian Kruk
Thomas Kurz
Steve Pettifer
Matthias Samwald
Jodi Schneider
Dagobert Soergel
Robert Stevens

First Workshop on Semantics in Governance and Policy Modelling

Organizing Committee

Yannis Charalabidis	University of the Aegean, Greece
Fenareti Lampathaki	National Technical University of Athens, Greece
Gianluca Misuraca	JRC - European Commission, IPTS, Spain

Program Committee

Dimitris Askounis
Enrico Ferro
Marko Grobelnik
Euripidis Loukis
Michele Missikoff
David Osimo
Maria Wimmer

First International Workshop on User Profile Data on the Social Semantic Web

Organizing Committee

Fabian Abel	WIS, Delft University of Technology, The Netherlands
Qi Gao	WIS, Delft University of Technology, The Netherlands
Eelco Herder	L3S Research Center, Leibniz University Hannover, Germany

Geert-Jan Houben	WIS, Delft University of Technology, The Netherlands
Daniel Olmedilla	Telefonica R&D, Spain
Alexandre Passant	DERI, National University of Ireland, Ireland

Program Committee

Lora Aroyo
Shlomo Berkovsky
Federica Cena
Florian Daniel
Juri Luca De Coi
Pasquale De Meo
Vania Dimitrova
Fabien Gandon
Dominikus Heckmann
Seigneur Jean-Marc
Lalana Kagal
Daniel Krause
Milos Kravcik
Philipp Kaerger
Erwin Leonardi
Javier Lopez
Axel Polleres
Matthew Rowe
Simon Schenk
Daniel Schwabe
Milan Stankovic
Krishnaprasad Thirunarayan
Alessandra Toninelli
Mischa Tuffield
Claudia Wagner
Rob Warren

Sponsoring Institutions

Platinum Sponsors

Gold Sponsors

Silver Sponsors

Best Paper Award Sponsors

FORTH-ICS

LarKC

LOD2

Creating Knowledge out of Interlinked Data

Video Recording Sponsors

PASCAL2

Pattern Analysis, Statistical Modelling and Computational Learning

videolectures.net

exchange ideas & share knowledge

Table of Contents

1st International Workshop on eLearning Approaches for the Linked Data Age

1st Workshop on High-Performance Computing for the Semantic Web

3rd International Workshop on Inductive Reasoning and Machine Learning for the Semantic Web

1st Workshop on Making Sense of Microposts

1st Workshop on Ontology and Semantic Web for Manufacturing

1st Workshop on Question Answering over Linked Data

4th International Workshop on REsource Discovery

6th International Workshop on Semantic Business Process Management

1st Workshop on Semantic Publication

1st Workshop on Semantics in Governance and Policy Modelling

1st International Workshop on User Profile Data on the Social Semantic Web

Linking Knowledge for Simulation Learning

Irene Celino and Daniele Dell'Aglio

CEFRIEL – ICT Institute, Politecnico of Milano
via Fucini 2, 20133 Milano, Italy
{irene.celino,daniele.dellaglio}@cefriel.it

Abstract. Simulation Learning is a frequent practice to conduct near-real, immersive and engaging training sessions. AI Planning and Scheduling systems are used to automatically create and supervise learning sessions; to this end, they need to manage a large amount of knowledge about the simulated situation, the learning objectives, the participants' behaviour, etc.

In this paper, we explain how Linked Data and Semantic Web technologies can help the creation and management of knowledge bases for Simulation Learning. We also present our experience in building such a knowledge base in the context of Crisis Management Training.

Keywords: Linked Data, Simulation Learning, Planning, Provenance, Semantic Web.

1 Introduction

Traditional research on Semantic Web in e-learning [23,24] are aimed at promoting interoperability between training systems, thus usually the core investigation targets are standards and schemata to describe learning objects [15,5].

Our research is focused on a different kind of e-learning system, i.e. Simulation Training to improve soft skills [1]. In this context, not only it is needed to describe learning objects, but also to fully plan simulation sessions; those sessions should be interactive and engaging to challenge the trainees to improve their skills. Simulation Learning systems generally re-create a near-real environment for training sessions, in which learners are subject to stimuli: they have to learn how to deal with the simulated situation and how to react to it.. Such simulations need to be effective and engaging, so that the learners do not simply memorise notions about the specific matter, question or theme, but they actively and permanently acquire skills, practice and knowledge.

The scenario production is therefore the core and critical activity when building a Simulation Learning system. Knowledge technologies are needed to model and manage all the required information, often generated and managed by different and independent sources: scenario descriptions, events and stimuli for the trainees, storyboards for the learning sessions, multimedia assets, supporting documents and guidelines, trainees description and behaviour/decisions, learning session monitoring, etc. Such a wealth of information makes the Simulation Learning a knowledge-intensive context, which requires smart solutions.

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 1–15, 2012.

We decided to adopt Linked Data and Semantic Web technologies to address the requirements of Simulation Learning. The knowledge diversity and scale calls for a solution which provides interlinking between different datasets while preserving possibly independent information sources; moreover, the knowledge coherence and consistency must be assured to guarantee the significance, meaningfulness and correctness of simulation scenarios and storyboards presented to trainees.

In this paper, we present our current effort in exploiting Linked Data and Semantic Web technologies to build a Knowledge Base for a Simulation Learning environment. We explain why we believe that the selected technologies not only offer a suitable means to knowledge representation and management, but they are specifically required to address the challenges of such an environment.

Section 2 introduces the basic concepts of Simulation Learning systems and a concrete scenario in Crisis Management Training; Section 3 details our exploration in the use of Linked Data and Semantic Web to build a Simulation Learning Knowledge Base illustrating the gained benefits; Section 4 specifies our modelling choices, while Section 5 suggests that such modelling could benefit from provenance tracking; finally, Section 6 concludes the paper.

2 Simulation Learning

Learning should be relevant to people's workplace and lives: learning content should be truly understood, remembered and applied to actual practices. Only in this way, by actively engaging participants in experiential training, learners can apply their knowledge and learn the best practices [1]; more and more often, indeed, it is not enough to read information and listen to a frontal lecture.

In this section, we introduce the theme of Simulation Learning for Decision-making, we draw a generic architecture of a system to support Simulation Learning and we describe a concrete scenario that we will use throughout the paper to exemplify our approach.

2.1 Simulation for Decision-Making

Training plays an important function in the preparation of professional practitioners. Currently, there are two main modalities for such training: table-top exercises and real-world simulations. *Table-top exercises* are low cost and can be easily and frequently organised. However, they cannot create a believable atmosphere of stress and confusion, which is prevailing in real-life situations and is crucial to the training of timely and effective decision making. On the other hand, training through *simulation exercises* on the field can be very effective [6], but it is considerably more expensive, it can require specialist equipment and it can be difficult to organise.

Simulation exercises require an Exercise Director (or *trainer*) who plays a key role in every form of exercise: the trainer has access to the whole exercise programme, ensures that it proceeds according to a plan, often feeds information to the "players" (the *trainees*) to let them make informed decisions in response (verbally or by written messages). Sometimes information fed to the trainees is timed in advance at pre-set intervals, regardless of the previous responses. However, flexibility allows a trainer to use judgement and experience in timing the inputs: his/her role should be aimed to facilitate rather than orchestrate the exercise, thus intervention should be minimal and trainees should be given time to recognise and correct problems. Nevertheless, usually it is up to the trainer to decide, for example, how much advice to give to trainees.

2.2 Architecture of a Simulation Learning System

The architecture of a Simulation Learning System is depicted in Figure 1. In the picture, we can identify the involved actors, which are the *trainees* – the learning participants engaged in the simulation – and the *trainer* – who activates the exercise and monitors the progress of actions during the training session.

The figure also shows the four main modules of such an architecture, the first three following the usual AI sense-think-act cycle:

- *Behaviour Sensing*: this module is aimed to create and update a model of each trainee from sensors information (e.g. heart rate, blood pressure, respiration); the model represents trainee's future and actual behaviour and provides indications on how to personalise the training path.
- *Simulation Planning*: this module is aimed to create and simulate a training scenario and its evolution, by combining the information in the behavioural model with knowledge about the learning scenarios; the output of this module is the actual simulation storyboard presented to the trainees.
- *Learning Delivery*: this module is aimed to effectively represent the simulation storyboard in the learning environment, including the rendering of audio-video inputs or Non-Player Characters (NPC, cf. Section 4.3).
- *Simulation Learning Environment*: this is the "place" where the training is conducted; the location can be a physical room or a virtual environment where the trainees interact and receive stimuli during a learning session.

The core of such system is therefore the Simulation Planning module, which contains the basic engine for creating active exercises for classes of trainees. The module is responsible for deciding which stimuli are sent to trainees and how they should be coordinated to create a meaningful and effective lesson plan. In broad terms, it is responsible for allocating over time the set of lesson stimuli indexed according to differences in presentation media, emotional characterization, personalization needs, etc.

2.3 Crisis Management Training Scenario

There is increasing recognition for the need to train non-technical skills like control and decision making for Crisis Management in national emergencies, high-reliability industries, as well as in industrial workplaces [20,22]. In the happening

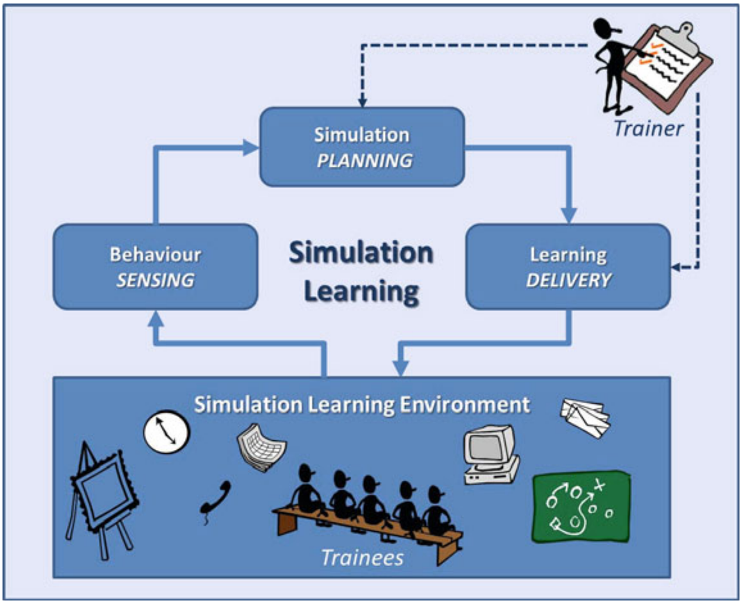


Fig. 1. High-level architecture of a Simulation Learning System (from the classical sense-think-act cycle of AI)

of a catastrophic event, it is human behaviour – and often human behaviour alone – that determines the speed and efficacy of the crisis management effects [17].

The Pandora project[1] aims to provide a framework to bridge the gap between table-top exercises and real-world simulation exercises for Crisis Management, providing a near-real training environment at affordable cost. Its training system captures the good practice tenets of experiential learning but with greater efficiency and focuses on real, rather than abstract learning environments. The effective use of integrated ICT reduces the high dependence upon the trainer that is currently required to deliver exercises. Moreover, the Pandora framework supports the measurement and performance assessment of Crisis Managers, the key decision makers participating in a training exercise event as trainees.

As such, Pandora is developing an enabling technology to simulate believable dynamic elements of an entire disaster environment by emulating a crisis room (the Simulation Learning Environment). In this context, we are developing a Knowledge Base that makes use of Linked Data and Semantic Web technologies to model and interlink the pieces of data needed in the training simulation sessions. In the rest of the paper, we will use the Crisis Management scenario to exemplify our approach.

[1] Cf. http://www.pandoraproject.eu/

3 Our Simulation Learning Linked Knowledge Base

Within a Simulation Learning system, knowledge exchange plays a central role. In this section we give some details about the Simulation Planning module, focusing on the requirements, design and implementation principles of its Knowledge Base. All the technical details are related to the choices made in the Pandora framework.

3.1 Knowledge Required to Plan a Simulation

To formalize the lesson plan, it is natural to choose a basic representation from timeline-based planning [10]. A plan is represented as a set of events having a temporal duration, distributed over a time horizon and indexed according to distinct features which should be planned for. This set of events is organized inside a data structure called *Event Network*, very common in current state of the art planning technology. The Event Network is a temporal plan of multimedia communicative acts toward trainees (e.g., e-mail messages, video news from an emergency location, etc.).

The Event Network can generated by a *Simulation Planner*. This planner compiles static information into the Event Network, and then adapts the events configuration according to the actions of the trainees, thus simulating different courses of action of the world. The planner can be adapted from a generic AI Timeline-based Planning and Scheduling module [10].

The core information item elaborated by a Simulation Planner is the so-called *synchronization*. Synchronizations are the causal rules that regulate the transitions between values on the same planning feature and the synchronization of values among different planning features. In the Crisis Management scenario, synchronizations are used to influence the Crisis Managers' decisions, e.g. to generate changes in the emergency conditions.

When adopting Planning and Scheduling technologies to simulate a scenario, it is worth highlighting how a great effort and amount of time is necessary to understand the problem, capturing all its specificity, and to create a model of the relevant aspects of the domains and the problem [9]. This consideration suggests, on the one hand, the need for identifying commonalities and similarities among the different domains and problems to operate in a more systematic way and, on the other hand, the opportunity to exploit Semantic Web technologies to ease and support the knowledge modelling task.

For those reasons, we have built a *Knowledge Base* with Linked Data and Semantic Web technologies. This KB is a central component in the Simulation Learning system, responsible for collecting and maintaining the "knowledge" about scenarios and training sessions. As such, the KB is the core information source for the simulation: it contains all the knowledge required by the Simulation Planner to "orchestrate" the events during the training sessions. All the causality in a simulation domain is modelled and stored in the KB; this knowledge is then converted by the Simulation Planner into the suitable data structures to synthesize the Event Network configurations for the lesson plan goals.

3.2 Requirements for the Knowledge Base

The Knowledge Base [8] was carefully designed to fulfil a pressing requirement: containing and managing all the knowledge needed to model and run the simulation scenarios, the training events, the trainees' behaviour, the time sequence, and so on.

To fulfil such a requirement, the KB must reuse pre-existing information (e.g., in the Crisis Management scenario, training procedures, emergency management guidelines) and, in the meantime, it must allow for customization and diversification of training knowledge (e.g., emergency policies and legislation change from country to country). Furthermore, since most of the related information can be pre-existing in a variety of formats, the KB must able to gather information from heterogeneous sources (e.g., location data from geographic datasets, audio and video inputs from multimedia archives, participants profiles) and to synthetize and interlink such knowledge into a coherent base.

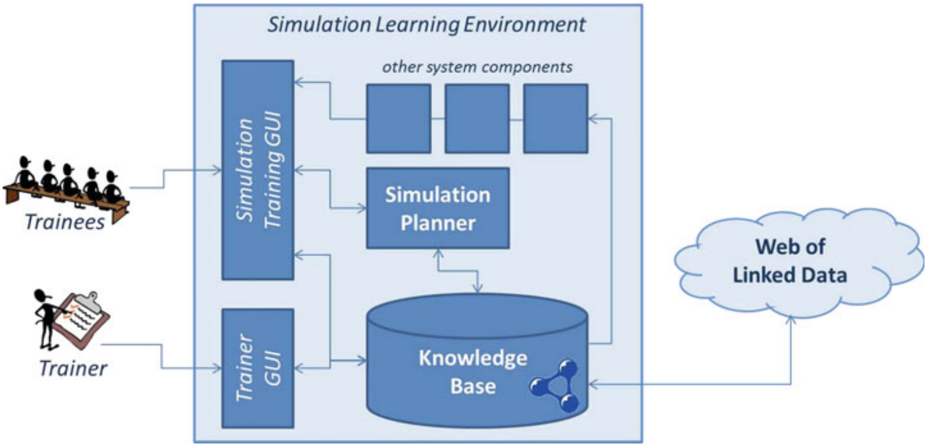


Fig. 2. Role of the Knowledge Base in a Simulation Learning Environment

The role of the KB in the Simulation Learning Environment and its interactions with other components is depicted in Figure 2:

- The KB is "initialized" by the trainer who *models the simulation scenarios* and the training path alternative options;
- It is accessed by the Simulation Planner that needs to understand what *"events" should be triggered* and presented to the trainees during the learning sessions;
- It is also accessed by other system components that need to get/give *information about the training session* and the knowledge exchanged during or after its delivery (cf. Section 4);

- It is used to record the events and decisions taken during each training session, in order to enable the semi-automatically *creation of an individual trainee debriefing report* at the end of the training session.

To cope with such challenges, we adopted Linked Data and Semantic Web technologies for the design and development of our Knowledge Base.

3.3 Benefits from the Adoption of Linked Data

The choice of Linked Data and Semantic Web technologies in our KB is motivated by the need for an easy access, (re)use and integration of data and knowledge [14].

The *ease of access* to the KB is implicit in the use of Web technologies, which represent a mature and established technology stack. Following the Linked Data principles [2], we provide a standard access means to the data and knowledge stored in the KB. Moreover, Linked Data and Semantic Web facilitate and enable an entity-centric design of Web APIs: in our implementation, on top of the KB, we have developed a RESTful service[2] with specific methods to get details about certain entities on the basis of the concepts (entity types) defined in the KB ontologies and models (cf. Section 4). The RESTful service is also employed to abstract from the physical location of data, as explained further on.

The *reuse of pre-existing datasets* is also enabled by our technological choice. Several useful data sources are already present on the Web of Data and, thus, immediately exploitable by the KB. For example, in the Crisis Management scenario, environment characteristics of crisis settings are retrieved from GeoNames[3], the geographical database containing over 10 million geographical names, 7.5 million unique features, 2.8 million populated places and 5.5 million alternate names. For example, a scenario about a river flood or a earthquake benefits from the retrieval of localized information from GeoNames. As a pragmatic solution, we are "caching" the relevant features from GeoNames locally to the KB. However, the reuse of GeoNames URIs constitutes a link to the remote dataset and allows for further knowledge retrieval. Specifically for the UK, geographic data come also from OrdnanceSurvey[4], a government mapping initiative that also releases open data. In the same way, we can connect the KB to other knowledge bases like Freebase[5] or DBpedia[6] [4] to get information on a number of general-purpose topics and entities. The linkage to the latter sources is still in progress.

But this re-usability benefit applies also to the knowledge explicitly modelled for domain-specific learning scenarios: the choice of RDF to encode the data and of RDFS/OWL to model their structure pays, since those data are partially published on the open Web, thus enriching the Web of Linked Data

[2] Cf. `http://pandoratest01.xlab.si:8080/pandora-ckb/`

[3] Cf. `http://www.geonames.org/`

[4] Cf. `http://www.ordnancesurvey.co.uk/oswebsite/products/os-opendata.html`

[5] Cf. `http://freebase.com/`

[6] Cf. `http://dbpedia.org/`

and becoming available for other Simulation Learning systems or for different tools. To this end, in our Crisis Management scenario, we decided to store the schemata and data generated by Pandora components natively as RDF triples in the KB; the knowledge coming from pre-existing sources in different formats (e.g., taxonomies, spreadsheets, guidelines) have been converted – manually or, whenever possible, semi-automatically – to a structured RDF format. The benefits of this approach are: the general Crisis Management knowledge is available to the whole community; the simulation scenarios can be reused by any installation of the training system; the further enhancements and extensions of the core knowledge are immediately "reflected" in all systems that make use of our KB.

The *ease of integration* comes from the native interlinking capability of Linked Data technologies. RDF provides the basic mechanism to specify the existence and meaning of connections between items through RDF links [3]. In other words, through the adoption of RDF, we not only give a structure to the data stored in the KB, but we also interlink the entities described by such data. Moreover, the links drawn between knowledge items are typed, thus conveying the "semantics" of such relationships and enabling the inference of additional knowledge. The information sources of the KB can be maintained and evolve over time in an independent way, but, in the meantime, can be connected via the Linked Data lightweight integration means.

The KB contains different (although interlinked) datasets, which also require diverse confidentiality/security levels for management and access. To this end, the KB can be designed as a set of federated RDF stores. The shared knowledge (e.g. general Crisis Management information, basic scenarios) could be "centralised", to let all training system instances access and use it, while the installation-specific knowledge (e.g., detailed or customized scenarios, trainees information, personalizations) could be managed in a local triple store, not accessible from outside the system (see Figure 3). The RESTful service on top of the KB, as explained earlier, can provide a uniform access to the KB and hide to other accessing components the existence of the various "realms" of distinct Linked Data sources.

In the Pandora project, the choice was made to have the KB as a unique triple store with all the integrated knowledge. In fact, while the federated design was recognized as an interesting and meaningful solution, the distinction between "local" and "global" knowledge within the KB was very hard if not impossible to define. Indeed, some intrinsically non-confidential knowledge like the simulation scenario descriptions can represent a valuable information asset for a training company, which can be unwilling to make this knowledge freely available as Linked Open Data to competitors. Conversely, while the possibility to introduce customizations and personalization in the simulation scenarios is a very appreciated feature, making modifications to a global copy of that kind of knowledge would impact on all Pandora installation using that knowledge. Summing up, while Linked Data technologies are acknowledged as a good technological choice, Linked Open Data are not always applicable because of business or pragmatic reasons.

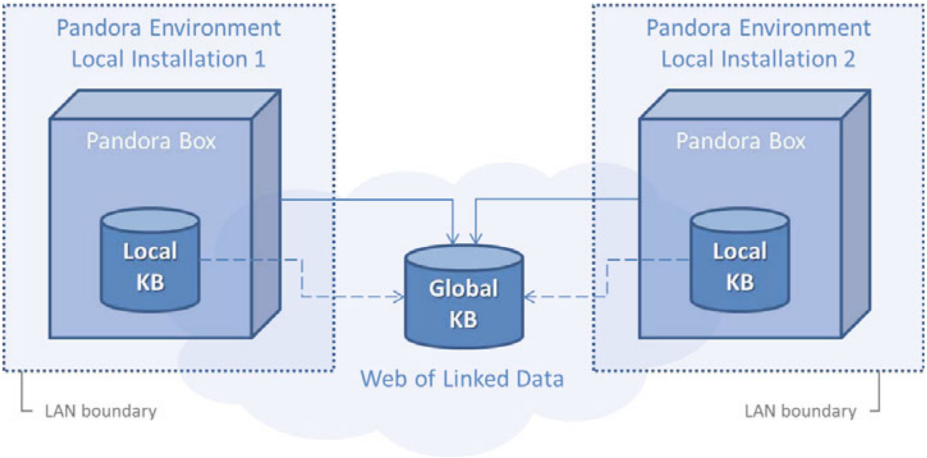


Fig. 3. The KB as federation of different triple stores to preserve security and confidentiality while benefiting from interlinking

Finally, the adoption of Semantic Web technologies in the form of ontologies and rules provides a further gain, since we can exploit *reasoning and inference* for knowledge creation and consistency checking, as explained in next section.

4 Modelling and Retrieval in Our Knowledge Base

As previously mentioned, our Knowledge Base manages several different and interlinked types of information. In this section, we introduce three "families" of data included in the KB and explain their modelling choices. We also illustrate their use in the Crisis Management training scenario within the Pandora Integrated Environment.

4.1 User Modelling

As introduced in Section 2.2, a Behaviour Sensing module is devoted to the "detection" of trainees' performance in order to create individual models that help in tailoring the learning strategy of each participant to the simulation. Prior to the training session, dedicated psychological tests and physiological assessment at rest (e.g., through a Holter that measures the heart rate activity at rest) are used to measure some relevant variables (like personality traits, leadership style, background experience, self-efficacy, stress and anxiety). Those variables are then updated during the training session, through self-assessment measurements (i.e., asking the trainee about his performance) or through the elaboration of the row data recorded by the sensors.

Those data about trainees' behaviour are stored and updated in our KB, as instances of ontology concepts that represent the "affective factors" that influence the decision-making of the trainees. Due to the sensitivity of such information, the individual performances of the trainees are modelled in RDF and stored in the "local" triple store (cf. Figure 3) for apparent privacy reasons. We are also investigating the possibility to exploit Named Graphs [7] for access control: if the training session recordings are "stored" in the KB as separated named graphs, a named graph-aware access control component could grant admission to the allowed users (e.g., the trainer) and could deny the access of the malicious or occasional users (e.g., the other trainees).

In the specific scenario of the Pandora Integrated Environment, the learning sessions are targeted to the training of Crisis Managers. Therefore, the KB stores and manages also a set of specific information about them.

The Crisis Managers are the so-called Gold Commanders, who are responsible for the strategic development of responses to crisis situations. The trainee group is usually composed of the representatives of the "command team", i.e. the core agencies involved in the strategic Crisis Management (e.g., police, local authority, fire brigade, ambulance); sometimes, other trainees can come from other utility companies (e.g. electricity, road transportation, environmental agency).

In our KB, therefore, we modelled the basic knowledge about those Gold Commanders by creating classes to represent the different trainees typologies. Those classes are "instantiated" per each training session, by adding the individual trainees to the KB. This lets the system record the training of each participant in relation to his/her role in the simulation; this knowledge is very precious for both the debriefing phase – when the trainer summarizes the performance results of each trainee (see also below) – and for a general analysis and mining of the achieved objectives and learning needs of the different agencies.

The initial version of the user modelling is part of the Pandora Ontology[7].

4.2 Training Simulation Modelling

The core module of the simulation learning system is the Simulation Planning (cf. Section 2.2). Our KB therefore must be able to manage the knowledge required for the planning, in terms of the basic entities used by AI Planning Applications based on Timeline Representations.

In literature, several attempts tried to formalize the semantics of planners [19,12]. However, those approaches, on the one hand, tried to specify a generic planning ontology and, on the other hand, were specifically tailored to some application domains.

Building on their experience, we decided to make our own formalization to encompass the family of techniques known under the name of Timeline-based Planning and Scheduling. In fact, current AI planning literature shows that timeline-based planning can be an effective alternative to classical planning for complex domains which require the use of both temporal reasoning and scheduling features [10]. Moreover, our modelling aims to become the foundation for the

[7] Cf. `http://swa.cefriel.it/ontologies/pandora`

investigation on the interplay between Semantic Web Technologies and Planning and Scheduling research [8]; Semantic Web knowledge bases, in fact, can represent a good alternative to the current domain modelling in the planning area, which encompasses a multitude of custom and not interoperable languages.

Our modelling is formalized in a Timeline-based Planning Ontology[8]. As in classical Control Theory, the planning problem is modelled by identifying a set of relevant features (called *components*) which are the primitive entities for knowledge modelling. Components represent logical or physical subsystems whose properties may vary in time; in the simulation learning, components are either trainees behavioural traits or learning scenario variables. Their *temporal evolutions* is controlled by the planner to obtain a desired *behaviour*. Therefore, our ontology includes a set of *time functions* that describe the evolution over temporal intervals. The evolution is modelled by *events* happening on modelled components. To this end, a set of *planning rules* (or synchronizations) specifies what events can be triggered to modify these evolutions. The task of the Simulation Planner is to find a sequence of events that brings the system entities into a desired final state.

The core concept of the Timeline-based Planning Ontology is therefore the planning rule: each rule puts in relation a "reference" event – which is the potential *cause* of some phenomena in the simulation – with a "target" event – which is the possible *consequence* –, under a set of conditions called *rule relations*. We modelled such conditions as SPARQL FILTER or LET clauses[9]; therefore, we reused the modelling of such clauses and functions included in the SPIN Modeling Vocabulary [16] and extended it with regards to temporal conditions.

At learning *design time* – i.e. prior to the simulation sessions –, the trainer has to model the possible training scenarios, by instantiating in the KB the ontology concepts, in particular the planning rules and the related events. The choice of Linked Data and Semantic Web technologies for our modelling is not only useful for reusing and exploiting pre-existing knowledge. In this case, we can also exploit the semantics of such ontology for the *consistency checking* of the simulation scenarios: by automatic means, we can check if all the planning rules are satisfiable, if they represent possible "states" of the world simulated during the sessions, if all the events can happen under opportune conditions, and so on.

At *run-time* – i.e. during the simulation learning sessions –, all the events and decisions taken by the trainees during their learning are recorded in the KB. The KB is therefore used by the Simulation Planner to create and update the simulation plan. SPARQL-based querying is used to perform the knowledge retrieval required in this step: based on the actual recorded events, only the admissible planning rules are returned to let the planner decide what events to trigger.

[8] Cf. `http://swa.cefriel.it/ontologies/tplanning`

[9] The SPARQL LET clause is defined in some implementations, like the Jena Semantic Web Framework `http://openjena.org/`

After the learning session, at *debriefing time*, the recording of trainees' behaviour and decision-taking is exploited to summarize the session progress. Also in this case, SPARQL-based querying on the KB is exploited to retrieve all the events and situations that involved each trainee; this knowledge is immediately at disposal of the trainer to produce a debriefing report for each participant and can be used to highlight personal performance, achieved training goals and attention points for improvement or further training.

4.3 Asset Modelling

The Learning Delivery module (cf. Figure 1) takes as input the simulation plan and "execute" it by sending the opportune stimuli to the trainee. To do this, it needs to recreate the actual simulation conditions, by pretending a near-real situation. For example, in the Crisis Management training scenario, the participants must be solicited by phone calls, mail, news, videos, etc. that give them updates on the evolution of the emergency. To this end, the Learning Delivery module manages two types of "learning objects" that are described in the KB.

The first type of simulation objects consists in *audio and video assets*, which give information to the trainees about what happens outside the simulation room. In the Pandora scenario, those assets are pre-canned recording of simulated video news or audio inputs – like phone calls from the crisis setting – which are used to put pressure on the trainees and, in the meantime, to give them further inputs on which they must base their decisions. To model such assets, it is possible to re-use existing learning objects modelling, such as [5,21]. In the Pandora project, we are still in the process of selecting the most suitable modelling for our purpose.

There is a second type of stimuli for the simulation trainees. Since the sensing system records the "performance" of each participant also in terms of stress and anxiety, the simulation can be adapted to the specific conditions and deliver tailored inputs for the individual trainees. For example, if the purpose is to augment the pressure on a participant, the input could be made more dramatic. To this end, the Learning Delivery module makes use of *Non-Player Characters* (NPC): in games terminology, elements that act as a fictional agents and that are animated and controlled by the system. Those NPCs simulate additional actors from outside the learning environment and are used to deliver information to the trainees.

Our KB, therefore, includes also the modelling of NPC descriptions, in terms of their role in the simulation, their basic characteristics (e.g. gender, ethnicity, disability), their profiles (expertise, experience, emotional type, communication skills, etc.), their multimedia rendering mode (from the simplest text representation to fully rendered 3D avatar), etc. For this modelling, Linked Data are exploited for the reuse of pre-existing descriptions and Semantic Web technologies are leveraged to retrieve and select the most suitable NPC to simulate a desired stress or anxiety situation.

5 Exploiting Provenance Tracking

As detailed in the previous section, our Linked Knowledge Base is used to manage the knowledge required to produce simulation-based learning sessions. Simulation Learning can be seen as a special case of the Open Provenance Model (OPM) [18]. The sessions are our main *process*, the trainees, as well as the simulated external characters, are our *agents* and the events and the decisions taken by the trainees are the *artifacts* of the learning sessions.

Our investigation has recently focused on the definition of the suitable OPM Profile for Simulation Learning systems; specifically, we have mapped[10] our Timeline-based Planning Ontology to the Open Provenance Model Vocabulary Specification [25]. Hereafter we give some hints on how we can build on the Open Provenance Model and why it is useful.

The provenance tracking in simulation learning can be done at two levels: at design time – when the learning scenarios are modelled in the KB with their possible planning rules –, and after the learning sessions – when the results of the simulations are analysed.

At design time, provenance can be used to trace the cause-consequence chains between the simulation events. As explained in Section 4.2, planning rules are used to model the admissible transitions between events in the simulation; the *completion and inference rules* defined in OPM [18] can be exploited for the consistency checking of the simulation modelling. On the one hand, those rules can help in refining the modelling, by eliminating useless entities, combining eventual repetitions and introducing missing entities; on the other hand, OPM rules can help in examining the possible decision-trees (i.e., the possible alternative planning options) to identify unreachable states or decision bottlenecks.

We have identified a set of *causality checking rules*, expressible in terms of OPMV and of our planning ontology, and we have implemented them as a set of SPARQL 1.1 queries [13] on the simulation scenario models. Those rules represent a valuable support means for the scenario modellers: when the simulations are described in terms of the events and the planning rules to trigger the events, it can be difficult for the modeller to identify potential problems. It is worth noting that the causality checks we have identified are different and complementary to the planning consistency checks usually employed in planning systems [11].

After the learning sessions, the simulation records can be analysed to understand and synthetise the learning outcomes. Tracking the provenance of trainees' decisions and mining the most popular causal chains across several sessions delivery can be of great help for identifying learning needs, common behaviours (as well as common trainees' mistakes), wide-spread procedures, etc. This information can become of considerable importance: on the one hand, to improve the learning simulations and better address learners requirements and, on the other hand, to better study and interpret learning outcomes for individual participants or for entire classes of trainees.

[10] Cf. http://swa.cefriel.it/ontologies/causality-provenance.html

6 Conclusions

In this paper, we presented our approach and experience in building a Linked Knowledge Base to support Simulation Learning systems. We introduced the general architecture of such a system together with a concrete scenario in Crisis Management training; we illustrated the benefits of the use of Linked Data and Semantic Web technologies and we summarised our modelling choices. We also suggested the introduction of provenance tracking, to further enrich and better analyse the contents of a Knowledge Base for Simulation Learning.

Our approach is being integrated in the Pandora Environment, which, in the second half of 2011, will be tested at the UK Emergency Planning College in their "Emergency Response and Recovery" training courses.

Acknowledgments. This research is partially funded by the EU PANDORA project (FP7-ICT-2007-1-225387). We would like to thank the project partner for their collaboration.

References

1. Aldrich, C.: Simulations and the Future of Learning: An Innovative (and Perhaps Revolutionary) Approach to e-Learning, Pfeiffer (September 2003)
2. Berners-Lee, T.: Linked Data – W3C Design Issues, Architectural and philosophical points (2006), http://www.w3.org/DesignIssues/LinkedData.html
3. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data – The Story So Far. International Journal on Semantic Web and Information Systems 5, 1–22 (2009)
4. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia – A Crystallization Point for the Web of Data. Journal of Web Semantics: Science, Services and Agents on the World Wide Web 7, 154–165 (2009)
5. Brase, J., Nejdl, W.: Ontologies and Metadata for eLearning. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies International Handbooks on Information Systems, pp. 555–574. Springer, Heidelberg (2004)
6. Caird-Daley, A., Harris, D., Bessell, K., Lowe, M.: Training Decision Making using Serious Games. Tech. rep. Human Factors Integration Defence Technology Centre (2007)
7. Carroll, J.J., Bizer, C., Hayes, P., Stickler, P.: Named graphs, provenance and trust. In: WWW 2005: Proceedings of the 14th International Conference on World Wide Web, pp. 613–622. ACM (2005)
8. Celino, I., Dell'Aglio, D., De Benedictis, R., Grilli, S., Cesta, A.: Ontologies, rules and linked data to support Crisis Managers Training. IEEE Learning Technology Newsletter, Special Issue Semantic Web Technologies for Technology Enhanced Learning 13(1) (2011)
9. Cesta, A., Cortellessa, G., Fratini, S., Oddi, A.: Developing an End-to-End Planning Application from a Timeline Representation Framework. In: 21st Applications of Artificial Intelligence Conference (2009)
10. Cesta, A., Fratini, S.: The Timeline Representation Framework as a Planning and Scheduling Software Development Environment. In: 27th Workshop of the UK Planning and Scheduling SIG (2008)

11. Gerevini, A., Long, D.: Plan Constraints and Preferences in PDDL3. Tech. rep. R.T. 2005-08-47, Dipartimento di Elettronica per l'Automazione, Università degli Studi di Brescia (2005)
12. Gil, Y., Blythe, J.: Planet: A sharable and reusable ontology for representing plans. In: The AAAI - Workshop on Representational Issues for Real-World Planning Systems, pp. 28–33 (2000)
13. Harris, S., Seaborne, A.: SPARQL 1.1 Query Language. W3C Working Draft (2011), http://www.w3.org/TR/sparql11-query/
14. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. In: Synthesis Lectures on the Semantic Web: Theory and Technology, 1st edn., vol. 1. Morgan & Claypool (2011)
15. Hodgins, W., Duval, E.: Draft standard for learning technology - Learning Object Metadata. Tech. rep., Learning Technology Standards Committee of the IEEE. IEEE Standards Department, New York (July 2002)
16. Knublauch, H.: SPIN Modeling Vocabulary (October 20, 2009), http://spinrdf.org/spin.html
17. Lehto, M., Nah, F.: Decision-making Models and Decision Support. In: Handbook of Human Factors and Ergonomics, John Wiley & Sons, Inc., NY (2006)
18. Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., Plale, B., Simmhan, Y., Stephan, E., den Bussche, J.V.: The Open Provenance Model core specification (v1.1). Future Generation Computer Systems (2010)
19. Rajpathak, D., Motta, E.: An ontological formalization of the planning task. In: International Conference on Formal Ontology in Information Systems (FOIS 2004), pp. 305–316 (2004)
20. Sniezek, J., Wilkins, D., Wadlington, P., Baumann, M.: Training for Crisis Decision-Making: Psychological Issues and Computer-Based Solutions. Journal of Management Information Systems 18(4), 147–168 (2002)
21. Steinmetz, R., Seeberg, C.: Meta-information for Multimedia eLearning. In: Computer Science in Perspective, pp. 293–303 (2003)
22. Stern, E., Sundelius, B.: Crisis Management Europe: An Integrated Regional Research and Training Program. International Studies Perspective 3(1), 71–88 (2002)
23. Stojanovic, L., Staab, S., Studer, R.: eLearning based on the Semantic Web. In: WebNet 2001 - World Conference on the WWW and Internet, pp. 23–27 (2001)
24. Tiropanis, T., Davis, H.C., Millard, D.E., Weal, M.J.: Semantic Technologies for Learning and Teaching in the Web 2.0 Era. IEEE Intelligent Systems 24(6), 49–53 (2009)
25. Zhao, J.: Open Provenance Model Vocabulary Specification (October 2010), http://purl.org/net/opmv/ns

Generating Educational Assessment Items from Linked Open Data: The Case of DBpedia

Muriel Foulonneau

Tudor Research Centre,
29, av. John F. Kennedy
L-1855 Luxembourg, Luxembourg
muriel.foulonneau@tudor.lu

Abstract. This work uses Linked Open Data for the generation of educational assessment items. We describe the streamline to create variables and populate simple choice item models using the IMS-QTI standard. The generated items were then imported in an assessment platform. Five item models were tested. They allowed identifying the main challenges to improve the usability of Linked Data sources to support the generation of formative assessment items, in particular data quality issues and the identification of relevant sub-graphs for the generation of item variables.

Keywords: Linked Data, open data, DBpedia, eLearning, e-assessment, formative assessment, assessment item generation, data quality, IMS-QTI.

1 Introduction

Assessment takes a very important role in education. Tests are created to evaluate what students have learned in the class, to assess their level at the beginning of a cycle, to enter a prestigious university, or even to obtain a degree. More and more assessment is also praised for its contribution to the learning process through formative assessment (i.e., assessment to learn, not to measure) and/or self-assessment whereby the concept of a third party controlling the acquisition of knowledge is totally taken out of the assessment process. The role of assessment in the learning process has considerably widened. The New York Times even recently published an article entitled “To Really Learn, Quit Studying and Take a Test” [1], reporting on a study by Karpicke et al. [2] which suggests that tests are actually the most efficient knowledge acquisition method.

The development of e-assessment has been hampered by a number of obstacles, in particular the time and effort necessary to create assessment items (i.e., test questions) [3]. Therefore, automatic or semi-automatic item generation has gained attention over the last years. Item generation consists in using an item model and creating automatically or semi-automatically multiple items from that model.

The Semantic Web can provide relevant resources for the generation of assessment items because it includes models of factual knowledge and structured datasets for the generation of item model variables. Moreover, it can provide links to relevant learning resources, through the interlinking between different data sources.

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 16–27, 2012.

Using a heterogeneous factbase for supporting the learning process however raises issues related for instance to the potential disparities of data quality. We implemented a streamline to generate simple choice items from DBpedia. Our work aims at identifying the potential difficulties and the feasibility of using Linked Open Data to generate items for low stake assessment, in this case formative assessment.

We present existing approaches to the creation of item variables, the construction of the assessment item creation streamline, and the experimentation of the process to generate five sets of items.

2 Existing Work

Item generation consists in creating multiple instances of items based on an item model. The item model defines variables, i.e., the parts which change for each item generated. There are different approaches to the generation of variables, depending on the type of items under consideration.

In order to fill item variables for mathematics or science, the creation of computational models is the easiest solution. Other systems use natural language processing (NLP) to generate for instance vocabulary questions and cloze questions (fill in blanks) in language learning formative assessment exercises ([4], [5], [6]). Karamanis et al. [7] also extract questions from medical texts.

The generation of variables from structured datasets has been experimented in particular in the domain of language learning. Lin et al. [8] and Brown et al. [9] for instance generated vocabulary questions from the WordNet dataset, which is now available as RDF data on the Semantic Web. Indeed, the semantic representation of data can help extracting relevant variables. Sung et al. [10] use natural language processing to extract semantic networks from a text and then generate English comprehension items.

Linnebank et al. [11] use a domain model as the basis for the generation of entire items. This approach requires experts to elicit knowledge in specifically dedicated models. However, the knowledge happens to already exist in many data sources (e.g., scientific datasets), contributed by many different experts who would probably never gather in long modeling exercises. Those modeling exercises would have to be repeated over time, as the knowledge of different disciplines evolves. Moreover, in many domains, the classic curricula, for which models could potentially be developed and maintained by authorities, are not suitable. This is the case of professional knowledge for instance.

Given the potential complexity of the models for generating item variables, Liu [12] defines reusable components of the generation of items (including the heuristics behind the creation of math variables for instance). Our work complements this approach by including the connection to semantic datasets as sources of variables. Existing approaches to item generation usually focus on language learning [13] or mathematics and physics where variable can be created from formulae [14]. We aim to define approaches applicable in a wider range of domains (e.g., history) by reusing existing interlinked datasets.

3 Generating Item Variables from a SPARQL Endpoint

An item model includes a stem, options, and potentially auxiliary information [15]. Only the stem (i.e., the question) is mandatory. Response options are provided in the case of a multiple choice item. Auxiliary information can be a multimedia resource for instance. In some cases, other parameters can be adapted, including the feedback provided to candidates after they answer the item.

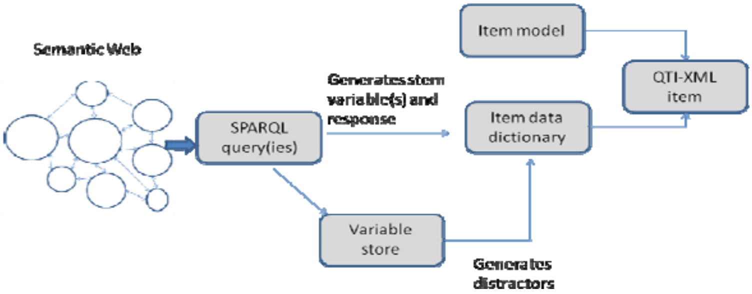


Fig. 1. Semi-automatic item generation from semantic datasets

In order to investigate the use of Linked Data as a source of assessment items, we built a streamline to generate simple choice items from a SPARQL endpoint on the Web. The item generation process is split in different steps detailed in this section. Figure 1 shows the item model represented as an item template, the queries to extract data from the Semantic Web, the generation of a set of potential variables as a variable store, the organization of all the values of variables for each item in data dictionaries, and the creation of items in QTI-XML format from the item template and item data dictionaries. These steps are detailed in this section.

3.1 Creating an IMS QTI-XML Template

In order to generate items which are portable to multiple platforms, it is necessary to format them in IMS-QTI (IMS Question & Test Interoperability Specification)[1]. IMS-QTI is the main standard used to represent assessment items [16]. It specifies metadata (as a Learning Object Metadata profile), usage data (including psychometric indicators), as well as the structure of items, tests, and tests sections. It allows representing multimedia resources in a test. IMS-QTI has an XML serialization.

No language exists for assessment item templates. We therefore used the syntax of JSON templates for an XML-QTI file (Figure 2). All variables are represented with the variable name in curly brackets. Unlike RDF and XML template languages, JSON templates can define variables for an unstructured part of text in a structured document. For instance, in Figure 2, the {prompt} variable is only defined in part of the content of the <*prompt*> XML element. Therefore, the question itself can be stored in the item model, only the relevant part of the question is represented as a variable.

[1] http://www.imsglobal.org/question/

```
<choiceInteraction responseIdentifier="RESPONSE" shuffle="false" maxChoices="1">
            <prompt>What is the capital of {prompt}?</prompt>
            <simpleChoice identifier="{responseCode1}">{responseOption1}</simpleChoice>
            <simpleChoice identifier="{responseCode2}">{responseOption2}</simpleChoice>
            <simpleChoice identifier="{responseCode3}">{responseOption3}</simpleChoice>
 </choiceInteraction>
```

Fig. 2. Extract of the QTI-XML template for a simple choice item

3.2 Collecting Structured Data from the Semantic Web

In order to generate values for the variables defined in the item template, data sources from the Semantic Web are used. The Semantic Web contains data formatted as RDF. Datasets can be interlinked in order to complement for instance the knowledge about a given resource. They can be accessed through browsing, through data dumps, or through a SPARQL interface made available by the data provider. For this experiment, we used the DBpedia SPARQL query interface (Figure 3). The query results only provide a variable store from which items can be generated. All the response options are then extracted from the variable store (Figure 1).

```
SELECT ?country ?capital
WHERE {
?c <http://dbpedia.org/property/commonName> ?country .
?c <http://dbpedia.org/property/capital> ?capital
}
LIMIT 30
```

Fig. 3. SPARQL query to generate capitals in Europe

Linked data resources are represented by URIs. However, the display of variables in an assessment item requires finding a suitable label for each concept. In the case presented on Figure 3, the *?c* variable represents the resource as identified by a URI. The <*http://dbpedia.org/property/commonName*> property allows finding a suitable label for the country. Since the range of the <*http://dbpedia.org/property/capital*> property is a literal, it is not necessary to find a distinct label.

The label is however not located in the same property in all datasets and for all resources. In the example of Figure 3, we used the property <*http://dbpedia.org/property/commonName*> which provides the capital names as literals. However, other properties, such as <*foaf:name*> are used for the same purpose. In any case, the items always need to be generated from a path in a semantic graph rather than from a single triple. This makes Linked Data of particular relevance since the datasets can complete each other.

3.3 Generating Item Distractors

The SPARQL queries aim to retrieve statements from which the stem variable and the correct answer are extracted. However, a simple or multiple choice item also needs distractors. Distractors are the incorrect answers presented as options in the items. In the case of Figure 3, the query retrieves different capitals, from which the distractors are randomly selected to generate an item. For instance, the capital of Bulgaria is Sofia. Distractors can be Bucarest and Riga.

3.4 Creating a Data Dictionary from Linked Data

The application then stores all the variables for the generated items in data dictionaries. Each item is therefore represented natively with this data dictionary. We created data dictionaries as Java objects conceived for the storage of QTI data. We also recorded the data as a JSON data dictionary.

In addition to the variables, the data dictionary includes provenance information, such as the creation date and the data source.

3.5 Generating QTI Items

QTI-XML items are then generated from the variables stored in the data dictionary and the item model formalized as a JSON template. We replaced all the variables defined in the model by the content of the data dictionary. If the stem is a picture, this can be included in the QTI-XML structure as an external link.

4 The DBpedia Experiment

In order to validate this process, we experimented the generation of assessment items for five single choice item models. We used DBpedia as the main source of variables. The item models illustrate the different difficulties which can be encountered and help assessing the usability of the Linked Data for the generation of item variables.

4.1 The Generation of Variables for Five Item Models

Q1 - What is the capital of { Azerbaijan }?

The first item model uses the query presented on Figure 3. This query uses the *http://dbpedia.org/property/* namespace, i.e., the Infobox dataset. This dataset however is not built on top of a consistent ontology. It rather transforms the properties used in Wikipedia infoboxes. Therefore, the quality of the data is a potential issue[2].

Out of 30 value pairs generated, 3 were not generated for a country (Neuenburg am Rhein, Wain, and Offenburg). For those, the capital was represented by the same literal as the country. Two distinct capitals were found for Swaziland (Mbabane, the administrative capital and Lobamba, the royal and legislative capital). The Congo is identified as a country, whereas it has been split into two distinct countries. Its capital

[2] `http://wiki.dbpedia.org/Datasets`

Leopoldville was since renamed Kinshasa. The capital of Sri Lanka is a URI, whereas the range of the capital property is usually a de facto literal. Finally the capital of Nicaragua is represented with display technical instructions “Managua right|20px”. Overall, 7 value pairs out of 30 were deemed defective.

Q2 - Which country is represented by this flag ?

```
SELECT ?flag ?country
WHERE {
?c  <http://xmlns.com/foaf/0.1/depiction> ?flag .
?c <http://dbpedia.org/property/commonName> ?country .
?c <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/class/yago/EuropeanCountries>
}
LIMIT 30
```

Q2 uses the Infobox dataset to identify the label of the different countries. However, the FOAF ontology also helps identifying the flag of the country and the YAGO (Yet Another Great Ontology) [17] ontology ensures that only European countries are selected. This excludes data which do not represent countries.

Nevertheless, it is more difficult to find flags for non European countries, while ensuring that only countries are selected. Indeed, in the YAGO ontology, *<http://dbpedia.org/class/yago/EuropeanCountries>* is a subclass of *<http://dbpedia.org/class/yago/Country108544813>*. But most European countries are not retrieved when querying the dataset with *<http://dbpedia.org/class/ yago/Country108544813>*. Indeed, the SPARQL endpoint does not provide access to inferred triples. It is necessary to perform a set of queries to retrieve relevant subclasses and use them for the generation of variables.

Out of 30 items including pictures of flags used as stimuli, 6 URIs did not resolve to a usable picture (HTTP 404 errors or encoding problem).

Q3 - Who succeeded to { Charles VII the Victorious } as ruler of France ?

```
SELECT DISTINCT ?kingHR ?successorHR
WHERE {
?x  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/class/yago/KingsOfFrance> .
?x <http://dbpedia.org/property/name> ?kingHR .
?x <http://dbpedia.org/ontology/successor> ?z .
?z <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/class/yago/KingsOfFrance> .
?z <http://dbpedia.org/property/name> ?successorHR
}
LIMIT 30
```

Q3 uses the YAGO ontology to ensure that the resource retrieved is indeed a king of France. Out of 30 results, one was incorrect (*The three Musketeers*). The query generated duplicates because of the multiple labels associated to each king. The same king was named for instance *Louis IX*, *Saint Louis*, *Saint Louis IX*. Whereas de-duplication is a straight forward process in this case, the risk of inconsistent naming patterns among options of the same item is more difficult to tackle. An item was indeed generated with the following 3 options: *Charles VII the Victorious*, *Charles 09 Of France*, *Louis VII*. They all use a different naming pattern, with or without the king's nickname and with a different numbering pattern.

Q4 - What is the capital of { Argentina }? With feedback

```
SELECT ?countryHR ?capitalHR ?pictureCollection
WHERE {
?country <http://dbpedia.org/property/commonName> ?countryHR .
?country <http://dbpedia.org/property/capital> ?capitalHR  .
?country                                   <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/class/yago/EuropeanCountries> .
?country <http://dbpedia.org/property/hasPhotoCollection> ?pictureCollection
}
LIMIT 30
```

The above question is a variation of Q1. It adds a picture collection from a distinct dataset in the response feedback. It uses the YAGO ontology to exclude countries outside Europe and resources which are not countries. A feedback section is added. When the candidate answers the item, he then receives a feedback if the platform allows it. In the feedback, additional information or formative resources can be suggested. Q4 uses the linkage of the DBpedia dataset with the Flickr wrapper dataset. However the Flickr wrapper data source was unavailable when we performed the experiment.

Q5 - Which category does { Asthma } belong to?

```
SELECT DISTINCT ?diseaseName ?category
WHERE {
?x <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ontology/Disease> .
?x <http://dbpedia.org/property/meshname> ?diseaseName .
?x <http://purl.org/dc/terms/subject> ?y .
?y <http://www.w3.org/2004/02/skos/core#prefLabel> ?category
}
LIMIT 30
```

Q5 aims to retrieve diseases and their categories. It uses SKOS and Dublin Core properties. The Infobox dataset is only used to find labels. Labels from the MESH vocabularies are even available. Nevertheless, the SKOS concepts are not related to a specific SKOS scheme. Categories retrieved range from *Skeletal disorders* to *childhood*. For instance, the correct answer to the question on *Obesity* is *childhood*.

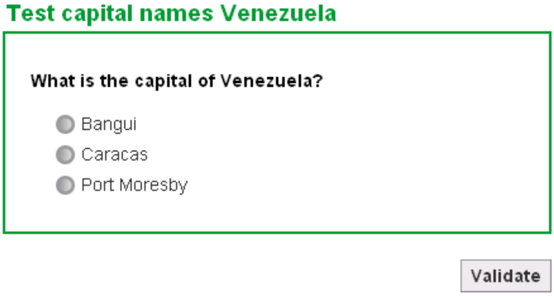
4.2 The Publication of Items on the TAO Platform

The TAO platform[3] is an open source semantic platform for the creation and delivery of assessment tests and items. It has been used in multiple assessment contexts,

[3] `http://www.tao.lu`

including large scale assessment in the PIAAC and PISA surveys of the OECD, diagnostic assessment and formative assessment.

We imported QTI items generated for the different item models in the platform, in order to validate the overall Linked Data based item creation streamline. Figure 4 presents an item generated from Q1 (Figure 3) imported in the TAO platform.



Test capital names Venezuela

What is the capital of Venezuela?

○ Bangui
○ Caracas
○ Port Moresby

Fig. 4. Item preview on the TAO platform

5 Data Analysis

The experimentation of the streamline was therefore tested with SPARQL queries which use various ontologies and which collect various types of variables. It raised two types of issues for which future work should find relevant solutions: the quality of the data and the relevance of particular statements for the creation of an assessment item.

5.1 Data Quality Challenges

In our experiment, the chance that an item will have a defective prompt or a defective correct answer is equal to the number of defective variables used for the item creation. Q1 uses the most challenging dataset in terms of data quality. 7 out of 30 questions had a defective prompt or a defective correct answer (23,33%).

The chance that an item will have defective distractors is represented by the following formula, where *D* is the total number of distractors, *d(V)* is the number of defective variables and *V* is the total number of variables:

$$D \times \frac{d(V)}{(V-1)}$$

We used 2 distractors. Among the items generated from Q1, 10 items had a defective distractor (33,33%). Overall, 16 out of 30 items had neither a defective prompt nor a defective correct answer nor a defective distractor (53,33%).

As a comparison, the items generated from unstructured content (text) that are deemed usable without edit were measured between 3,5% and 5% by Mitkov et al. [18] and between 12% and 21% by Karamanis et al. [7]. The difficulty of generating items from structured sources should be lower. Although a manual selection is necessary in any case, the mechanisms we have implemented can be improved.

The ontology

Q1 used properties from the Infobox dataset, which has no proper underlying ontology. Q1 can therefore be improved by using ontologies provided by DBpedia, as demonstrated by Q2 for which no distractor issue was identified. We present Q1 and Q2 to illustrate this improvement but it should be noted that there is not always a straight equivalent to the properties extracted from the Infobox dataset.

Q5 could be improved either if the dataset would be linked to a more structured knowledge organization system (KOS) or through an algorithm which would verify the nature of the literals provided as a result of the SPARQL query.

The labels

The choice of the label for each concept to be represented in an item is a challenge when concepts are represented by multiple labels (Q4). The selection of labels and their consistency can be ensured by defining representation patterns or by using datasets with consistent labeling practices.

Inaccurate statements

Most statements provided for the experiment are not inaccurate in their original context but they sometimes use properties which are not sufficiently precise for the usage envisioned (e.g., administrative capital). In other cases, the context of validity of the statement is missing (e.g., Leopoldville used to be the capital of a country called Congo). The choice of DBpedia as a starting point can increase this risk in comparison to domain specific data sources provided by scientific institutions for instance. Nevertheless, the Semantic Web raises similar quality challenges as the ones encountered in heterogeneous and distributed data sources [19]. Web 2.0 approaches, as well as the automatic reprocessing of data can help improve the usability of the Semantic Web statements. This requires setting up a traceability mechanism between the RDF paths used for the generation of items and the items generated.

Data linkage

Data linkage clearly raises an issue because of the reliability of the mechanism on different data sources. Q3 provided 6 problematic URIs out of 30 (i.e., 20%). Q4 generated items for which no URI from the linked data set was resolvable since the whole Flickr wrapper data source was unavailable. This clearly makes the generated items unusable. The creation of infrastructure components such as the SPARQL Endpoint status for CKAN[4] registered data sets[5] can help provide solutions to this quality issue over the longer run.

Missing inferences

Finally, the SPARQL endpoint does not provide access to inferred triples. Our streamline does not tackle transitive closures on the data consumer side (e.g., through repeated queries), as illustrated with Q3. Further consideration should be given to the provision of data including inferred statements. Alternatively, full datasets could be imported. Inferences could then be performed in order to support the item generation process.

[4] `http://www.ckan.net`

[5] `http://labs.mondeca.com/sparqlEndpointsStatus/index.html`

Different strategies can therefore be implemented to cope with data quality issues we encountered. Data publishers can improve the usability of the data, for instance with the implementation of an upper ontology in DBpedia. However, other data quality issues require data consumers to improve their data collection strategy, for instance to collect as much information as possible on the context of validity of the data, whenever it is available.

5.2 Data Selection

The experiment also showed that the Linked Data statements should be selected. The suitability of an assessment item for a test delivered to a candidate or a group of candidates is measured in particular through such information as the item difficulty. The difficulty can be assessed through a thorough calibration process in which the item is given to beta candidates for extracting psychometric indicators. In low stake assessment, however, the evaluation of the difficulty is often manual (candidate or teacher evaluation) or implicit (the performance of previous candidates who took the same item). In the item generation models we have used, each item has a different construct (i.e., it assesses a different knowledge). In this case, the psychometric variables are more difficult to predict [20]. A particular model is necessary to assess the difficulty of items generated from Semantic Web sources. For instance, it is likely that for a European audience, the capital of the Cook Islands will raise a higher rate of failure than the capital of Belgium. There is no information in the datasets, which can support the idea of a higher or lower difficulty. Moreover, the difficulty of the item also depends on the distractors, which in this experiment were generated on a random basis from a set of equivalent instances. As the generation of items from structured Web data sources will become more elaborated, it will therefore be necessary to design a model for predicting the difficulty of generated items.

6 Conclusion and Future Work

The present experimentation shows the process for generating assessment items and/or assessment variables from Linked Data. The performance of the system in comparison with other approaches shows its potential as a strategy for assessment item generation. It is expected that data linkage can provide relevant content for instance to propose formative resources to candidates who failed an item or to illustrate a concept with a picture published as part of a distinct dataset.

The experimentation shows the quality issues related to the generation of items based on DBpedia. It should be noted that the measurements were made with a question which raises particular quality issues. It can be easily improved as shown with other questions. Nevertheless the Linked Data Cloud also contains datasets published by scientific institutions, which may therefore raise less data accuracy concerns. In addition, the usage model we are proposing is centered on low stake assessment, for which we believe that the time saved makes it worthwhile having to clean some of the data, while the overall process remains valuable.

Nevertheless, additional work is necessary both on the data and on the assessment items. The items created demonstrate the complexity of generating item variables for simple assessment items. We aim to investigate the creation of more complex items and the relevance of formative resources which can be included in the item as feedback. Moreover, the Semantic Web can provide knowledge models from which items could be generated. Our work is focused on semi-automatic item generation, where users create item models, while the system aims to generate the variables. Nevertheless, the generation of the items from a knowledge model as in the work carried out in the scope of the DynaLearn project [11] requires that more complex knowledge is encoded in the data (e.g., what happens to water when the temperature decreases). The type and nature of data published as Linked Data need therefore to be further analyzed in order to support the development of such models for the fully automated generation of items based on knowledge models.

The case of inaccurate statements also illustrates the lack of current tools to provide a feedback on the original knowledge model. Considering the *Three Musketeers* erroneously presented as kings of France, a manual verification showed that it was a property of <*http://dbpedia.org/resource/Louis_XIII_of_France*> that however did not appear in the current version of the corresponding Wikipedia page. It is necessary to allow users to complement or contradict statements from the Semantic Web without having to maintain a dataset, i.e., include Web 2.0 mechanisms on the LoD cloud construction. This requires the development of tools and common practices for the validation of data or the representation of multiple perspectives on the Semantic Web.

We will focus our future work on the creation of an authoring interface for item models with the use of data sources from the Semantic Web, on the assessment of item quality, on the creation of different types of assessment items from Linked Data sources, on the traceability of items created, including the path on the Semantic Web datasets which were used to generate the item, and on the improvement of data selection from semantic datasets.

Acknowledgments. This work was carried out in the scope of the iCase project on computer-based assessment. It has benefited from the TAO semantic platform for e-assessment (https://www.tao.lu/) which is jointly developed by the Tudor Research Centre and the University of Luxembourg, with the support of the Fonds National de la Recherche in Luxembourg, the DIPF (Bildungsforschung und Bildungsinformation), the Bundesministerium für Bildung und Forschung, the Luxemburgish ministry of higher education and research, as well as OECD.

References

1. Belluck, P.: To Really Learn, Quit Studying and Take a Test. New York Times (January 20, 2011)
2. Karpicke, J.D., Blunt, J.R.: Retrieval Practice Produces More Learning than Elaborative Studying with Concept Mapping. Science (2011)
3. Gilbert, L., Gale, V., Warburton, B., Wills, G.: Report on Summative E-Assessment Quality (REAQ). In: Joint Information Systems Committee, Southampton (2008)
4. Aldabe, I., Lopez de Lacalle, M., Maritxalar, M., Martinez, E., Uria, L.: ArikIturri: An Automatic Question Generator Based on Corpora and NLP Techniques. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 584–594. Springer, Heidelberg (2006)

5. Lee, J.S.Y.: Automatic correction of grammatical errors in non-native English text. PhD dissertation at The Massachussets Institute of Technology (2009)
6. Goto, T., Kojiri, T., Watanabe, T., Iwata, T., Yamada, T.: Automatic Generation System of Multiple-Choice Cloze Questions and its Evaluation. Knowledge Management & E-Learning: An International Journal (KM&EL) 2(3), 210 (2010)
7. Karamanis, N., Ha, L.A., Mitkov, R.: Generating multiple-choice test items from medical text: a pilot study. In: Proceedings of the Fourth International Natural Language Generation Conference, pp. 111–113 (2006)
8. Lin, Y.C., Sung, L.C., Chen, M.C.: An Automatic Multiple-Choice Question Generation Scheme for English Adjective Understanding. In: Workshop on Modeling, Management and Generation of Problems/Questions in eLearning, the 15th International Conference on Computers in Education (ICCE 2007), pp. 137–142 (2007)
9. Brown, J.C., Frishkoff, G.A., Eskenazi, M.: Automatic question generation for vocabulary assessment. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 819–826 (2005)
10. Sung, L.-C., Lin, Y.-C., Chen, M.C.: The Design of Automatic Quiz Generation for Ubiquitous English E-Learning System. In: Technology Enhanced Learning Conference (TELearn 2007), Jhongli, Taiwan, pp. 161–168 (2007)
11. Linnebank, F., Liem, J., Bredeweg, B.: Question generation and answering. DynaLearn, EC FP7 STREP project 231526, Deliverable D3.3 (2010)
12. Liu, B.: SARAC: A Framework for Automatic Item Generation. Presented at the 2009 Ninth IEEE International Conference on Advanced Learning Technologies (ICALT), Riga, Latvia, pp. 556–558 (2009)
13. Xu, Y., Seneff, S.: Speech-Based Interactive Games for Language Learning: Reading, Translation, and Question-Answering. Computational Linguistics and Chinese Language Processing 14(2), 133–160 (2009)
14. Lai, H., Alves, C., Gierl, M.J.: Using automatic item generation to address item demands for CAT. In: Proceedings of the 2009 GMAC Conference on Computerized Adaptive Testing (2009)
15. Gierl, M.J., Zhou, J., Alves, C.: Developing a Taxonomy of Item Model Types to Promote Assessment Engineering. Journal of Technology, Learning, and Assessment 7(2) (2008)
16. Sarre, S., Foulonneau, M.: Reusability in e-assessment: Towards a multifaceted approach for managing metadata of e-assessment resources. In: Fifth International Conference on Internet and Web Applications and Services (2010)
17. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, pp. 697–706 (2007)
18. Mitkov, R., An Ha, L., Karamanis, N.: A computer-aided environment for generating multiple-choice test items. Natural Language Engineering 12(02), 177–194 (2006)
19. Foulonneau, M., Cole, T.W.: Strategies for Reprocessing Aggregated Metadata. In: Rauber, A., Christodoulakis, S., Tjoa, A.M. (eds.) ECDL 2005. LNCS, vol. 3652, pp. 290–301. Springer, Heidelberg (2005)
20. Bejar, I.I., Lawless, R.R., Morley, M.E., Wagner, M.E., Bennett, R.E., Revuelta, J.: A feasibility study of on-the-fly item generation in adaptive testing. Educational Testing Service (2002)

Using Linked Data to Reduce Learning Latency for e-Book Readers

Julien Robinson, Johann Stan, and Myriam Ribière

Alcatel-Lucent Bell Labs France, 91620 Nozay, France
Julien.Robinson@alcatel-lucent.com

Abstract. Web 2.0 technologies provide an opportunity to transform learning into a social experience: social learning can directly benefit from user-generated content analysis. The e-book device is an object central to the learning process; it becomes an ideal medium to promote innovative learning tools. In this paper, we describe how we analyse user annotations in e-books using Linked Data to reduce the latency between professor knowledge, book content and student assimilation.

Keywords: Social Learning, Semantic Web, Linked Data, e-Book.

1 Introduction

People learning with a given e-book form a de-facto community, which is even more relevant if they are reading at the same time, with the same purpose or during the same activity, e.g. a class. Currently, very few platforms allow social interactions between such people that are dedicated to their reading activity; readers can connect but at a very high level and generally outside the platform (e.g. forums in LibraryThing[1], chatting in Bookglutton[2] or more general social cataloging features GoodReads[3] and Shelfari[4]).

The main problem with these platforms is that interactions cannot be directly linked to the context of reading and thus do not allow fine-grained interactions about a specific part of the book. Bookglutton goes a step further by letting users chat while reading, but does not exploit the contents and context of annotations in the e-book. This issue was already explored [1] [2], but no solution was proposed based on the semantic analysis of the annotations.

An important observation for our work is the fact that in most cases, annotations in e-books are in the form of short, unstructured textual messages, associated to images, paragraphs or chapters. These can include transcriptions, reformulations, additional comments, questions or explanations (such as student annotations in BookGlutton[5]). Clearly, services that leverage these annotations

[1] http://www.librarything.com/ - visited September 2010

[2] http://www.bookglutton.com/ - visited January 2011

[3] http://www.goodreads.com/ - visited October 2010

[4] http://www.shelfari.com/ - visited March 2011

[5] e.g. http://www.bookglutton.com/glutton/melissthangx0x0/14459

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 28–34, 2012.

can increase the social dimension of reading. They can help readers determine a common environment to socially learn by seeing others' annotations or to generate interactions between students during a same activity.

In this paper we present our ongoing work on services that leverage semantic technologies and especially Linked Data for providing a social dimension for reading and implicitly, learning. First we summarize how we associate annotations, in natural language, to Linked Data concepts; then we describe the heatmap, the social navigation feature that leverages this work for efficient reading; finally we present the Tagging Beak, a research prototype that should pave the way for the next version of the heatmap.

2 Management of Shared Annotations Using Linked Data

Linked Data is an evolving component of the Semantic Web. It can be defined as a community effort to extract or publish data from various sources, interlink it and store it in a semantic web format, such as RDF. The content of these online linked datasets is diverse. It consists mostly of data about people, companies, books, geographic locations, films, music, television and radio programs, information about medicine (genes, proteins, clinical trials) and online communities (statistical data). The most used online dataset is certainly DBPedia[3].

An important source of social knowledge is the set of annotations produced by the user. In our framework, our primary hypothesis is to consider these annotations on a conceptual level. More concretely, this means that we associate each annotation to one or several concepts in a Linked Data source (e.g. DBPedia). There are several reasons to make this association: (i) provide a rich approximation of their meaning, (ii) the capability to retrieve related concepts and category terms, operation that we call semantic expansion[6] and (iii) the fact that annotations are generally short and that contextual cues are not always available. Therefore, Linked Data sources can enrich the description of such annotations with additional metadata and related concepts.

The association of annotations to concepts in Linked Data is performed by a disambiguation process that leverages the context of the annotation to match it with the most relevant concept in the knowledge base. Each annotation provided by the user is associated to a context, which is constructed from different sources, such as previous annotations shared by the user in the e-book and the set of annotations shared by others that are related to the given paragraph. Parallel to this, a set of candidate concepts are retrieved from DBPedia, that contain the given annotation in their label. The candidate concepts are ranked using cosine similarity between the vector constructed from their abstracts and the contextual cue vector. Once this disambiguation has been performed, the second use of Linked Data is the expansion of the concept by exploring both its semantic neighborhood and hierarchical links.

[6] As an example, here is the semantic expansion of the annotation "Facebook": [Global Internet Community, Student Culture, Social Information Processing, Social Networking Service, Social Media, Web 2.0, Internet Technologies].

3 Easy Social Navigation with Heatmaps

Key to the exploitation of Linked Data results in social learning is the way the user will perceive the analysis results and whether it will be useful or not. In [1], Jae K. Kim et al. describe how they let students place symbolic evaluations as annotations, and then graphically represent the synthesis of these annotations as colors in the e-book, thus enabling what is called **Social Navigation**. We follow the same investigation track, but instead of focusing on design choices and symbolic evaluations, we analyze free text annotations using Semantic Web techniques.

The user, a learning reader (our target is post-graduate students), opens an e-book and states their interest using freely chosen keywords; the keywords express their goal for the reading session. The keywords are semantically analyzed and a global similarity computation is run on the annotations of the book. Each annotation is attributed a heat value based on its relevance to the user's goal in reading the book. The appearance of the book is modified to display in a colored way the zones where the annotations are the most heated.

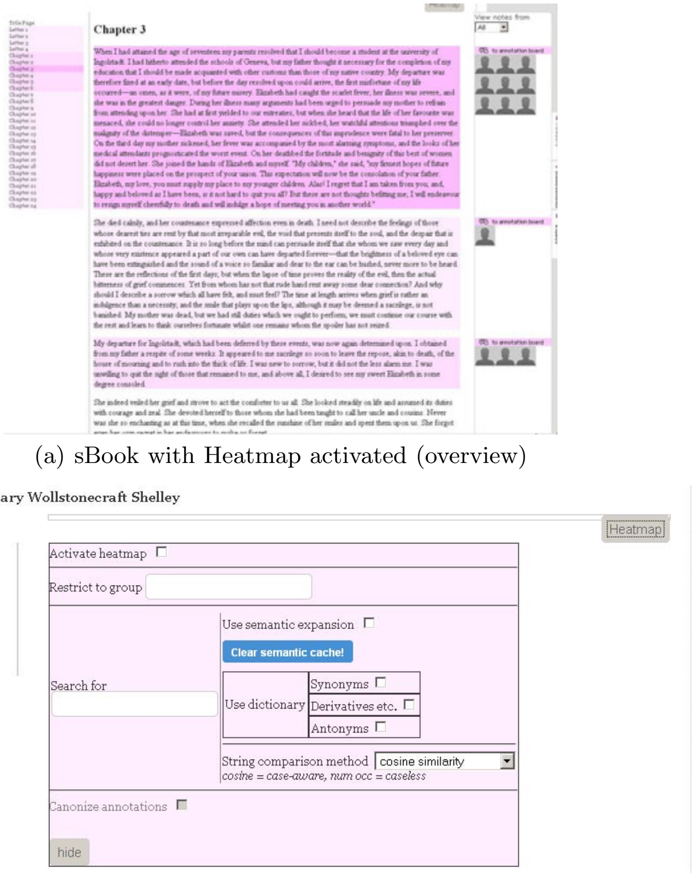
Figure 1(a) presents an example of the heatmap feature. This was produced by our prototype of the sBook [4] using data gathered from annotations from BookGlutton (229 annotations by 39 different users on the e-book "Frankenstein“ by Mary Shelley).

The choice was made to use the paragraph as the finest precision of annotation heat. The heat of a paragraph is the total sum of the heat of each annotation on the paragraph. Note that the heatmap is a navigation indication for the user, it is not meant to be the final step for a learning reader; it is completed by the display of authors of annotations (on the right-hand panel) and the use of what we call annotation boards: these are separate windows that browse through all the annotations of a paragraph and the comments inspired by these annotations. The annotation board is where the user may find valuable information and participate in discussions.

The choice of the algorithm for heating an annotation is the focus of our current research. Figure 1(b) presents the administration interface that enables to choose different implementations for the user to experiment with. Annotations are heated based on their proximity to the resulting vector of words. We currently implement three expansion methods:

- no expansion, we simply search for the keyword
- a dictionary-based heatmap, that may take into account synonyms, antonyms and derived words (e.g. adjectives for a noun)
- a Linked-Data-based heatmap, that takes into account the expansion of the search keyword as described in the present paper.

Our first experiments show that different expansion techniques may be relevant for different types of words. For instance, the word "Paris" is expected to be linked to "France", which is the case using DBpedia but not using a dictionary; this hints that Linked Data would be appropriate for named entities. On the other hand, the word "character" expands to "personage" and "reputation" using



(a) sBook with Heatmap activated (overview)

(b) Administration Heatmap Configuration

Fig. 1. Heatmap screenshots

a dictionary, while no such expansion is provided by DBpedia; this would indicate that dictionary expansion is more appropriate for literary theory terms. This is why we keep both options open and also separate. We intend to validate the utility of each expansion based on user experimentation by the end of 2011.

In a first step the annotations are processed as text; currently being implemented, we first process each annotation by associating it to a cluster of tags, obtained by semantic evaluation techniques. The similarity measure will be computed between the cluster of tags of the annotation and the vector of words expanded from the user keyword.

It may be noted that, while we explore keyword-based heatmaps, we also intend to explore other options: time-based heatmap, where the heat of an annotation is related to its creation date; location-based heatmap, taking into account the location of the annotation's author; or more generally context-based and activity-based heatmaps (which present intersections). These will be used for further user experimentations, first to compare their effectiveness with keyword-based heatmaps, second because they are relevant to our general research on the link between the digital and physical worlds.

4 User Interaction Profiles

While the heatmap is a social navigation tool based on a request by the current reader, we intend to go further in our recommendation strategy. We aim to recommend not only annotations or parts of a book, but also other members of the reading community. This social recommendation process requires specific user profiles.

In this section, we focus on our global approach to building these user profiles based on interactions such as shared annotations or messages. Such a profile is called *"user interaction profile"*, it is based on individual content sharing activity or interactions with peers. Its aim is to identify concepts that either represent an expertise field of the user or a topic that motivates them to interact. A more detailed definition of interaction profiles and the corresponding algorithms can be found in [5].

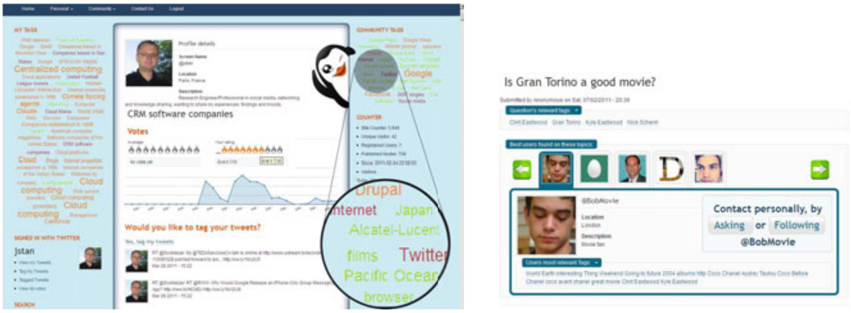
In a first step, the algorithms to build these interactions profiles have been developed in the scope of a Twitter analysis tool. Twitter proposes a wealth of public social interactions, in a format close to annotations: short but meaningful messages, sometimes general comments and sometimes addressed to specific contacts. We use this as a means to tune and validate the algorithm before integrating it in the social book. We propose an online interactive tool, the Tagging Beak[7], to display these profiles for Twitter users (Figure 2(a)) and also to recommend people relevant to an information need expressed as a question in natural language (Figure 2(b)).

Users connect with their Twitter account; the system reads their tweets and extracts keywords and named entities. Profiles are built by contextualized semantic matching (*disambiguation*) and expansion of such items: extracted keywords are used to find concepts in DBpedia; in order to disambiguate homonyms, the concepts are then evaluated for their proximity to the previous tweets of the user and of their community[8] using semantic techniques. The most relevant concepts are then expanded along chosen dimensions in DBpedia, which yields neighbouring concepts, who are in turn filtered based on their proximity to the user profile. The resulting set of ponderated concepts is the user's interaction profile.

Sentiment analysis is also used, as well as declarative feedback from users. This helps determine not only the expertise of a user on a subject but also their willingness to share. They can for example say that they have high expertise on a subject but that they wish to share this expertise only with classmates and teachers and that they are ready to interact on this topic only when they are at the library or at school. With the help of DBPedia, we also retrieve the

[7] The Tagging Beak Collaborative Social Search System - http://tbeak.com

[8] In Twitter, communities are built upon the following / follower link between users.



(a) Interaction Profile of a user in the Tagging Beak (overview)

(b) People Recommendation Interface in the Tagging Beak (overview)

Fig. 2. Tagging Beak screenshots

hierarchical tree associated to the concept, which allows the user to share the same concept with other social spheres, but with less granularity. In this way, the user can share different levels of granularity about the same knowledge with multiple social spheres (e.g. detailed knowledge for the class, less detailed for friends).

Based on expertise and willingness to share, a user may be recommended to another user for answering a question. This recommendation strategy is key to social learning. It ranks people in the close and more distant (e.g. friends of friends) community of the user according to the semantic similarity between the topics of the question and their interaction profiles. The Tagging Beak then enables either to put the question to the recommended contact or to start following them in Twitter (if this is not already the case).

Linked Data allows to retrieve the hierarchical tree and semantic neighborhood associated to a given concept. In the case of collaborative learning, we leverage this feature by offering users a more granular way of sharing their interaction profile, constructed from the shared annotations. Such interaction profiles make it possible for the system to select the right person to ask a question on a specific topic. In order to overcome the problem of cold start in case of new members in such learning communities, we are currently investigating the possibility to connect this service to other Social Platforms and to build a user interaction profile from annotations shared in them. In this way, the user will be recommended people to interact with even if they did not share sufficient annotations. More specifically, each time the user opens the e-book, they will be presented with the most relevant readers to interact with, based on complementary expertise in specific topics of interest and declared sharing preferences related to that topic. This provides students with a seamless way of linking to co-learners and thus form spontaneous learning communities on a specific part of a lecture.

5 Conclusion

In this paper we have presented our research, which focuses on integrating Linked Data results into e-book navigation tools for students in order to enhance social learning. We base our work on user annotations, we process these annotations with Linked Data technologies, we use the produced synthetized results to modify our e-book display; we hope to maximize reading and learning efficiency, to create common environments and implicitly, communication opportunities. At the basis of our innovation is the hypothesis that the huge amount of shared content in different Social Platforms offers exciting opportunities to enrich the learning experience and reduce the knowledge latency in the class, which we define as the delay between the encoding of the knowledge of a professor on a specific subject into information (e.g. a lecture or slides) and the assimilation of this information as knowledge by students.

Acknowledgment. We would like to thank BookGlutton for providing open data that we were able to exploit, and in particular Aaron Miller (CTO). The set of ideas about the future of e-books we have presented in this paper are currently being addressed in the scope of a collaborative project between Alcatel-Lucent, Abilene Christian University and Cambridge University Press[9].

References

1. Kim, J.K., Farzan, R., Brusilovsky, P.: Social navigation and annotation for electronic books. In: BooksOnline 2008: Proceeding of the 2008 ACM Workshop on Research Advances in Large Digital Book Repositories, pp. 25–28. ACM, New York (2008)
2. Freyne, J., Farzan, R., Brusilovsky, P., Smyth, B., Coyle, M.: Collecting community wisdom: integrating social search & social navigation. In: IUI 2007: Proceedings of the 12th International Conference on Intelligent User Interfaces, pp. 52–61. ACM Press, New York (2007)
3. Lehmann, J., Bizer, C., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - a crystallization point for the web of data. Journal of Web Semantics 7(3), 154–165 (2009)
4. Ribière, M., Picault, J., Squedin, S.: The sbook: towards social and personalized learning experiences. In: Proceedings of the Third Workshop on Research Advances in Large Digital Book Repositories and Complementary Media, BooksOnline 2010, pp. 3–8. ACM, New York (2010)
5. Johann, S., Viet-Hung, D., Pierre, M.: Semantic user interaction profiles for better people recommendation. In: Proceedings of ASONAM 2011, Kaohsiung, Taiwan (to appear, 2011)

[9] Cambridge University - http://www.cambridge.org/press/article.asp?artid=154961

The OU Linked Open Data: Production and Consumption

Fouad Zablith, Miriam Fernandez, and Matthew Rowe

Knowledge Media Institute (KMi), The Open University
Walton Hall, Milton Keynes, MK7 6AA, United Kingdom
{f.zablith,m.fernandez,m.c.rowe}@open.ac.uk

Abstract. The aim of this paper is to introduce the current efforts toward the release and exploitation of The Open University's (OU) Linked Open Data (LOD). We introduce the work that has been done within the LUCERO project in order to select, extract and structure subsets of information contained within the OU data sources and migrate and expose this information as part of the LOD cloud. To show the potential of such exposure we also introduce three different prototypes that exploit this new educational resource: (1) the OU expert search system, a tool focused on finding the best experts for a certain topic within the OU staff; (2) the Social Study system, a tool that relies on Facebook information to identify common interest between a user's profile and recommend potential courses within the OU; and (3) Linked OpenLearn, an application that enables exploring linked courses, Podcasts and tags to OpenLearn units. Its aim is to enhance the browsing experience for students, by detecting relevant educational resources on the fly while studying an OpenLearn unit.

Keywords: Linked Data, education, expert search, social networks.

1 Introduction

The explosion of the Linked Open Data (LOD) movement in the last few years has produced a large number of interconnected datasets containing information about a large variety of topics, including geography, music and research publications among others. [2]

The movement is receiving worldwide support from public and private sectors like the UK[1] and US[2] governments, international media outlets, such as the BBC [5] or the New York Times [1], and companies with a social base like Facebook.[3] Such organisations are supporting the movement either by releasing large datasets of information or by generating applications that exploit it to connect data across different locations.

[1] http://data.gov.uk

[2] http://www.data.gov/semantic/index

[3] http://developers.facebook.com/docs/opengraph

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 35–49, 2012.

Despite its relevance and the support received in the last few years, very few pieces of work have either released or exploited LOD in the context of education. One of these few examples is the DBLP Bibliography Server Berlin,[4] which provides bibliographic information about scientific papers. However, education is principally one of the main sectors where the application of the LOD technologies can provoke a higher impact.

When performing learning and investigation tasks, students and academics have to go through the tedious and laborious task of browsing different information resources, analysing them, extracting their key concepts and mentally linking data across resources to generate their own conceptual schema about the topic. Educational resources are generally duplicated and dispersed among different systems and databases, and the key concepts within these resources as well as their inter and intra connections are not explicitly shown to users. We believe that the application of LOD technologies within and across educational institutions can explicitly generate the necessary structure and connections among educational resources, providing better support to users in their learning and investigation tasks.

In this context, the paper presents the work that has been done within The Open University (OU) towards the release and exploitation of several educational and institutional resources as part of the LOD cloud. First, we introduce the work that has been done within the LUCERO project to select, extract and structure subsets of OU information as LOD. Second, we present the potential of this data exposure and interlinking by presenting three different prototypes: (1) the OU expert search system, a tool focused on finding the best experts for a certain topic within the OU staff; (2) the Social Study system, a tool focused on exploiting Facebook information to identify common interests between a user and recommend potential courses within the OU, and; (3) Linked Open Learn, an application that enables exploring linked courses, Podcasts and tags to OpenLearn units.

The rest of the paper is organised as follows: Section 2 presents the state of the art in the areas of LOD within the education context. Section 3 presents the work that has been done within the LUCERO project to expose OU data as part of the LOD cloud. Sections 4, 5 and 6 present example prototype applications that consume the OU's LOD for Expert Search, Social Study and Linked OpenLearn respectively. Section 7 describes the conclusions that we have drawn from this work, and section 8 presents our plans for future work.

2 Related Work

While LOD is being embraced in various sectors as mentioned in the previous section, we are currently witnessing a substantial increase in universities adopting the Linked Data initiative. For example, the University of Sheffield's Department of Computer Science[5] provides a Linked Data service describing research

[4] http://www4.wiwiss.fu-berlin.de/dblp/

[5] http://data.dcs.shef.ac.uk

groups, staff and publications, all semantically linked together [6]. Similarly the University of Southampton has recently announced the release of their LOD portal (http://data.southampton.ac.uk), where more data will become available in the near future. Furthermore, the University of Manchester's library catalogue records can now be accessed in RDF format[6]. In addition, other universities are currently working on transforming and linking their data: University of Bristol,[7] Edinburgh (e.g., the university's buildings information is now generated in LOD[8], and Oxford[9]. Furthermore the University of Muenster announced a funded project, LODUM, the aim of which is to release the university's research information as Linked Data. This includes information related to people, projects, publications, prizes and patents.[10]

With the increase of the adoption of LOD publishing standards, the exchange of data will be much easier, not only within one university, but also across the LOD ready ones. This enables, for example, the comparison of specific qualifications offered by different universities in terms of courses required, pricing and availability.

3 The Open University Linked Open Data

The Open University is the first UK University to expose and publish its organisational information in LOD.[11] This is accomplished as part of the LUCERO project (Linking University Content for Education and Research Online)[12], where the data extraction, transformation and maintenance are performed. This enables having multiple hybrid datasets accessible in an open way through the online access point: http://data.open.ac.uk.

The main purpose of releasing all this data as part of the LOD cloud is that members of the public, students, researchers and organisations will be able to easily search, extract and, more importantly, reuse the OU's information and data.

3.1 Creating the OU LOD

Detailed information about the process of LOD generation within the OU is available at the LUCERO project website.[12] We briefly discuss in this section the steps involved in the creation of Linked Data. To achieve that, the main requirement is to have a set of tools that generate RDF data from existing data sources, load such RDF into a triple store, and make it accessible through a web access point.

[6] http://prism.talis.com/manchester-ac

[7] https://mmb.ilrt.bris.ac.uk/display/ldw2011/University+of+Bristol+data

[8] http://ldfocus.blogs.edina.ac.uk/2011/03/03/university-buildings-as-linked-data-with-scraperwiki

[9] http://data.ox.ac.uk

[10] http://www.lodum.de

[11] http://www3.open.ac.uk/media/fullstory.aspx?id=20073

[12] http://lucero-project.info

Given the fact that the OU's data repositories are scattered across many departments, using different platforms, and subject to constant update, a well-defined overflow needs to be put in place. The initial workflow is depicted in Figure 1, and is designed to be efficient in terms of time, flexibility and reusability. The workflow is component based, and the datasets characteristics played a major role in the implementation and setup of the components. For example, when the data sources are available in XML format, the XML updater will handle the process of identifying new XML entities and pass them to the RDF extractor, where the RDF data is generated, and ready to be added to (or removed from) the triple store. Finally the data is exposed to the web, and can be queried through a SPARQL endpoint.[13]

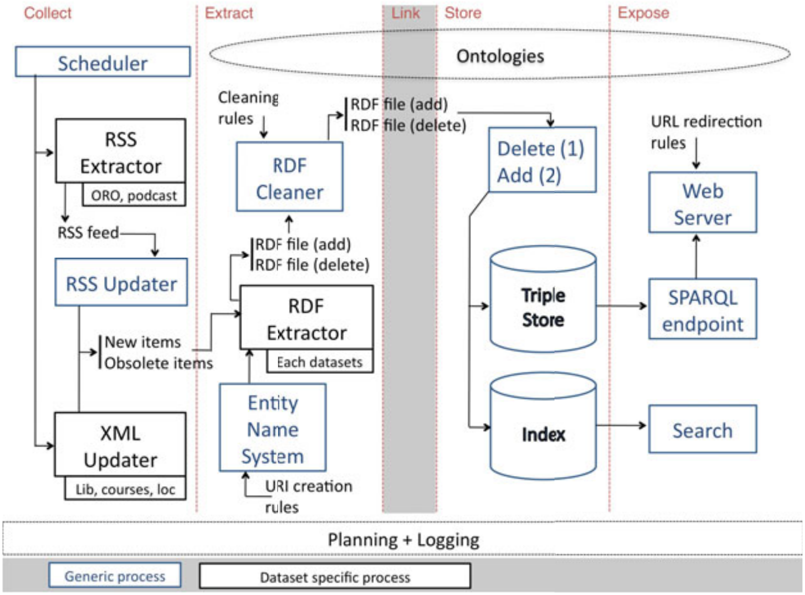


Fig. 1. The LUCERO workflow

The scheduler component takes care of initiating the extraction/update process at specific time intervals. This update process is responsible for checking what was added, modified, or removed from the dataset, and accordingly applies to the triple store the appropriate action. Having such a process in place is important in the OU scenario where the data sources are continuously changing. Another point worth mentioning is the linking process that links entities coming from different OU datasets (e.g., courses mentioned in Podcast data and library records), in addition to linking external entities (e.g., course offerings in

[13] http://data.open.ac.uk/query

a GeoNames defined location[14]). To achieve interlinking OU entities, independently from which dataset the extraction is done, we rely on an Entity Named System, which generates a unique URI (e.g., based on a course code) depending on the specified entity (this idea was inspired from the Okkam project[15]) . Such unique URIs enable a seamless integration and extraction of linked entities within common objects that exist in the triple store and beyond, one of the core Linked Data requirements [3].

3.2 The Data

Data about the OU courses, Podcasts and academic publications are already available to be queried and explored, and the team is now working to bring together educational and research content from the university's campus information, OpenLearn (already available for testing purposes) and library material. More concretely, data.open.ac.uk offers a simple browsing mechanism, and a SPARQL endpoint to access the following data:

- The Open Research Online (ORO) system[16], which contains information about academic publications of OU research. For that, the Bibliographic Ontology (bibo)[17] is mainly used to model the data.
- The OU Podcasts,[18] which contain Podcast material related to courses and research interests. A variety of ontologies are used to model this data, including the W3C Media Ontology,[19] in addition to a specialised SKOS[20] representation of the iTunesU topic categories.
- A subset of the courses from the Study at the OU website,[21] which provides courses information and registration details for students. We model this data by relying on the Courseware,[22] AIISO,[23] XCRI,[24] MLO[25] and GoodRelations ontologies [4], in addition to extensions that reflect OU specific information (e.g., course assessment types).

  Furthermore, there are other sources of data that are currently being processed. This includes for example the OU YouTube channel,[26] the library catalogue, and public information about locations on the OU campus (e.g., buildings) and university staff.

[14] http://www.geonames.org
[15] http://www.okkam.org
[16] http://oro.open.ac.uk
[17] http://bibliontology.com/specification
[18] http://podcast.open.ac.uk
[19] http://www.w3.org/TR/mediaont-10
[20] http://www.w3.org/2004/02/skos
[21] http://www3.open.ac.uk/study
[22] http://courseware.rkbexplorer.com/ontologies/courseware
[23] http://vocab.org/aiiso/schema
[24] http://svn.cetis.ac.uk/xcri/trunk/bindings/rdf/xcri_rdfs.xml
[25] http://svn.cetis.ac.uk/xcri/trunk/bindings/rdf/mlo_rdfs.xml
[26] http://www.youtube.com/user/TheOpenUniversity

4 The OU Expert Search

Expert search can be defined as the task of identifying people who have relevant expertise in a topic of interest. This task is key for every enterprise, but especially for universities, where interdisciplinary collaborations among research areas is considered a high success factor. Typical user scenarios in which expert search is needed within the university context include: a) finding colleagues from whom to learn, or with whom to discuss ideas about a particular subject; b) assembling a consortium with the necessary range of skills for a project proposal, and; c) finding the most adequate reviewers to establish a program committee.

As discussed by Yimam-Seid and Kobsa [7], developing and manually updating an expert system database is time consuming and hard to maintain. However, valuable information can be identified from documents generated within an organisation [8]. Automating expert finding from such documents provides an efficient and sustainable approach to expertise discovery.

OU researchers, students and lecturers constantly produce a plethora of documents, including for example conference articles, journal papers, thesis, books, reports and project proposals. As part of the LUCERO project, these documents have been pre-processed and made accessible as LOD. The purpose of this application is therefore to exploit such information so that OU students and researchers can find the most appropriate experts starting from a topic of interest.[27]

4.1 Consumed Data

This application is based on two main sources of information: (a) LOD from the Open Research Online system, and (b) additional information extracted from the OU staff directory. The first information source is exploited in order to extract the most suitable experts about a certain topic. The second information source complements the previous recommended set of experts by providing their corresponding contact information within the OU. Note that sometimes, ex-OU members and external collaborators or OU researchers may appear in the ranking of recommended experts. However, for those individuals, no contact information is provided, indicating that those experts are not part of the OU staff.

As previously mentioned, the information provided by Open Research Online contains data that describe publications originating from OU researchers. In particular, among the properties provided for each publication, this system exploits the following ones: a) the title, b) the abstract, c) the date, d) the authors and, e) the type of publication, i.e., conference paper, book, thesis, journal paper, etc.

To exploit this information the system performs two main steps. Firstly when the system receives the user's query, i.e., the area of expertise where a set of experts need to be found (e.g., “semantic search”), the system uses the title and

[27] The OU Expert Search is accessible to OU staff at:
`http://kmi-web15.open.ac.uk:8080/ExpertSearchClient`

abstract of the publications to find the top-n documents related to that area of expertise. At the moment n has been empirically set to 10.

Secondly, once the top-n documents have been selected, the authors of these documents are extracted and ranked according to five different criteria: (a) original score of their publications, (b) number of publications, (c) type of publications, (d) date of the publications and, (e) other authors of the publication.

The initial score of the publications is obtained by matching the user's keyword query against the title and the abstract of the OU publications. Publications that provide a better match within their title and abstract against the keywords of the query are ranked higher. This matching is performed and computed using the Lucene[28] text search engine. Regarding the number of publications, authors with a higher number of publications (among the top-n previously retrieved) are ranked higher. Regarding the type of publication, theses are ranked first, then books, then journal papers, and finally conference articles. The rationality behind this is that an author writing a thesis or a book holds a higher level of expertise than an author who has only written conference papers. Regarding the date of the publication, we consider the 'freshness' of the publications and continuity of an author's publications within the same area. More recent publications are ranked higher than older ones, and authors publishing in consecutive years about a certain topic are also ranked higher than authors that have sporadic publications about the topic. Regarding other authors, experts sharing a publication with fewer colleagues are ranked higher. The rationality behind this is that the total knowledge of a publication should be divided among the expertise brought into it, i.e., the number of authors. Additionally we also consider the order of authors in the publication. Main authors are considered to have a higher level of expertise and are therefore ranked higher.

To perform the first step (i.e., retrieving the top-n documents related to the user's query) we could have used the SPARQL endpoint and, at run-time, searched for those keywords within the title and abstract properties of the publications. However, to speed the search process up, and to enhance the query-document matching process, we have decided to pre-process and index the title and abstract information of the publications using the popular Lucene search engine. In this way, the fuzzy and spelling check query processing and ranking capabilities of the Lucene search engine are exploited to optimise the initial document search process.

To perform the second step, once the top-n documents have been selected, the rest of the properties of the document (authors, type, and date) are obtained at run-time using the SPARQL endpoint.

Finally, once the set of authors have been ranked, we look for them in the OU staff directory (using the information about their first name and last name). If the author is included in the directory, the system provides related information about the job title, department within the OU, e-mail address and phone number. By exploiting the OU staff directory we are able to identify which experts are

[28] `http://lucene.apache.org/java/docs/index.html`

members of the OU and which of them are external collaborators, or old members not further working for the institution.

Without the structure and conceptual information provided by the OU LOD, the implementation of the previously described ranking criteria, as well as the interlinking of data with the OU staff directory, would have required a huge data pre-processing effort. The OU LOD provides the information with a fine-grained structure that facilitates the design of ranking criteria based on multiple concepts, as well as the interlinking of information with other repositories.

4.2 System Implementation

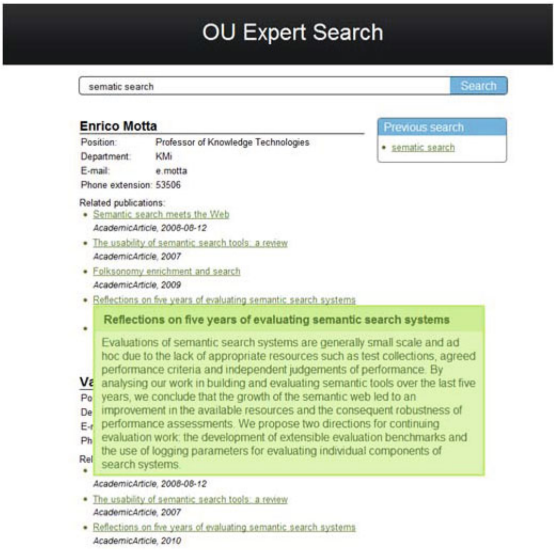
The system is based on lightweight client server architecture. The back end (or server side) is implemented as a Java Servlet, and accesses the OU LOD information by means of HTTP requests to the SPARQL endpoint. Some of the properties provided by the LOD information (more particularity the title and the abstract of the publications) are periodically indexed using Lucene to speed-up and enhance the search process by means of the exploitation of its fuzzy and spell checker query processing, and ranking capabilities. The rest of the properties (authors, date, and type of publications) are accessed at run time, once the top-n publications have been selected.

The front end is a thin client implemented as a web application using HTML, CSS and Javascript (jQuery).[29] The client doesn't handle any processing of the data, it only takes care of the visualisation of the search results and the search input. It communicates with the back-end by means of an HTTP request that passes as a parameter the user's query and retrieves the ranking of authors and their corresponding associated information by means of a JSON object.

4.3 Example and Screenshot

In this section, we provide an example of how to use the OU expert search system. As shown in Figure 2, the system receives as a keyword query input "*semantic search*", with the topic for which the user aims to find an expert. As a result, the system provides a list of authors ("*Enrico Motta*", "*Vanessa Lopez*", etc), who are considered to be the top OU experts in the topic. For each expert, if available, the system provides the contact details (department, e-mail, phone extension) and the top publications about the topic. For each publication, the system shows its title, the type of document, and its date. If the user passes the cursor on the top of the title of the publication, the summary is also visualised (see the example in Figure 2 for the publication "*Reflections of five years of evaluating semantic search systems*"). In addition the title of the publication also constitutes a link to its source page in data.open.ac.uk.

[29] `http://www.jquery.com`



Enrico Motta

Position: Professor of Knowledge Technologies
Department: KMi
E-mail: e.motta
Phone extension: 53506

Related publications:

- Semantic search meets the Web
  AcademicArticle, 2008-08-12
- The usability of semantic search tools: a review
  AcademicArticle, 2007
- Folksonomy enrichment and search
  AcademicArticle, 2009
- Reflections on five years of evaluating semantic search systems

Reflections on five years of evaluating semantic search systems

Evaluations of semantic search systems are generally small scale and ad hoc due to the lack of appropriate resources such as test collections, agreed performance criteria and independent judgements of performance. By analysing our work in building and evaluating semantic tools over the last five years, we conclude that the growth of the semantic web led to an improvement in the available resources and the consequent robustness of performance assessments. We propose two directions for continuing evaluation work: the development of extensible evaluation benchmarks and the use of logging parameters for evaluating individual components of search systems.

AcademicArticle, 2008-08-12

- The usability of semantic search tools: a review
  AcademicArticle, 2007
- Reflections on five years of evaluating semantic search systems
  AcademicArticle, 2010

Fig. 2. The OU Expert Search system

5 Social Study

The Open University is a well-established institution in the United Kingdom, offering distance-learning courses covering a plethora of subject areas. A common problem when deciding on which course to study is choosing a course that is relevant and close to an individual's interests. One solution to this problem is to take advantage of existing profile information to bootstrap the decision process, in doing so leveraging information describing a person's interests upon which possible course for studying could be pursued.

Based on this thesis, Social Study[30] combines the popular social networking platform Facebook with the OU Linked Data service, the goal being to suggest Open University courses that share common themes with a user's interests.

5.1 Consumed Data

Social Study combines information extracted from Facebook with Linked Data offered by The Open University, where the former contains the profile information describing a given user - i.e. his/her interests, activities and '*likes*' - while the latter contains structured, machine-readable information describing courses offered by The Open University.

Combining the two information sources, in the form of a 'mashup', is performed using the following approach. First the user logs into the application – using Facebook Connect – and grants access to their information. The application then extracts the concepts that the user has expressed an interest in on his/her profile.

[30] http://www.matthew-rowe.com/SocialStudy

In Facebook such interests can be expressed through one of three means: *interests* - where the user explicitly states that they are interested in a given subject or topic; *activities* - where the user describes his/her hobbies and pastimes, and; *likes* - where the user identifies web pages that he/she is interested in that are then shared with the individual's social network.

This collection of concepts extracted from each of these interest facets provides the profile of the given user. To suggest courses from this collection, the OU SPARQL endpoint is queried for all courses on offer, returning the title and description of each course. This information is then compared with the profile of the user as follows: each of the concepts that make up the user's interest profile - in the form of ngrams - are compared against the description and title of each course, and the frequency of concepts matches is recorded for each course.

The goal of Social Study is to recommend relevant courses to the user given their interests, therefore the greater the number of concept matches, the greater the likelihood that the course is suited to the user. The courses are then ranked based on the number of overlapping concepts, allowing the user to see the most relevant courses at the top of the list, together with the list of concepts that informed the decision for the rank position of the course.

If for a moment we assume a scenario where Linked Data is not provided by the OU, then the function of Social Study could, in theory continue, by consuming information provided in an alternative form - given that the query component for the course information could be replaced by another process. However, the presence of Linked Data made the effort required to access and process courses information minimal. This work was an evolution of previous work that attempted to utilise the terms found in wall posts on Facebook in order to inform study partners and relevant courses. In evolving such work our intuition is that the user interest profile that is presented on such a platform can be bootstrapped to suggest courses, thereby avoiding the time-consuming task of manual profile population - from which course suggestions would then be derived.

5.2 System Implementation

The application is live and available online at the previously cited URL. It is built using PHP, and uses the Facebook PHP Software Development Kit (SDK)[31]. Authentication is provided via Facebook Connect,[32] enabling access to Facebook information via the Graph API. The ARC2 framework[33] is implemented to query the remote SPARQL endpoint containing The Open University's Linked Data, and parse the returned information accordingly.

5.3 Example and Screenshot

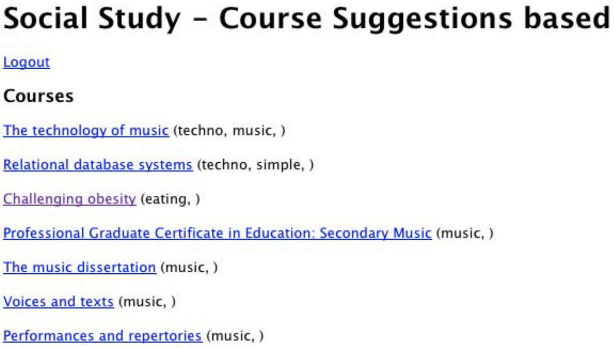
To ground the use of Social Study, Figure 3 shows an example screenshot from the application when recommending courses for Matthew Rowe – one of the

[31] https://github.com/facebook/php-sdk

[32] http://developers.facebook.com/docs/authentication

[33] http://arc.semsol.org

authors of this paper. The screenshot displays to the end user the order of courses together with the common interest concepts that their interest profile shares with those courses. The top-ranked course "*The technology of music*" matches the interest concepts *music* and *techno* that the user has specified in their profile. The greater the number of shared concepts with the course is, the greater the likelihood that the user will be interested in the course.



Social Study – Course Suggestions based

Logout

Courses

The technology of music (techno, music,)

Relational database systems (techno, simple,)

Challenging obesity (eating,)

Professional Graduate Certificate in Education: Secondary Music (music,)

The music dissertation (music,)

Voices and texts (music,)

Performances and repertories (music,)

Fig. 3. Social Study showing the top ranked courses together with the interest concepts

6 Linked OpenLearn

The Open University offers a set of free learning material through the OpenLearn website.[34] Such material cover various topics ranging from Arts[35], to Sciences and Engineering.[36] In addition to that, the OU has other learning resources published in the form of Podcasts, along with courses offered at specific presentations during the year. While all these resources are accessible online, connections are not always explicitly available, making it hard for students to easily exploit all the available resources. For example, while there exists a link between specific Podcasts and related courses, such links do not exist between OpenLearn units and Podcasts. This leaves it to the user to infer and find the appropriate and relevant material to the topic of interest.

Linked OpenLearn[37] is an application that enables exploring linked courses, Podcasts and tags to OpenLearn units. It aims to facilitate the browsing experience for students, who can identify on the spot relevant material without leaving the OpenLearn page. With this in place, students are able, for example, to easily find a linked Podcast, and play it directly without having to go through the Podcast website.

[34] http://openlearn.open.ac.uk

[35] OpenLearn unit example in Arts:
http://data.open.ac.uk/page/openlearn/a216_1

[36] A list of units and topics is available at: http://openlearn.open.ac.uk/course

[37] http://fouad.zablith.org/apps/linkedopenlearn

6.1 Consumed Data

Linked OpenLearn relies on The Open University's Linked Data to achieve what was previously considered very costly to do. Within large organisations, it's very common to have systems developed by different departments, creating a set of disconnected data silos. This was the case of Podcasts and OpenLearn units at the OU. While courses were initially linked to both Podcasts and OpenLearn in their original repositories, it was practically hard to generate the links between Podcasts and OpenLearn material. However, with the deployment of Linked Data, such links are made possible through the use of coherent and common URIs of represented entities.

To achieve our goals of generating relevant learning material, we make use of the courses, Podcasts, and OpenLearn datasets in data.open.ac.uk. As a first step, while the user is browsing an OpenLearn unit, the system identifies the unique reference number of the unit from the URL. Then this unique number is used in the query passed to the OU Linked Data SPARQL endpoint (http://data.open.ac.uk/query), to generate the list of related courses including their titles and links to the study at the OU pages.

In the second step, another query is sent to retrieve the list of Podcasts related to the courses fetched above. At this level we get the Podcasts' titles, as well as their corresponding downloadable media material (e.g., video or audio files), which enable users to play the content directly within the OpenLearn unit page. Finally the list of related tags are fetched, along with an embedded query that generates the set of related OpenLearn units, displayed in a separate window. The user at this level has the option to explore a new unit, and the corresponding related entities will be updated accordingly. The application is still a prototype, and there is surely room for further data to extract. For example, once the library catalogue is made available, a much richer interface can be explored by students with related books, recordings, computer files, etc.

6.2 System Implementation

We implemented the Linked OpenLearn application in PHP, and used the ARC2 library to query the OU Linked Data endpoint. To visualise the data on top of the web page, we relied on the jQuery User Interface library,[38] and used the dialog windows for displaying the parsed SPARQL results. The application is operational at present, and is launched through a Javascript bookmarklet, which detects the OpenLearn unit that the user is currently browsing, and opens it in a new iFrame, along with the linked entities visualised in the jQuery boxes.

6.3 Example and Screenshot

To install the application, the user has to drag the applications' bookmarklet[39] to the browser's toolbar. Then, whenever viewing an OpenLearn unit, the user

[38] `http://www.jqueryui.com`

[39] The bookmarklet is available at: `http://fouad.zablith.org/apps/linkedopenlearn`, and has been tested in Firefox, Safari and Google Chrome.

clicks on the bookmarklet to have the related entities displayed on top of the unit page. Figure 4 illustrates one arts related OpenLearn unit (referenced earlier), with the connected entities displayed on the right, and a running Podcast selected from the "Linked Podcasts" window. The user has the option to click on the related course to go directly to the course described in the Study at the OU webpage, or click on linked tags to see the list of other related OpenLearn units, which can be browsed within the same window.

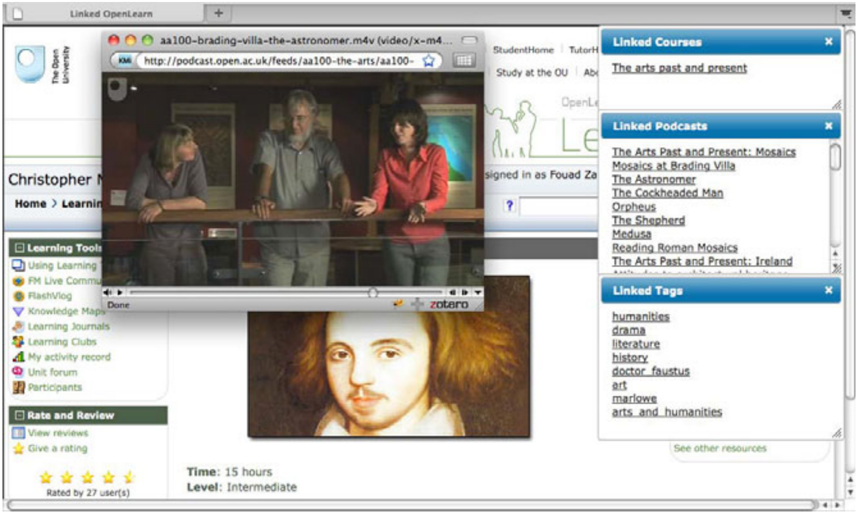


Fig. 4. Linked OpenLearn screenshot

7 Conclusions

In this section we report on our experiences when generating and exploiting LOD within the context of an educational institution. Regarding our experience on transforming information distributed in several OU repositories and exposing it as LOD, the process complexity was mainly dependent on the datasets in terms of type, structure and cleanliness. Initially, before any data transformation can be done, it was required to decide on the vocabulary to use. This is where the type of data to model plays a major role. With the goal to reuse, as much as possible, already existing ontologies, it was challenging to find the adequate ones for all our data. While some vocabularies are already available, for example to represent courses, it required more effort to model OU specific terminologies (e.g., at the qualifications level). To assure maximum interoperability, we chose to use multiple terminologies (when available) to represent the same entities. For example, courses are represented as modules from the AIISO ontology, and at the same time as courses from the Courseware ontology. Other factors that affected the transformation of the data are the structure and cleanliness of the data sources. During the transformation process, we faced many cases where duplication, and information not abiding to the imposed data structure, hampered the transformation stage. However, this initiated the need to generate the data following well-defined patterns and standards, in order to get easily processable data to add to the LOD.

Regarding our experience exploiting the data, we have identified three main advantages of relying on the LOD platform within the context of education. Firstly the exposure of all these material as free Web resources provides open opportunities for the development of novel and interesting applications like the three presented in this paper. The second main advantage is the structure provided by the data. This is apparent in the OU Expert Search system, where the different properties of articles are exploited to generate different ranking criteria, which when combined, provide much stronger support when finding the appropriate expertise. Finally, the links generated across the different educational resources have provided a new dimension to the way users can access, browse and use the provided educational resources. A clear example of this is the exploitation of LOD technology within the OpenLearn system, where OpenLearn units are now linked to courses and Podcasts, allowing students to easily find through a single Web page relevant material that could be of interest.

We believe that universities need to evolve the way they expose knowledge, share content and engage with learners. We see LOD as an exciting opportunity that can be exploited within the education community, especially by interlinking people and educational resources within and across institutions. This interlinking of information will facilitate the learning and investigation process of students and research staff, enhancing the global productivity and satisfaction of the academic community. We hope that, in the near future, more researchers and developers will embrace LOD approach, by creating new applications and learning from previous experiences to expose more and more educational data in a way that is directly linkable and reusable.

8 Future Work

The application of Linked Data within the OU has opened multiple research paths. Regarding the production of Linked Data, in addition to transforming the library records to LOD, the LUCERO team is currently working on connecting the OU's Reading Experience Database (RED)[40] to the Web of data. Such database aims to provide access and information about reading experiences around the world. It helps the readership for books issued in new editions for new audiences in different countries to be tracked. Its publication as LOD is an interesting example about how the integration of Linked Data technology can open new investigation paths to different research areas, in this case humanities.

Regarding the consumption of LOD, we envision, on the one hand, to enhance the three previously mentioned applications and, on the other hand to generate new applications as soon as more information is available and interconnected.

[40] http://www.open.ac.uk/Arts/reading

As example of the former, for the Social Study application we plan to extend the current approach for identifying common terms between users' interests and courses information, to instead utilise common more generic concepts. At present the use of merely interest concepts from within Facebook may be too specific to suggest relevant course to users - in many cases users expressed interest in a particular niche band that could instead have been replaced with a concept describing the genre of music. By instead using concepts, we believe that the suggested courses would be more accurate and suitable for studying. As an example of the latter, we aim to generate a search application over the RED database, able to display search results on an interactive map and link them not just to relevant records within the RED database, but also with relevant objects of the LOD cloud.

References

1. Bizer, C.: The emerging web of linked data. IEEE Int. Systems, 87–92 (2009)
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked data-the story so far. Int. J. Semantic Web Inf. Syst. 5(3), 1–22 (2009)
3. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space (2011)
4. Hepp, M.: GoodRelations: an ontology for describing products and services offers on the web. Knowledge Engineering: Practice and Patterns, 329–346 (2008)
5. Kobilarov, G., Scott, T., Raimond, Y., Oliver, S., Sizemore, C., Smethurst, M., Bizer, C., Lee, R.: Media meets semantic web how the bbc uses dbpedia and linked data to make connections, pp. 723–737 (2009)
6. Rowe, M.: Data.dcs: Converting legacy data into linked data. In: Linked Data on the Web Workshop, WWW 2010 (2010)
7. Yimam-Seid, D., Kobsa, A.: Expert-finding systems for organizations: Problem and domain analysis and the DEMOIR approach. Journal of Organizational Computing and Electronic Commerce 13(1), 1–24 (2003)
8. Zhu, J., Huang, X., Song, D., Rüger, S.: Integrating multiple document features in language models for expert finding. Knowledge and Information Systems 23(1), 29–54 (2010)

RDFPath: Path Query Processing on Large RDF Graphs with MapReduce

Martin Przyjaciel-Zablocki, Alexander Schätzle,
Thomas Hornung, and Georg Lausen

Lehrstuhl für Datenbanken und Informationssysteme
Albert-Ludwigs-Universität Freiburg
Georges-Köhler-Allee 51
79110 Freiburg im Breisgau
{zablocki,schaetzl,hornungt,lausen}@informatik.uni-freiburg.de

Abstract. The MapReduce programming model has gained traction in different application areas in recent years, ranging from the analysis of log files to the computation of the RDFS closure. Yet, for most users the MapReduce abstraction is too low-level since even simple computations have to be expressed as Map and Reduce phases. In this paper we propose RDFPath, an expressive RDF path query language geared towards casual users that benefits from the scaling properties of the MapReduce framework by automatically transforming declarative path queries into MapReduce jobs. Our evaluation on a real world data set shows the applicability of RDFPath for investigating typical graph properties like shortest paths.

Keywords: MapReduce, RDFPath, RDF Query Languages, Social Network Analysis, Semantic Web.

1 Introduction

The proliferation of data on the Web is growing tremendously in recent years. According to Eric Schmidt, CEO of Google, more than five Exabyte of data are generated collectively *every two days*, which corresponds to the whole amount of data generated up to the year 2003[1]. Another example is Facebook with currently more than 500 million active users interacting with more than 900 million different objects like pages, groups or events.

In a Semantic Web environment this data is typically represented as a RDF graph [20], which is a natural choice for social network scenarios [21], thus facilitating exchange, interoperability, transformation and querying of data. However, management of large RDF graphs is a non-trivial task and single machine approaches are often challenged with processing queries on such graphs [30]. One solution is to use high performance clusters or to develop custom distributed systems that are commonly not very cost-efficient and also do not scale with respect to additional hardware [1,8,9].

[1] http://techonomy.com

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 50–64, 2012.

The MapReduce programming model introduced by Google in [8] runs on regular off-the-shelf hardware and shows desirable scaling properties, e.g. new computing nodes can easily be added to the cluster. Additionally, the distribution of data and the parallelization of calculations is handled automatically, relieving the developer from having to deal with classical problems of distributed applications such as the synchronization of data, network protocols or fault tolerance strategies. These benefits have led to the application of this programming model to a number of problems in different areas, where large data sets have to be processed [1,2,7]. One line of research is centered around the transformation of existing algorithms into the MapReduce paradigm [19], which is a time consuming process that requires substantial technical knowledge about the framework. More in line with the approach presented in this paper is the idea to use a declarative high-level language and to provide an automatic translation into a series of Map and Reduce phases as proposed in [15,29] for SPARQL and in [24] for Pig Latin, a data processing language for arbitrary data.

Contributions. In this paper we present RDFPath, a declarative path query language for RDF that by design has a natural mapping to the MapReduce programming model while remaining extensible. We also give details about our system design and implementation. By its intuitive syntax, RDFPath supports the exploration of graph properties such as shortest connections between two nodes in a RDF graph. We are convinced that RDFPath is a valuable tool for the analysis of social graphs, which is highlighted by our evaluation on a real-world data set based on user profiles crawled from Last.fm. The implementation of RDFPath is available for download from our project homepage[2].

Related Work. There is a large body of work dealing with query languages for (RDF) graphs considering various aspects and application fields [6,10,12,17,27,34]. Besides classical proposals for graphs as introduced in [27] and in [17] with RQL, there are also many proposals for specific RDF graph languages (cf. [6,10,12] for detailed surveys). Taking this into account, we extended the proposed comparison matrix for RDF query languages from [4,5] by two additional properties, namely the support for shortest path queries and aggregate functions, as well as the additional RDF query languages SPARQL [26], RPL [34], and RDFPath, as depicted in Table 1. For a more detailed description of the properties occurring in Table 1 the interested reader is referred to [5].

According to Table 1, RDFPath has a competitive expressiveness to other RDF query languages. For the missing diameter property, which is not considered in any of the listed languages, a MapReduce solution has been proposed in [16], regardless of a syntactically useful integration into any path query language. There are also further approaches to extend SPARQL with expressive navigational capabilities such as nSPARQL [25], (C)PSPARQL [3] as well as

[2] http://dbis.informatik.uni-freiburg.de/?project=DiPoS/RDFPath.html

Table 1. Comparison of RDF Query Languages (adapted from [4,5])

Property	**RQL**	**SeRQL, RDQL[3], Triple**	**N3**	**Versa**	**RxPath**	**RPL**	**SPARQL**		**RDFPath**
							1.0	**1.1**	
Adjacent nodes	±	±	±	±	×	√	√	√	√
Adjacent edges	±	±	×	×	×	×	√	√	√
Degree of a node	±	×	×	×	×	×	×	√	√
Path	×	×	×	×	±	±	×	±	±
Fixed-length Path	±	±	±	×	±	±	√	√	√
Distance between 2 nodes	×	×	×	×	×	×	×	×	±
Diameter	×	×	×	×	×	×	×	×	×
Shortest Paths	×	×	×	×	×	×	×	×	±
Aggregate functions	±	×	×	×	±	×	×	√	±

(×: not supported, ±: partially supported, √: fully supported)

property paths, that are a part of the proposal for SPARQL 1.1[4]. In contrast, we focus on path queries and study their implementation based on MapReduce. A more detailed discussion on SPARQL 1.0 and especially the current SPARQL 1.1 working draft can be found in the appendix.

Another area, which is related to our research, is the distributed processing of large data sets with MapReduce. *Pig* is a system for analyzing large data sets, consisting of the high-level language *Pig Latin* [24] that is automatically translated into MapReduce jobs. Furthermore there are serveral recent approches for evaluating SPARQL queries with MapReduce [15,22,29]. However, because of the limited navigational capabilities of SPARQL [25], as opposed to RDFPath, these approaches do not offer a sufficient functionality to support a broad range of analysis tasks for RDF graphs.

Besides the usage of a general purpose MapReduce cluster, some systems rely on a specialized computer cluster. Virtuoso Cluster Edition [9] is a cluster extension of the RDF Store Virtuoso and BigOWLIM[5] is a RDF database engine with extensive reasoning capabilities, both allowing to store and process billions of triples. In [32] the authors propose an extension of Sesame for querying distributed RDF repositories. However, such specialized clusters have the disadvantage that they require individual infrastructures, whereas our approach is based on a general framework that can be used for different purposes.

Paper Structure. Section 2 provides a brief introduction to the MapReduce framework. Section 3 introduces the RDFPath language, while Section 4 discusses the components of the implemented system and the evaluation of RDFPath queries. Section 5 presents our system evaluation based on a real-world data set and Section 6 concludes this paper with an outlook on future work.

[3] In [14] the authors describe how to extend RDQL to support aggregates.

[4] `http://www.w3.org/TR/sparql11-query`

[5] `http://www.ontotext.com/owlim`

2 MapReduce

The MapReduce programming model was originally introduced by Google in 2004 [8] and enables scalable, fault tolerant and massively parallel calculations using a computer cluster. The basis of Google's MapReduce is the distributed file system GFS [11] where large files are split into equal sized blocks, spread across the cluster and fault tolerance is achieved by replication. The workflow of a MapReduce program is a sequence of MapReduce jobs each consisting of a *Map* and a *Reduce* phase separated by a so-called *Shuffle* & *Sort* phase. A user has to implement the *map* and *reduce functions* which are automatically executed in parallel on a portion of the data. The Mappers invoke the map function for every record of their input data set represented as a key-value pair. The map function outputs a list of new intermediate key-value pairs which are then sorted according to their key and distributed to the Reducers such that all values with the same key are sent to the same Reducer. The reduce function is invoked for every distinct key together with a list of all according values and outputs a list of values which can be used as input for the next MapReduce job. The signatures of the map and reduce functions are therefore as follows:

```
map:    (inKey, inValue) -> list(outKey, intermediateValue)
reduce: (outKey, list(intermediateValue)) -> list(outValue)
```

3 RDFPath

A RDF data set consists of a set of RDF triples in the form <subject, predicate, object> that can be interpreted as "*subject* has the property *predicate* with value *object*". It is possible to represent a RDF data set as directed, labeled graph where every triple corresponds to an edge (predicate) from one node (subject) to another node (object). For clarity of presentation, we use a simplified RDF notation without URI prefixes in the following. Strings and numbers are mapped to their corresponding datatypes in RDF.

Executing path queries on very large RDF data sets like social network graphs with billions of entries is a non-trivial task that typically requires many resources and computational power [1,8,9,21,30]. RDFPath is a declarative RDF path query language, inspired by XPath and designed especially with regard to the MapReduce model. A query in RDFPath is composed by a sequence of *location steps* where the output of the i^{th} location step is used as input for the $(i+1)^{th}$ location step. Conceptually, a location step adds one or more additional edges and nodes to an intermediate path that can be restricted by filters. The result of a query is a set of paths, consisting of edges and nodes of the given RDF graph. In the following we give an example-driven introduction to RDFPath.

3.1 RDFPath by Example

Start Node. The start node is the first part of a RDFPath query, separated by `"::"` from the rest of the query and specifies the starting point for the evaluation of a path query as shown in Query 1. Using the symbol `"*"` indicates an arbitrary start node where every subject with the denoted predicate of the first location step is considered (see Query 2).

$$\texttt{Chris :: knows} \tag{1}$$

$$\texttt{* :: knows} \tag{2}$$

Location Step. Location steps are the basic navigational component in RDFPath, specifying the next edge to follow in the query evaluation process. The usage of multiple location steps, separated by `">"`, defines the order as well as the amount of edges followed by the query (Query 3). If the same edge is used in several consecutive location steps one can use an abbreviation by specifying the number of iterations within parentheses as shown in Query 4. Instead of specifying a fixed edge, the symbol `"*"` can be used to follow an arbitrary edge as illustrated in Query 5 that determines all adjacent edges and nodes of Chris.

$$\texttt{Chris :: knows > knows > age} \tag{3}$$

$$\texttt{Chris :: knows (2) > age} \tag{4}$$

$$\texttt{Chris :: *} \tag{5}$$

Filter. Filters can be specified within any location step using square brackets. There are two types of filters to constrain the value (Queries 6, 7) or the properties (Query 8) of a node reached by the location step. Multiple filters are specified in a sequence and a path has to satisfy all filters. If a node does not have the desired property, the filter evaluates to `false`. Up to now, the following filter expressions are applicable: `equals(), prefix(), suffix(), min(), max()`.

$$\texttt{Chris :: knows > age [min(18)] [max(67)]} \tag{6}$$

$$\texttt{Chris :: * > * [equals('Peter')]} \tag{7}$$

$$\texttt{Chris :: knows [age = min(30)] [country = prefix('D')] > name} \tag{8}$$

Bounded Search. This type of query starts with a fixed node and computes the shortest paths between the start node and all reachable nodes within a user-defined bound. For this purpose we extend the notation of the previously introduced abbreviations with an optional symbol `"*"`. While the abbreviations indicate a fixed length, the `"*"` symbol indicates to use the number as upper bound for the maximum search depth. As an example, in Query 9 we search for all German people with a maximum distance of three to Chris.

$$\texttt{Chris :: knows [country = equals('DE')] (*3)} \tag{9}$$

Bounded Shortest Path. This type of query computes the shortest path between two nodes in the graph with a user-defined maximum distance. As we are often interested in the length of the path, the query outputs the shortest distance and the corresponding path between two given nodes. To do this one has to extend a bounded search query with a final `distance()` function specifying the target node as shown in Query 10.

$$\texttt{Chris :: knows (*3).distance('Peter')} \tag{10}$$

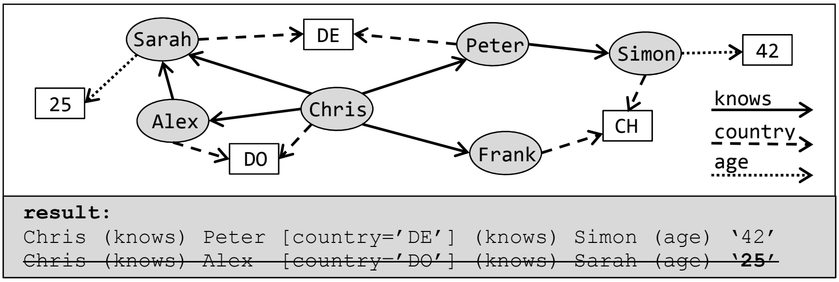
Aggregation Functions. It is possible to count the number of resulting paths for a query (Query 11 calculates the degree of Chris) or to apply some aggregation functions to the last nodes of the paths, respectively. The following functions are available: `count()`, `sum()`, `avg()`, `min()` and `max()`. It should be noted that aggregation functions can only be applied to nodes of numeric type (e.g. `integer` or `double`) as shown in Query 12.

$$\texttt{Chris :: *.count()} \tag{11}$$

$$\texttt{Chris :: knows > age.avg()} \tag{12}$$

Example. Figure 1 shows the evaluation of the last location step of Query 13 on the corresponding RDF graph. The second path is rejected as the age of Sarah does not satisfy the filter condition.

$$\texttt{Chris :: knows [country=prefix('D')] > knows > age [min(30)]} \tag{13}$$



result:
Chris (knows) Peter [country='DE'] (knows) Simon (age) '42'
~~Chris (knows) Alex [country='DO'] (knows) Sarah (age) '25'~~

Fig. 1. RDFPath Example

3.2 Expressiveness

In this section we will evaluate the expressiveness of RDFPath w.r.t. the properties listed in Table 1. A detailed discussion can be found on our project homepage[6] and in [28]. Query 5 shows an example for the calculation of all *adjacent edges* and *nodes* of a node by using the symbol `"*"` instead of specifying a fixed edge. Query 11 calculates the *degree of a node* by applying the aggregation function `count()` on the resulting paths and Query 7 gives all *paths with a fixed length* of two from Chris to Peter by specifying two location steps with arbitrary edges. The properties *path*, *distance between 2 nodes* and *shortest paths* are only partially supported by RDFPath because in general to answer these properties one has to calculate paths of arbitrary length where RDFPath only supports paths of a fixed maximum length. Furthermore aggregation functions are partially supported as they can only be applied in the last location step of a query.

[6] `http://dbis.informatik.uni-freiburg.de/?project=DiPoS/RDFPath.html`

4 Query Evaluation

We implemented RDFPath based on the well-known *Apache Hadoop Framework*, an open source implementation of Google's MapReduce and GFS. Our system loads the considered RDF graph into the Hadoop Distributed File System (HDFS) once in advance, translates RDFPath queries into a sequence of MapReduce jobs, executes them in the framework and stores the results again in HDFS. A location step in RDFPath mostly follows a fixed edge (predicate) which means that only a portion of the RDF graph has to be considered in many cases. In these cases, it is advantageous to read only those triples concerning the relevant edge which can be achieved by partitioning the triples of the RDF graph according to their predicates. This principle is also knows as *vertical partitioning* [2] and forms the basis of our data model. Hence an input RDF graph is loaded in advance to apply vertical partitioning and store resulting partitions in HDFS. Certainly, in the case of a not fixed edge ("*" symbol), all partitions must be considered and we cannot benefit from this strategy.

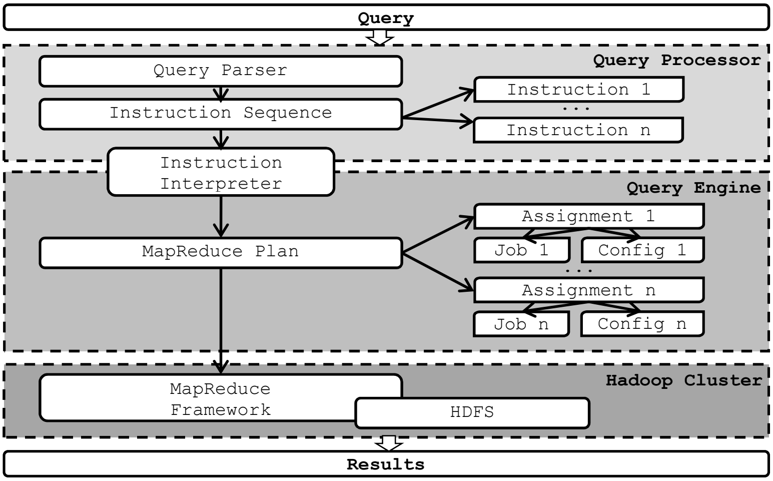


Fig. 2. Query Processing

The Query Processor parses the query and generates a general execution plan that consists of a sequence of instructions where each instruction describes e.g. the application of a filter, join or aggregation function. In the next step, the general execution plan is mapped to a specific MapReduce plan that consists of a sequence of MapReduce assignments. An assignment encapsulates the specific MapReduce job together with a job configuration. The Query Engine runs the MapReduce jobs in sequence, collects information about the computation process like time and storage utilization and cleans up temporary files. A schematic representation of this procedure is shown in Figure 2.

Fault-tolerance, i.e. relaunching failed tasks, is managed by the Hadoop framework automatically. Currently there are no join-ordering optimizations implemented to determine the best join execution order as proposed in [32], for example. A query in RDFPath is processed sequentially from left to right. Although queries in RDFPath do not need to have a fixed start node, this is often advisable as arbitrary start nodes dramatically increase the number of intermediate results in general. For queries with a fixed start node, a processing order from left to right can often cut down the costs for processing a sequence of joins, as it usually corresponds to the most selective join-order. In other cases, e.g. fixed start node combined with fixed end node for computing shortest paths, it is likely that join-ordering optimizations could have a significant impact on query evaluation time and space [32].

4.1 Mapping of Location Steps to MapReduce Jobs

A query in RDFPath is composed of a sequence of location steps that is translated into a sequence of MapReduce jobs automatically. As illustrated in Figure 3 a location step corresponds to a join in MapReduce between an intermediate set of paths and the corresponding RDF graph partition. Joins are implemented as so-called *Reduce-Side-Joins* since the assumption of the more efficient *Map-Side-Joins* that both inputs must be sorted is not fulfilled in general. The principles of Reduce-Side-Joins can be looked up in [19,33]. Filters are applied in the Map phase by rejecting all triples that do not satisfy the filter conditions and aggregation functions are computed in parallel in the Reduce phase of the last location step. The computation of shortest paths is based on a parallel breadth-first search approach and requires at most one additional MapReduce job for selecting shortest paths. This selection is usually applied in the subsequent location step. If there is no subsequent location step, an additional MapReduce job becomes necessary. We also implemented a mechanism to detect cycles when extending an intermediate path where the user can decide at runtime whether (1) cycles are allowed, (2) only allowed if the cycle contains two or more distinct edges or (3) not allowed at all. Considering Figure 3 the given query requires two joins and is therefore mapped into a MapReduce plan that consists of two MapReduce jobs. While the first job computes all friends of "Chris" that can be reached by following the edge `knows` at most two times, the second MapReduce job follows the edge `country` and restricts the value to "DE".

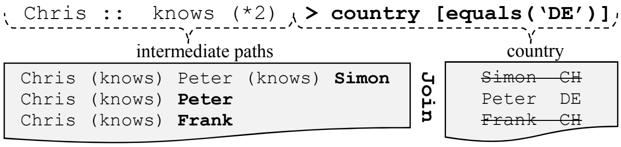


Fig. 3. Joins and Location Steps

5 Evaluation

We evaluated our implementation on two different data sources to investigate the scalability behavior. First, we used artificial data produced by the SP^{2}Bench generator [31] which allows to generate arbitrary large RDF documents that contain bibliographic information about synthetic publications. The generated RDF documents contained up to 1.6 billion RDF triples. Second, we collected 225 million RDF triples of real world data from the online music service Last.fm that are accessible via a public API. Due to space limitations we only discuss some results for the Last.fm dataset, which is a more appropriate choice for path queries and can also be interpreted in a more intuitive way. Figure 4 illustrates the dataset.

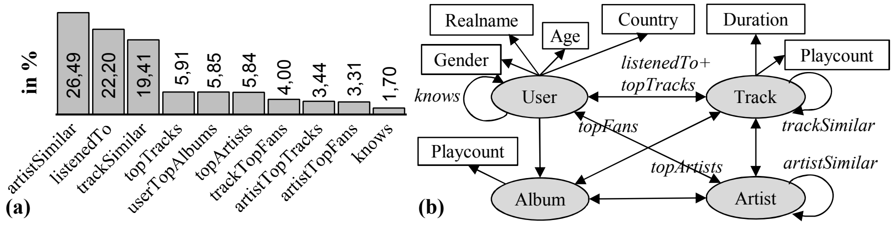


Fig. 4. (a) Histogram of Last.fm data (b) Simplified RDF graph of the Last.fm data set. The missing edge labels are named like the target nodes. In the case of ambiguity, the edge label is extended by the type of source (e.g. `trackPlaycount` and `albumPlaycount`).

We used a cluster of ten Dell PowerEdge R200 servers connected via a gigabit network and Cloudera's Distribution for Hadoop 3 Beta (CDH3). Each server had a Dual Core 3,16 GHz processor, 4 GB RAM and 1 TB harddisk. One of the servers was exclusively used to distribute the MapReduce jobs (Jobtracker) and store the metadata of the file system (Namenode). Query 1 to 3 were evaluated on a fixed cluster size of 9 nodes with varying dataset sizes, whereas Query 4 and 5 used a fixed dataset size of 200 million triples while varying the number of cluster nodes.

Query 1. Starting from a given track this query determines the album name for all similar tracks that can be reached by following the edge `trackSimilar` at most four times. The overall execution times of this query are shown in the left diagram of Figure 5 and exhibit a linear scaling behavior in the size of the graph. Furthermore it turns out that this is also the case for the amount of transferred data (SHUFFLE), intermediate data (LOCAL) and data stored in HDFS. These values are shown in the right diagram of Figure 5. We conclude that the execution time is mainly influenced by the number of intermediate results stored locally as well as the transferred data between the machines.

Query 2. Starting from all tracks of Michael Jackson that are on the album "Thriller" the query determines all similar tracks that have a minimum duration

```
The Disco Boys - I Surrender :: trackSimilar(*4) > trackAlbum
```

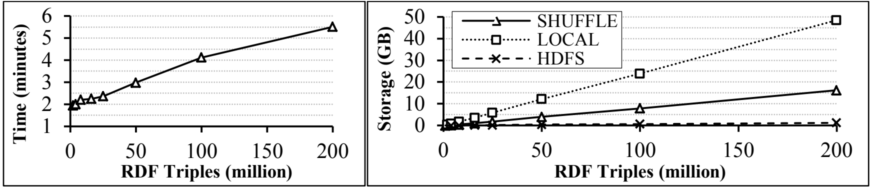


Fig. 5. Query 1

of 50 seconds. The last location step then looks for the top fans of these tracks who live in Germany. The idea behind this query was to have a look at the impacts of using filters to reduce the amount of intermediate results. The number of results to the query and therefore the used HDFS storage do not increase significantly with the size of the graph as the tracks of the album "Thriller" are fixed. This also explains the execution times of the query as illustrated in the left diagram of Figure 6 and confirms that the execution time is mainly determined by the amount of intermediate results.

```
Michael_Jackson :: artistTracks [trackAlbum=equals('Michael_Jackson_-_Thriller')]
 > trackSimilar[trackDuration=min(50000)] > trackTopFans[country=equals('DE')]
```

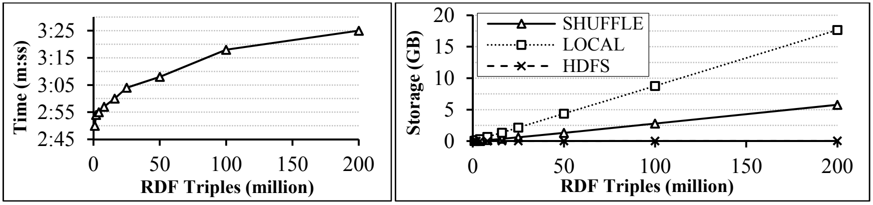


Fig. 6. Query 2

Query 3. These queries determine the friends of Chris reached by following an increasing number of edges. The first query starts by following the edge `knows` once and the last query ends by following the edge `knows` at most ten times. This corresponds to the computation of the *Friend of a Friend*[7] paths starting from Chris with an increasing maximum distance. The left chart of Figure 7 illustrates the percentage of reached people, in accordance to the maximum Friend of a Friend distance, where the total percentage represents all reachable people. Starting with a fixed person we can reach over 98% of all reachable persons by following the edge `knows` seven times which corresponds to the well-known *six degrees of separation* paradigm [18]. The right chart of Figure 7 shows the execution times depending on the maximum Friend of a Friend distance. We

[7] `http://www.foaf-project.org`

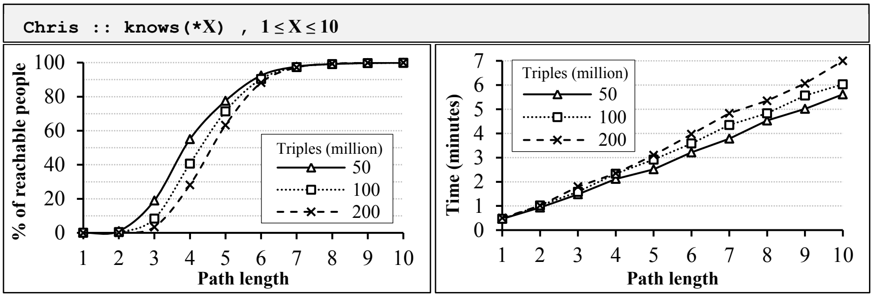


Fig. 7. Query 3

can observe a linear scaling behaviour that is mainly determined by the number of joins rather than computation and data transfer time.

Query 4. & 5. Query 4 is a kind of recommendation query that gives, for every user, those tracks that are similar to the tracks the user listened to. On the other hand, Query 5 is an analytical query where we want to know, for every artist, where the top Fans of a similar artist come from. The execution times of both queries for a fixed input size of 200 million triples and a variable number of nodes in the cluster are shown in the left diagram of Figure 8. We can observe that the overall execution times for Query 4 as well as for Query 5 improve with the number of nodes but the benefit of an additional node decreases continuously which is an expectable behaviour of the MapReduce framework. The storage utilization for both queries is given in the right diagram of Figure 8. The values for different numbers of nodes were almost equal in size, with a maximum deviation of 1% for Shuffle & HDFS storage and 9% for local storage.

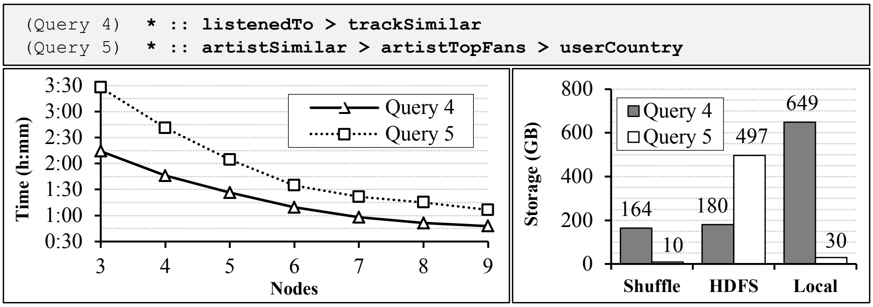


Fig. 8. Query 4 & 5

Results. Our evaluation shows that RDFPath allows to express and compute interesting graph issues such as Friend of a Friend queries, small world properties like six degrees of separation or the *Erdös number*[8] on large RDF graphs.

[8] http://www.oakland.edu/enp

The execution times for the surveyed queries on real-world data from Last.fm scale linear in the size of the graph where the number of joins as well as the amount of data, that must be stored (local/HDFS) and transferred over the network, determine the complexity of a query. Taking this into account it is promising to observe an almost constant storage utilization with an increasing number of nodes. On the other hand, adding additional nodes can improve the overall executions time significantly, which shows that RDFPath benefits from the horizontal scaling properties of MapReduce.

6 Conclusion

The amount of available Semantic Web data is growing constantly, calling for solutions that are able to scale accordingly. The RDF query language RDFPath, that is presented in this paper, was designed with this constraint in mind and combines an intuitive syntax for path queries with an effective execution strategy using MapReduce. Our evaluation confirms that both large RDF graphs can be handled while scaling linear with the size of the graph and that RDFPath can be used to investigate graph properties such as a variant of the famous six degrees of separation paradigm typically encountered in social graphs.

As future work we plan to extend RDFPath with more powerful language constructs geared towards the analysis of social graphs, e.g. to express the full list of desiderata stated in [21]. In parallel, we are optimizing our implementation on the system level by incorporating current results for the efficient computation of joins with MapReduce [7,23].

References

1. Abadi, D.J.: Tradeoffs between Parallel Database Systems, Hadoop, and HadoopDB as Platforms for Petabyte-Scale Analysis. In: Gertz, M., Ludäscher, B. (eds.) SSDBM 2010. LNCS, vol. 6187, pp. 1–3. Springer, Heidelberg (2010)
2. Abadi, D.J., Marcus, A., Madden, S., Hollenbach, K.J.: Scalable Semantic Web Data Management Using Vertical Partitioning. In: VLDB, pp. 411–422 (2007)
3. Alkhateeb, F., Baget, J.F., Euzenat, J.: Extending sparql with regular expression patterns (for querying rdf). J. Web Sem. 7(2), 57–73 (2009)
4. Angles, R., Gutierrez, C.: Querying RDF Data from a Graph Database Perspective. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 346–360. Springer, Heidelberg (2005)
5. Angles, R., Gutierrez, C., Hayes, J.: RDF Query Languages Need Support for Graph Properties. Tech. Rep. TR/DCC-2004-3, University of Chile (June 2004)
6. Bailey, J., Bry, F., Furche, T., Schaffert, S.: Web and Semantic Web Query Languages: A Survey. In: Eisinger, N., Małuszyński, J. (eds.) Reasoning Web. LNCS, vol. 3564, pp. 35–133. Springer, Heidelberg (2005)
7. Blanas, S., Patel, J.M., Ercegovac, V., Rao, J., Shekita, E.J., Tian, Y.: A Comparison of Join Algorithms for Log Processing in MapReduce. In: SIGMOD Conference, pp. 975–986 (2010)
8. Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. In: OSDI, pp. 137–150 (2004)

9. Erling, O., Mikhailov, I.: Towards Web Scale RDF. In: Proc. SSWS (2008)
10. Furche, T., Linse, B., Bry, F., Plexousakis, D., Gottlob, G.: RDF Querying: Language Constructs and Evaluation Methods Compared. In: Barahona, P., Bry, F., Franconi, E., Henze, N., Sattler, U. (eds.) Reasoning Web 2006. LNCS, vol. 4126, pp. 1–52. Springer, Heidelberg (2006)
11. Ghemawat, S., Gobioff, H., Leung, S.T.: The Google File System. In: Proc. SOSP, pp. 29–43 (2003)
12. Haase, P., Broekstra, J., Eberhart, A., Volz, R.: A Comparison of RDF Query Languages. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 502–517. Springer, Heidelberg (2004)
13. Harris, S., Seaborne, A.: SPARQL 1.1 Query Language. W3C Working Draft (May 2011), http://www.w3.org/TR/sparql11-query/
14. Hung, E., Deng, Y., Subrahmanian, V.S.: RDF Aggregate Queries and Views. In: ICDE, pp. 717–728 (2005)
15. Husain, M.F., Khan, L., Kantarcioglu, M., Thuraisingham, B.: Data Intensive Query Processing for Large RDF Graphs Using Cloud Computing Tools. In: Proc. CLOUD, pp. 1–10. IEEE (2010)
16. Kang, U., Tsourakakis, C.E., Faloutsos, C.: PEGASUS: A Peta-Scale Graph Mining System. In: ICDM, pp. 229–238 (2009)
17. Karvounarakis, G., Alexaki, S., Christophides, V., Plexousakis, D., Scholl, M.: RQL: A Declarative Query Language for RDF. In: WWW, pp. 592–603 (2002)
18. Leskovec, J., Horvitz, E.: Planetary-Scale Views on a Large Instant-Messaging Network. In: Proc. WWW 2008, pp. 915–924 (2008)
19. Lin, J., Dyer, C.: Data-intensive text processing with MapReduce. Synthesis Lectures on Human Language Technologies 3(1), 1–177 (2010)
20. Manola, F., Miller, E.: RDF Primer (2004), http://www.w3.org/TR/rdf-primer/
21. Martín, M.S., Gutierrez, C.: Representing, Querying and Transforming Social Networks with RDF/SPARQL. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 293–307. Springer, Heidelberg (2009)
22. Myung, J., Yeon, J., Lee, S.: SPARQL Basic Graph Pattern Processing with Iterative MapReduce. In: Proc. MDAC 2010, pp. 1–6. ACM (2010)
23. Okcan, A., Riedewald, M.: Processing Theta-Joins using MapReduce. In: SIGMOD Conference, pp. 949–960 (2011)
24. Olston, C., Reed, B., Srivastava, U., Kumar, R., Tomkins, A.: Pig Latin: A Not-So-Foreign Language for Data Processing. In: SIGMOD, pp. 1099–1110 (2008)
25. Pérez, J., Arenas, M., Gutierrez, C.: nSPARQL: A navigational language for RDF. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 66–81. Springer, Heidelberg (2008)
26. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and Complexity of SPARQL. ACM Trans. Database Syst. 34(3) (2009)
27. Pratt, T.W., Friedman, D.P.: A Language Extension for Graph Processing and Its Formal Semantics. Commun. ACM 14(7), 460–467 (1971)
28. Przyjaciel-Zablocki, M.: RDFPath: Verteilte Analyse von RDF-Graphen. Master's thesis, Albert-Ludwigs-Universität Freiburg (2010)
29. Schätzle, A., Przyjaciel-Zablocki, M., Lausen, G.: PigSPARQL: Mapping SPARQL to Pig Latin. In: Proceedings of the International Workshop on Semantic Web Information Management, SWIM 2011, pp. 4:1–4:8. ACM (2011)

30. Schmidt, M., Hornung, T., Küchlin, N., Lausen, G., Pinkel, C.: An Experimental Comparison of RDF Data Management Approaches in a SPARQL Benchmark Scenario. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 82–97. Springer, Heidelberg (2008)
31. Schmidt, M., Hornung, T., Lausen, G., Pinkel, C.: SP2Bench: A SPARQL Performance Benchmark. In: ICDE, pp. 222–233 (2009)
32. Stuckenschmidt, H., Vdovjak, R., Broekstra, J., Houben, G.J.: Towards distributed processing of RDF path queries. Int. J. Web Eng. Technol. 2(2/3), 207–230 (2005)
33. White, T.: Hadoop: The Definitive Guide, 1st edn. O'Reilly (2009)
34. Zauner, H., Linse, B., Furche, T., Bry, F.: A RPL Through RDF: Expressive Navigation in RDF Graphs. In: Hitzler, P., Lukasiewicz, T. (eds.) RR 2010. LNCS, vol. 6333, pp. 251–257. Springer, Heidelberg (2010)

A Comparison with SPARQL

SPARQL 1.0 is the W3C recommended query language for RDF. Compared with RDFPath, we can note the following interesting relations: First of all, it is important to mention that SPARQL 1.0 is a general purpose query language, designed for a wide range of analysis tasks, whereas RDFPath focuses on expressing path queries. Thus, SPARQL 1.0 provides only limited navigation capabilities (e.g. no abbreviation for following the same edge). Furthermore, both approaches differ in the kind of output. The result of a query in RDFPath is a set of paths in contrast to a set of variable mappings in SPARQL. Next, SPARQL 1.0 does not support aggregate functions and shortest path expressions, whereas RDFPath supports both.

The SPARQL 1.1 [13] working draft addresses, among other things, some of the issues mentioned above: Property paths, introduced with SPARQL 1.1, add support for navigational queries similar to RDFPath but with some additional features like inverse, negated and alternative paths. They also allow to abbreviate occurrences of edges in more detail and follow paths of arbitrary length. However, property paths do not provide access to the whole path, but only to the first and last node. Accordingly, the result in SPARQL 1.1 is projected to the variables given in the query, i.e. the first and the last node of a path. In contrast, RDFPath provides access to the whole path in such a way that it is possible to express filter conditions on arbitrary location steps and to emit whole paths with all intermediate steps as output. Furthermore, predicates in property paths are always fixed, i.e. variable edges are not expressible. RDFPath provides a possibility to follow arbitrary edges and supports three types of cycle treatment. Expressing a query that determines the shortest path between two nodes is also not possible with property paths. Certainly, SPARQL 1.1 allows multiple property path expressions in one query, whereas queries in RDFPath must be composed of a single sequence of location steps. The following two examples illustrate how queries could be expressed with SPARQL 1.0, SPARQL 1.1 and RDFPath, despite of the different kind of output.

Example 1.	*Friend of a Friend query starting with ’Allen’*

SPARQL 1.0	`SELECT ?tmp1, ?tmp2, ?tmp3, ?tmp4, ?tmp5`
	`WHERE { Allen knows ?tmp1 . ?tmp1 knows ?tmp2 .`
	`        ?tmp2 knows ?tmp3 . ?tmp3 knows ?tmp4 .`
	`        ?tmp5 knows ?tmp5 }`

SPARQL 1.1	`SELECT  ?tmp { Allen knows{5} ?tmp }`
RDFPath	`Allen :: knows(5)`

Example 2.	*Friend of a Friend query with two path restrictions*

SPARQL 1.0
&
SPARQL 1.1
	`SELECT ?tmp1, ?tmp2, ?age`
	`WHERE { Allen knows ?tmp1 . ?tmp1 age ?age .`
	`        FILTER (?age >= 20)`
	`        ?tmp1 knows ?tmp2 . ?tmp2 country ’DE’}`

RDFPath	`Allen :: knows[age=min(20)] > knows[country=equals(’DE’)`

A Novel Metric for Information Retrieval in Semantic Networks

Joshua L. Moore[1,2], Florian Steinke[1], and Volker Tresp[1]

[1] Siemens AG, Corporate Technology, München, Germany
[2] Cornell University, Ithaca, NY
jlmo@cs.cornell.edu,{Florian.Steinke,Volker.Tresp}@siemens.com

Abstract. We propose a novel graph metric for semantic entity-relationship networks. The metric is used for solving two tasks. First, given a semantic entity-relationship graph, such as for example DBpedia, we find relevant neighbors for a given query node. This could be useful for retrieving information relating to a specific entity. Second, we search for paths between two given nodes to discover interesting links. As an example, this can be helpful to analyze the various relationships between Albert Einstein and Niels Bohr. Compared to using the default step metric our approach yields more specific and informative results, as we demonstrate using two semantic web datasets. The proposed metric is defined via paths that maximize the log-likelihood of a restricted round trip and can intuitively be interpreted in terms of random walks on graphs. Our distance metric is also related to the commute distance, which is highly plausible for the described tasks but prohibitively expensive to compute. Our metric can be calculated efficiently using standard graph algorithms, rendering the approach feasible for the very large graphs of the semantic web's linked data.

Keywords: Entity-relationship graph, information retrieval, random walk, commute distance, graph metric, path finding.

1 Introduction

Large entity-relationship (ER) graphs have recently become available on the semantic web. Sources like DBpedia (Auer et al., 2008), YAGO (Suchanek et al., 2007), OpenCyc[1] or Linked Life Data[2] (Momtchev et al., 2009) now encode useful information on a large scale. Simple and efficient information retrieval methods for these data sources are a pressing need.

Here, we focus on two query tasks. First, given a query node in the graph, e.g. a person or a category in DBpedia, which other nodes in the graph represent entities that are useful in the context of the given query node? Answers might be other concepts that could be used to refine or extend interactive search.

[1] http://www.cyc.com/opencyc

[2] http://linkedlifedata.com/

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 65–79, 2012.

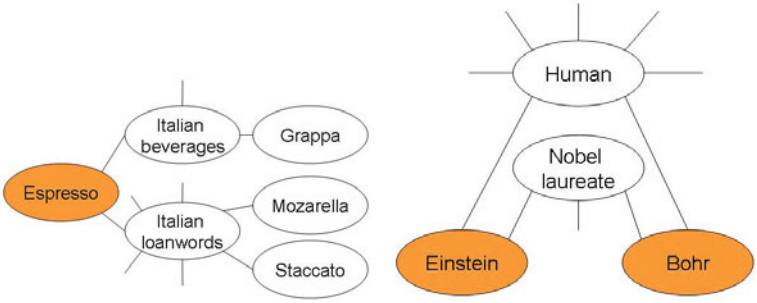


Fig. 1. Stylized examples of the two proposed tasks (data taken from DBpedia) and the associated challenges. In the left example, “Espresso” might be considered to be semantically closer to ”Grappa”, if compared to “Mozzarella” and “Staccato”. But since “Italian loanwords” has a higher degree than “Italian beverages”, ranking algorithms, e.g., based on PageRank, might conclude differently. A similar phenomenon occurs on the right side where the link via “Nobel laureates” is more informative than the link via “Human”, despite the fact that “Human” has a higher degree than “Nobel laureate”.

In the second task, one selects two nodes in the graph and would like to find out how those two are related. For example, a user might be interested to know via which people or concepts Albert Einstein and Niels Bohr are related in DBpedia. In the field of bioinformatics such a link query could be applied to genes and diseases and one would be able to discover novel pathways unknown to the existing literature (Antezana et al., 2009; Bundschus et al., 2008).

In principle, the two described tasks could both be solved with shortest-path search on the ER graph, most simply by assuming a step metric, i.e. one assigns every edge in the graph a unit cost. For the first task, one would compute the shortest-path distance from the query node to all other nodes, e.g. using Dijkstra's algorithm (Dijkstra, 1959). For the second task, one would calculate the k-shortest paths connecting the two given nodes, e.g. by using the k-shortest paths algorithm of (Yen, 1971).

While shortest-path search is straightforward and efficient to implement, it often returns highly irrelevant results. Consider for example a graph which contains a node that is connected to nearly all other nodes, such as a broad category that encompasses most entities in the graph. In the first task, shortest-path search would return the high-degree node for almost any query nodes. This lack of discrimination hinders context search, since a broad topic would lack specific relevance to any one node in the graph. Also, most other graph nodes would be returned with distance two, although many of them would be unrelated to the query node. In the second task, consider a database in which every person is connected to a “Human” category node. In this case, the fact that Einstein and Bohr are both humans is less informative than the fact that they are both Nobel laureates, but the paths via “Human” and “Nobel laureates” nodes would have equal distance. These problems are schematically depicted in Figure 1.

One might consider putting higher weight on interesting nodes in ER graphs using PageRank-like approaches (Brin and Page, 1998). However, a simple implementation of this idea might have an adverse effect in our setting: in PageRank nodes are deemed popular and thus important if many links point to them. Thus high-level, highly connected nodes would become more important, although they are often uninformative from our view point as argued above. A different popularity-based ranking concept (Kasneci et al., 2008) assumes that facts that have many witnesses in a corpus are highly informative. This, however, requires additional input apart from the graph, and the assumption that important facts are expressed more often than others may not hold in curated data stores like wikipedia or company networks.

An approach more interesting in our context is based on the graph structure alone and uses properties of random walks on the graph. It has been argued that the commute time between nodes in an ER graph is a useful distance measure to find relevant neighboring nodes (Baluja et al., 2008). In each step, the random walk jumps from one node with equal probability to any neighboring node. The commute time is the expected number of steps required for a random walk starting from one node to reach another before returning to the first. Using this metric, the problems with the step distance are alleviated in that the commute distance decreases not only if there is a short path between two nodes, but also if there are several short paths between them. Thus, a single link over a high-degree hub is not likely to yield a small distance. Moreover, if two nodes are joined by a path containing a high-degree node, a random walk is likely to “get lost” at the high-degree node, taking steps into unrelated regions of the graph, increasing the commute time between the two nodes.

While these are strong intuitive arguments for the commute distance, it is very expensive to compute. There exists an analytic formula for the commute distance in terms of the pseudoinverse of the graph Laplacian (Klein and Randić, 1993). This, however, is computationally prohibitive for the large graphs encountered in the semantic web: The pseudoinverse of the sparse graph Laplacian matrix is in general not sparse such that storage requirements increase quadratically with the number of graph nodes. More efficient approximations of the commute distance have been developed for citation graphs in Sarkar et al. (2008). Their methods, however, still require 4 seconds for a graph of 600k nodes, which is only a moderate size in the context of the semantic web. Moreover, it is not clear how their methods would perform on more structurally complex graphs such as DBpedia.

Our novel approach combines the simplicity and speed of shortest path finding with the properties of the commute distance. In essence, we perform shortest path finding with a problem-adapted graph metric that assigns to each edge a weight which depends on the degrees of its endpoints. Finding shortest paths in our novel metric can then be interpreted in terms of maximizing the log-likelihood of the path between the two nodes in a random walk on the graph. It can be seen as an optimally adapted first-order approximation to the commute distance, and thus experimentally inherits many of the favorable properties of the commute distance. At the same time the computations are very efficient since they reduce to purely local shortest path searches that can be performed with standard graph algorithms.

In the next section we introduce the novel metric for solving the two tasks. In Section 3 we justify it in terms of random walks, and discuss how it can be seen as an approximation to the commute distance. In Section 4 we introduce a number of examples and a numeric evaluation on several semantic datasets, demonstrating the superior behavior both in comparison to the step-distance path-finding approach and to another simple approximation of the commute distance.

2 Proposed Approach

Let the semantic ER graph be represented as $G = (V, E)$ where V is the set of nodes or entities and E is the set of edges or relations holding between the entities. We do not distinguish between different relation types for the edges, opting to treat them all equally. Moreover, we also remove edge directions from the graph, since the semantic direction of a relation statement is often not syntactically obvious; for example, “buys” or “is bought by” might both appear in a graph.

For each edge $(u, v) \in E$, we then define a weight

$$w_{(u,v)} = \log(\deg(u)) + \log(\deg(v)),$$

where $\deg(u)$ and $\deg(v)$ are the degrees of the node u and node v, respectively. If G is connected, then the degree $\deg(u)$ of all nodes u is greater than or equal to one and thus $w_{(u,v)} \geq 0$ for all $(u, v) \in E$. The weights therefore define a valid positive semi-definite path metric on G. The two described tasks can now be solved using these novel edge weights in standard shortest paths routines.

In our first task, a node is specified as a query input and we must retrieve a set of other nodes that are ranked based on how relevant or related they are to the query node. The results of this search might include nodes that are related, for example, to topics that are contained within the query topic, to topics which contain the query node, or to topics that are related by common membership within a category or broader topic. In order to solve this task, we find the shortest paths between the query node and all other nodes (considering the weights) and rank the results. We apply Dijkstra’s algorithm, which allows us to stop short and directly retrieve the top ranked nodes without computing the shortest paths to all other nodes.

In our second task, we are given two distinct nodes as input and wish to find paths between both that, ideally, provide unique insight into the relationship between the two nodes. This might include interesting or distinct ways that the two nodes are related. We solve this by finding the k shortest paths between the two nodes in the weighted graph, where k is a free parameter. We return the sequence of nodes in each of the k-shortest paths.

The proposed metric can be justified intuitively: In the metric the distance to high-degree nodes often carrying very unspecific information, e.g. the "Human" node, is large. In contrast, the distance to more specific, low degree nodes is small. Effectively, we are searching in compactly connected, local subgraphs, assumed to carry context-specific information.

As we will see in the experimental section, the proposed approach yields matches for query nodes that are highly specific in subject matter and are very appropriate for someone who, for example, wants to explore a particular academic subject in detail. In addition, our metric facilitates the discovery of novel, distinct relations between nodes: nodes that are related to each other in some unique way (i.e. there is a path between them that is connected to relatively few other nodes outside of that path) are closer to each other than nodes that are linked by a very common relationship.

These intuitions can be further motivated by relating our approach to random walks on semantic graphs. Before we do so in the next section, note that the proposed approach only requires standard graph algorithms and is thus simple to implement. It also runs very efficiently even for large graphs. For example with Dijkstra's algorithm, we only have to visit on the order of k nodes to find the k closest neighbors.

3 The Connection to Random Walks

Consider a random walk on G. In each step the walk moves from one node v to an arbitrary adjacent node with probability $\deg(v)^{-1}$, where $\deg(v)$ is the degree of v. Denote the set of paths between two fixed nodes u and v by $\Pi_{u,v}$, i.e. $\pi = (\pi_1, \pi_2, .., \pi_{len(\pi)}) \in \Pi_{u,v}$ iff $\pi_1 = u$ and $\pi_{len(\pi)} = v$. The probability of the random walk following such a path and returning on the same route then is

$$\begin{aligned} p(\pi) &= \left(\prod_{i=1}^{n-1} \deg(\pi_i)^{-1}\right)\left(\prod_{i=2}^{n} \deg(\pi_i)^{-1}\right) \\ &= \deg(\pi_1)^{-1} \prod_{i=2}^{n-1} \deg(\pi_i)^{-2} \deg(\pi_n)^{-1}. \end{aligned}$$

The negative log-likelihood follows as

$$-\log p(\pi) = \log \deg(\pi_1) + 2\sum_{i=2}^{n-1} \log \deg(\pi_i) + \log \deg(\pi_n) = \sum_{i=1}^{n-1} w_{(\pi_i,\pi_{i+1})}.$$

This result shows that the negative log-likelihood of the path is exactly equal to the path length in our proposed metric. Moreover, shortest path finding between u and v using this metric is thus equivalent to finding that path in $\Pi_{u,v}$ that has minimal negative log-likelihood, or maximal probability, of a random walk following that path back and forth.

3.1 Approximation of the Commute Distance

Random walk probabilities also determine the commute time which has been proposed as an information metric on ER graphs before (Baluja et al., 2008). In contrast to our approach, the commute distance does not only measure whether there is a single high-probability connection between two nodes, but also takes into account how many such paths there are.

Since the commute distance uses more of the structure of the graph, it is potentially more robust. However, this comes at a huge computational cost and, in comparison, our approach is extremely efficient. Still, our approach can be seen as a first order approximation of the commute distance, as we will now discuss.

The commute time $C(u, v)$ between nodes u and v is

$$C(u,v) = \sum_{\pi:(\pi_1=u,\ldots\pi_k=v,\ldots\pi_{len(\pi)}=v)} len(\pi)\, p(\pi) = \sum_{\pi} len(\pi) \prod_{i=1}^{len(\pi)-1} \deg(\pi_i)^{-1}.$$

The sum is over all paths that start and end at u and visit v in between. Since all terms are positive, a first order lower bound can be achieved by taking into account only a single such path $\hat{\pi}$, i.e.

$$C(u,v) \geq len(\hat{\pi})\, p(\hat{\pi}).$$

Whether this is a tight bound depends on how concentrated the path probabilities are on a single term. While there are certainly situations where the path probabilities are not concentrated, we would argue that for many semantic graphs the approximation might be acceptable. The reason is that the degree of the nodes enters multiplicatively into the sum. Consider query nodes that are both members of two categories of highly different numbers of instances. Then the path through the smaller category and back on the same way is actually quadratically preferred over the one through the larger category.

Given the above lower bound we now try to find the optimal lower bound for the commute distance $C(u, v)$. That is we search for that path $\hat{\pi}$ that contributes the most to the sum above. This then leads to the criterion

$$\max_{\hat{\pi}}\; len(\pi)p(\hat{\pi}) = \min_{\hat{\pi}}\; -\log len(\pi) + \sum_{i=1}^{n-1} \log \deg(\hat{\pi}_i).$$

The second term is additive in the length of the path and quickly dominates the first term whose magnitude increases sub-linearly. At the same time, for paths of equal length only the second term has to be considered for the minimization. Without too large an error we can thus neglect the first term in most cases. Moreover, we restrict the optimization set to those paths that go from u to v and return the same way. The result will still be a lower bound on the commute distance, and it allows us to rewrite the problem using our proposed metric as

$$\min_{\hat{\pi}\in\Pi_{u,v}}\; 2\sum_{i=1}^{n-1} w_{(\hat{\pi}_i,\hat{\pi}_{i+1})}.$$

This is equivalent to our proposed approach up to a constant factor. We can thus interpret our approach as (approximately) finding an optimal lower bound to the commute distance, with the advantage that our bound can be computed very efficiently and with simple standard graph algorithms.

This derivation has involved a number of approximation steps that are not necessarily the tightest ones possible, see the review of Lovász (1993) for other approximations. Yet, this argument still gives some intuition why minimizing our proposed objective might be sensible.

4 Experiments

We demonstrate our methods using two large, real world semantic ER graphs, namely DBpedia and OpenCyc.

For the DBpedia dataset, we use the category (skos) and the article-category data files and create an unweighted, undirected graph neglecting the different relationship types and directions. We ignore literals since they do not add information to the graph structure. In the interest of a fair comparison we discard the "concept" node to which each category is connected.

We similarly define the graph for the OpenCyc dataset. An overview of the properties of both graphs is given in Table 1.

Table 1. Basic statistics of the used datasets

	DBPedia	OpenCyc
nodes	3,660,898	150,088
Edges	8,947,631	554,762
Average degree	4.88	7.39

As baseline methods for our comparisons we use the following two approaches. First, we compare our method to using shortest paths with the step distance. Second, we compute a simple approximation of the commute distance as follows. The exact computation of the commute distance on the full graph is intractable, since it requires the graph Laplacian's pseudoinverse, a matrix that for most graphs is too big to even be stored. Instead, we assume here that the commute distance is moderately local. For each query, we extract the 1000 closest nodes to the query node – in step distance – and only use the subgraph spanned by these nodes and the edges between them to compute the commute distance using the analytic formula of Klein and Randić (1993). If the subgraph has only few edges connecting it to the remaining graph, the approximation is fairly reasonable. However, if a very unspecific node with many neighbors is among the closest nodes to the query node, then it will connect almost any node in the graph to the query by a path of, say, length 2. In this case the selection of the 1000 closest neighbor nodes is arbitrary and not much can be expected from our approximation of the commute distance. The baseline should thus not be regarded as an accurate representative of the true commute distance.

4.1 First Task: Neighborhood Finding

In the following we discuss a number of example results from the two datasets. In Table 2 we list the results of a search for the query node "Espresso." In this case, the step distance gets easily distracted by the high-degree neighbor "Italian loanwords." As a result, the majority of the results listed are unrelated Italian terms which refer mostly to music and food. The commute distance approximation returns highly irrelevant words that are also related mostly to food and music. This is probably due to the nature of the approximation we use – most of the 1000 nodes nearest to the espresso node are probably also connected to the query node via the "Italian loanwords" node. Our method, on the other hand, returns a list of about one third Italian sodas and non-coffee beverages and about two thirds drinks made with espresso or at least coffee, as well as a few other types of terms.

In Table 3 we performed another search for the term "iPod." The step distance mostly gives us various categories relating to hardware or software, and the commute distance mixes these results with a few more specific terms relating to the iPod's function and to the related iPhone. Our method, on the other hand, yields mostly articles relating specifically to variations and functions of the iPod and the iTunes software, which is integral to the use of the iPod.

In Figure 2, we have a graphical representation of the results of the query "Atlanta" with our distance metric. Compare these with the results using the step

Table 2. Top 30 results of neighborhood search for query node "Espresso" in DBpedia, along with the distances from the query node. The first column labeled "Step" contains results for shortest path finding with the step distance; the second column reports results using our proposed approach; the last column shows the results of our simple approximation of the commute distance. Entities marked with (C) represent skos categories, other items are regular DBpedia resources.

Step		Our approach		Approx. Commute	
Espresso	0	Espresso	0	Espresso	0
(C)Italian beverages	1	(C)Italian beverages	4.5	(C)Italian loanwords	1295.75
(C)Italian loanwords	1	(C)Coffee beverages	5.05	(C)Coffee beverages	1296.86
(C)Coffee beverages	1	(C)Italian loanwords	6.09	(C)Italian beverages	1297.35
(C)Italian cuisine	2	Bombardino	7.9	(C)Italian cuisine	1339.5
(C)Italian words and phrases	2	Caffè corretto	8.59	(C)Opera terminology	1401.79
(C)Italian language	2	Grappa	8.59	(C)Italian words and phrases	1452.94
(C)English words foreign origin	2	Torani	8.59	(C)Pasta	1467.75
(C)Romance loanwords	2	Lemonsoda	8.59	(C)Mediterranean cuisine	1529.31
(C)Beverages by region	2	Oransoda	8.59	(C)Cuisine by nationality	1544.99
(C)Italian alcoholic beverages	2	Pelmosoda	8.59	(C)Opera genres	1582.18
(C)Coffee preparation	2	Beverly (drink)	8.59	(C)Opera	1584.02
Castrato	2	Doppio	8.59	(C)Performing arts	1599.79
Da capo	2	Caffè	9	(C)Musical notation	1601.59
Graffiti	2	Chinotto	9	(C)European cuisine	1664.97
Glissando	2	Ammazzacaffè	9	(C)Italian language	1685.92
Macaroni	2	Stappj	9	Turkish coffee	1691.75
Mozzarella	2	Galvanina	9	(C)Beverages by region	1721.4
Opera	2	Irish coffee	9	(C)Dried meat	1737.1
Pasta	2	Cortado	9	(C)Musical theatre	1740.42
Pizza	2	Iced coffee	9	(C)Music	1743.96
Spaghetti	2	Pepsi Kona	9	(C)Articulations	1756.06
Tempo	2	Flat white	9	(C)English words foreign origin	1756.7
Cappuccino	2	Mochasippi	9	(C)Singing	1760.76
Legato	2	Red eye (drink)	9	(C)Salumi	1764.59
Staccato	2	Liqueur coffee	9	(C)Croatian cuisine	1769.7
Operetta	2	Lungo	9	(C)Entertainment	1773.37
Cadenza	2	Caffè Americano	9	(C)Theatrical genres	1788.05
Concerto	2	Espresso con panna	9	(C)Italian culture	1795.82
Cantata	2	Caffè breve	9	(C)Italian prod. protected origin	1799.13

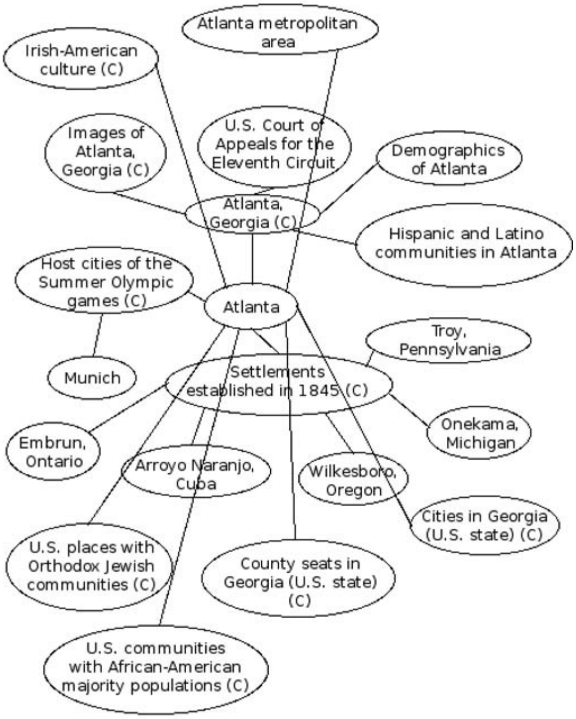

Atlanta metropolitan area
Irish-American culture (C)
Images of Atlanta, Georgia (C)
U.S. Court of Appeals for the Eleventh Circuit
Demographics of Atlanta
Atlanta, Georgia (C)
Hispanic and Latino communities in Atlanta
Host cities of the Summer Olympic games (C)
Atlanta
Troy, Pennsylvania
Munich
Settlements established in 1845 (C)
Onekama, Michigan
Embrun, Ontario
Arroyo Naranjo, Cuba
Wilkesboro, Oregon
Cities in Georgia (U.S. state) (C)
U.S. places with Orthodox Jewish communities (C)
County seats in Georgia (U.S. state) (C)
U.S. communities with African-American majority populations (C)

Fig. 2. Graphical representation of the results of the query "Atlanta" in our new distance. In the middle of the figure is the query node, Atlanta. The remaining nodes are the search results. This presentation of the results shows the connection of each of the search results to the query node – that is, the shortest path found from the query to each of the results is preserved in this graph representation.

distance, graphically represented in Figure 3. While the step distance returns mostly generic facts such as the fact that Atlanta is a U.S. city, our distance returns more interesting results including the connection to the U.S. Court of Appeals for the Eleventh Circuit and the connection to Munich – they are both Olympic cities.

We also provide results for the OpenCyc dataset, which is of a slightly different nature. It contains many rather unspecific nodes like "temporally stuff like thing" which are nice examples of how such high-degree nodes are avoided by our algorithm. In Table 4 we show the results of a search for "Machine learning." While the results of the other methods become wildly irrelevant after only the first few matches, nearly the first half of the results of our approach are still relevant to the topic at hand.

To demonstrate the dramatic computational advantage of our method against the described approximation of the commute distance, we picked 1000 query

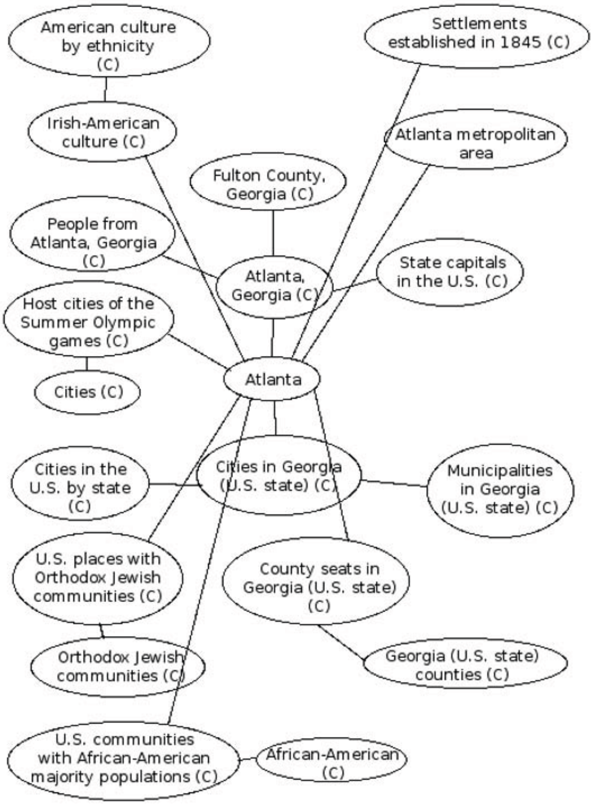

American culture by ethnicity (C)
Settlements established in 1845 (C)
Irish-American culture (C)
Atlanta metropolitan area
Fulton County, Georgia (C)
People from Atlanta, Georgia (C)
Atlanta, Georgia (C)
State capitals in the U.S. (C)
Host cities of the Summer Olympic games (C)
Atlanta
Cities (C)
Cities in the U.S. by state (C)
Cities in Georgia (U.S. state) (C)
Municipalities in Georgia (U.S. state) (C)
U.S. places with Orthodox Jewish communities (C)
County seats in Georgia (U.S. state) (C)
Orthodox Jewish communities (C)
Georgia (U.S. state) counties (C)
U.S. communities with African-American majority populations (C)
African-American (C)

Fig. 3. Graphical representation of the results of the query "Atlanta" in the step distance. As in figure 2, the shortest paths from the query node to each of the results is presented.

nodes at random and performed a query using all three methods. The mean run-times on a standard desktop PC as well as the standard deviation for each method is given below

Step	Our Approach	Approx. Commute
0.13s (0.07s)	0.11s (0.04s)	10.43s (9.51s)

The average run-time for our method was 0.11 seconds, compared to an average of 10.43 seconds for our approximation of the commute time – a difference of two orders of magnitude. As would be expected, our method runs approximately as fast as the step distance method.

It is worth noting that the time taken to approximate the commute time was extremely high in some cases. The longest time taken with the commute distance was over one minute, whereas the longest time taken with our method was only 0.66 seconds. Furthermore, one should also consider that the method we have used to calculate the commute distance is only an approximation using

Table 3. Top 30 results for calculating neighborhood search for query node “iPod” in DBpedia. Labels as in Table 2.

Step		Our approach		Approx. Commute	
IPod	0	IPod	0	IPod	0
(C)2001 introductions	1	(C)IPod	4.97	(C)ITunes	695.93
(C)IPod	1	(C)Industrial designs	5.78	(C)Portable media players	698.52
(C)Portable media players	1	(C)ITunes	5.98	(C)Digital audio players	750.23
(C)ITunes	1	(C)2001 introductions	6.29	(C)IPhone OS software	757.78
(C)IPhone OS software	1	(C)Portable media players	6.49	(C)IPod	784.31
(C)Industrial designs	1	(C)IPhone OS software	6.52	(C)Industrial designs	857.01
(C)2001	2	IPod click wheel	8.15	(C)Smartphones	889.69
(C)Apple Inc. software	2	IPod Photo	8.84	(C)2001 introductions	907.63
(C)Industrial design	2	List of iPod models	8.84	(C)Mac OS X software	929.97
(C)Windows software	2	Dock Connector	8.84	(C)Touchscreen portable media players	955.66
(C)Software by operating system	2	IPod Mini	9.25	(C)Consumer electronics brands	959.29
(C)Apple Inc. hardware	2	IPod advertising	9.25	(C)Apple Inc. software	973.22
(C)Windows media players	2	IPhone Touch	9.25	(C)IPhone	974.07
(C)Mac OS X software	2	IPod Nano	9.53	(C)2007 introductions	1010.71
(C)Digital audio players	2	Neistat Brothers	9.53	IPhone	1025.22
(C)USA PATRIOT Act	2	IPod Classic	9.53	(C)IPhone OS	1031.79
(C)MPEG	2	Ipod+HP	9.53	(C)Web 2.0	1035.86
(C)IPod accessories	2	List of iPhone OS devices	9.53	(C)Windows software	1047.63
(C)IPod software	2	IPod Shuffle	9.76	ITunes	1049.63
(C)21st-century introductions	2	Juicy Salif	9.77	(C)Apple Inc. hardware	1057.46
(C)ITunes-exclusive releases	2	DADVSI	10.09	(C)Software by operating system	1075.57
(C)IPhone OS games	2	NextWorth Solutions	10.09	(C)Online social networking	1096.2
(C)Mac OS X media players	2	IMix	10.17	(C)Mac OS software	1096.22
(C)Apple Inc. peripherals	2	Genius (iTunes)	10.17	(C)Personal digital assistants	1112.79
(C)Apple Inc. services	2	AirTunes	10.17	(C)Brands	1126.36
(C)Vehicles introduced in 2001	2	ITunes law	10.17	(C)Media players	1126.94
(C)IPhone	2	ITunes Music Store	10.17	(C)Creative Technology products	1129.28
(C)2001 comic debuts	2	ITunes U	10.17	(C)IPod software	1151.43
(C)IPhone OS	2	ITunes Applications	10.17	Nimbuzz	1156.45

Table 4. Top 30 results of neighborhood search for query node “Machine learning” in OpenCyc. Labels as in Table 2.

Step		Our approach		Approx. Commute	
machine learning	0	machine learning	0	machine learning	0
temporal stuff also a durative event	1	machine rule induction	2.48	first-order collection	875.61
computer activity	1	discriminative weight learning	2.89	temp stuff also a durative event	887.91
discriminative weight learning	1	generative weight learning	2.89	computer activity	897.63
generative weight learning	1	MLN Generated Using Learning Type	3.18	temporal stuff	921.03
machine rule induction	1	computer activity	6.27	employee computer activity type	1061.05
MLN Generated Using Learning Type	1	markov logic network	6.87	computer activity type	1090.34
alcoholism	2	temporal stuff also a durative event	7.75	athletic activity	1104.59
burning	2	MLN Data File Pathname	9.86	physical information transfer	1115.24
flowing	2	MLN File Pathname	9.86	biological transportation	1138.37
anthem	2	MLN Generated Using Cmd String	9.86	body movement	1152.19
the union of ensemble showman	2	MLN Rule File Pathname	9.86	recreational activity	1169.12
playing	2	MLN Type Const Dec File Pathname	9.86	using a computer	1181.74
halt	2	MLN Represented By Microtheory	10.27	information-accessing event	1195.75
rock climbing	2	Content Of MLN Fn	10.56	physical event	1196.47
snow-skiing	2	computer activity that computer did	11.85	structured information source	1213.39
Iter. Event Scene Fn id veg. 1-3 km	2	computer activity that person did	11.85	type of accomplishment	1236.74
rafting	2	hack	11.85	individual	1239.97
candy making	2	computer thread	11.85	computer editing	1256.46
composting	2	help desk session	11.85	internet activity	1266.56
woodworking	2	network packet filtering	11.85	running computer process	1280.32
diagnosis of Wegeners granulomatosis	2	network packet routing	11.85	locomotion event	1280.92
breast cancer treatment	2	opening presents	11.85	ride	1303.08
AIDS treatment	2	packet sniffing	11.85	CW instantiating	1313.32
acne care	2	partitioning a disk	11.85	unnatural thing	1315.36
affliction procedure	2	placing a residual malicious program	11.85	biological process	1321.43
allergic reaction treatment	2	browser requests a secure connection	12.13	QA clarifying collection type	1338.81
atrial septal aneurysm med treatment	2	locking computer display	12.13	internet communication	1355.33
most autistic procedure	2	website maintenance	12.13	network propagation	1357.71
vision impairment treatment	2	network prop. malicious program	12.13	candidate KB completeness node	1360.95

a graph of 1000 nodes. The most computationally intensive step required of the commute distance is the calculation of the pseudoinverse. Since this step requires cubic time to calculate, an attempt to improve the accuracy of the estimate by adding more nodes to the approximation would drastically increase the time required for computation, while an exact computation would be intractable for most practical problems.

4.2 Task 2: Path Finding

In this section, we present an example of our method as applied to the path finding task, displaying the paths that our method is able to find between two nodes in the graph – i.e., between two concepts in our semantic network. We also compare our method to path finding using the step distance to show the advantage that our method has in discovering truly distinct and specific connections between concepts.

Consider paths between the nodes “Computer vision” and “Machine learning”, again with data from DBpedia. The resulting paths are listed in Table 5. Many of the results of our method provide insight into exactly how machine learning is used to solve specific tasks in the computer vision domain. Although insightful, some of the paths returned here by our method have significant intersections with each other. This could, however, be remedied by, for example, modifying the k-shortest paths algorithm to add extra weight to the edges equivalent to the ones traversed in previously discovered paths. Such a modification would lead to increased diversity in the results.

The step distance, on the other hand, gives us only very vague, general connections between the two subjects. The most that we learn from these results is that computer vision and machine learning are both within the subject of artificial intelligence.

Table 5. Path finding between the terms “Computer vision” and “Machine learning” in DBpedia

Our Approach:

– Path 1 (length 15.2407): Computer vision - (C)Computer vision - (C)Learning in computer vision - Machine learning
– Path 2 (length 22.1722): Computer vision - (C)Computer vision - (C)Object recognition and categorization - Boosting methods for object categorization - (C)Learning in computer vision - Machine learning
– Path 3 (length 22.4706): Computer vision - (C)Artificial intelligence - (C)Cybernetics - Machine learning
– Path 4 (length 23.5585): Computer vision - (C)Computer vision - Segmentation based object categorization - (C)Object recognition and categorization - Boosting methods for object categorization - (C)Learning in computer vision - Machine learning
– Path 5 (length 23.5585): Computer vision - (C)Computer vision - Object recognition (computer vision) - (C)Object recognition and categorization - Boosting methods for object categorization - (C)Learning in computer vision - Machine learning

Step Distance:

– Path 1 (length 3): Computer vision - (C)Artificial intelligence - (C)Machine learning - Machine learning
– Path 2 (length 3): Computer vision - (C)Computer vision - (C)Learning in computer vision - Machine learning
– Path 3 (length 3): Computer vision - (C)Artificial intelligence - (C)Cybernetics - Machine learning
– Path 4 (length 4): Computer vision - (C)Artificial intelligence - (C)Machine learning - (C)Learning - Machine learning
– Path 5 (length 4): Computer vision - (C)Computer vision - (C)Artificial intelligence - (C)Machine learning - Machine learning

Table 6. Path finding between the terms “Seattle” and “Quantum mechanics” in DBpedia

Our Approach:

- Path 1 (length 42.957): Seattle - (C)Seattle, WA - Homelessness in Seattle - (C)Articles Created via the Article Wizard - Magnetic translation - (C)Quantum Mechanics - Quantum Mechanics
- Path 2 (length 44.3577): Seattle - (C)Isthmuses - (C)Coastal and oceanic landforms - Sound (geography) - (C)Sound - Amplitude - (C)Fundamental physics concepts - Quantum mechanics
- Path 3 (length 44.933): Seattle - (C)Isthmuses - (C)Coastal and oceanic landforms - Sound (geography) - (C)Sound - Node (physics) - (C)Fundamental physics concepts - Quantum mechanics
- Path 4 (length 45.744): Seattle - (C)Isthmuses - Isthmus - (C)Coastal and oceanic landforms - Sound (geography) - (C)Sound - Amplitude - (C)Fundamental physics concepts - Quantum mechanics
- Path 5 (length 46.1677): Seattle - Seattle, Washington - (C)Education in Seattle, Washington - Washington Large Area Time Coincidence Array - (C)Cosmic-ray experiments - (C)Experimental particle physics - Cherenkov radiation - (C)Fundamental physics concepts - Quantum mechanics

Step Distance:

- Path 1 (length 6): Seattle - (C)Seattle, Washington - Homelessness in Seattle - (C)Articles created via the Article Wizard - Magnetic translation - (C)Quantum mechanics - Quantum mechanics
- Path 2 (length 7): Seattle - (C)Isthmuses - (C)Coastal and oceanic landforms - Archipelago - (C)Greek loanwords - Atom - (C)Fundamental physics concepts - Quantum mechanics
- Path 3 (length 7): Seattle - (C)Seattle, Washington - (C)People from Seattle, Washington - Virgil Bogue - (C)1916 deaths - Kārlis Mīlenbah - (C)Quantum mechanics - Quantum mechanics
- Path 4 (length 7): Seattle - (C)Seattle, Washington - Homelessness in Seattle - (C)Articles created via the Article Wizard - Photomechanical effect - (C)Mechanics - (C)Quantum mechanics - Quantum mechanics
- Path 5 (length 7): Seattle - (C)Seattle, Washington - Homelessness in Seattle - (C)Articles created via the Article Wizard - Magnetic translation - (C)Quantum magnetism - (C)Quantum mechanics - Quantum mechanics

Note that our method is actually able to find informative paths of significant length. While for the step distance the exponential number of possibilities for such paths quickly renders the retrieval infeasible, our method is still able to discriminate between the many choices. This might be an important advantage when applying this framework to biomedical databases, such as for example Linked Life Data. Here, one often tries to find non-obvious rather long distance interactions between different genes and diseases to discover novel pathways. Focusing on the most discriminative ones might save significant research effort in this domain.

In Table 6, we present the results of a shortest paths query of five steps between Seattle and Quantum Physics in our metric and the step metric. In addition to the connection through the interesting semantic ambiguity of the word “Sound,” our method yields what could be argued to be the most valuable link between the two concepts: the involvement in the Washington Large Area Time Coincidence Array, a distributed physics experiment. The paths in the step distance are mostly dominated by the tenuous link of articles created with the Article Wizard. It is also interesting (although not vital to the interpretation of the results) to note that although Kārlis Mīlenbah is indeed listed in the category Quantum Physics, at the time of publication, the article for Kārlis Mīlenbah seems to imply that he probably has nothing to do with quantum physics.

5 Conclusion

We have presented a novel metric for solving information retrieval tasks in semantic networks. The metric just depends on the degrees of adjacent nodes and favors paths via low-degree nodes. As our experiments demonstrated, the approach is capable of finding atypical but interesting neighbors of a query node. In addition, the approach is able to find original informative paths between two specified nodes.

Often the authors themselves discovered novel, interesting information when querying the test datasets DBpedia and OpenCyc with different entities. This makes us strongly believe that the proposed approach could also be helpful to others.

A detailed user study is currently under way. From a technical point one could imagine mixing the step metric and the proposed one to obtain a tunable trade-off between the length and the distinctiveness of a path. It would also be interesting to explore ways to learn additional parameters in the metric, e.g. by assigning different weights to specific edge types. Such parametric learning approaches, however, would require a benchmarking dataset which is currently not available to us. In contrast, the proposed approach is parameter free and solely dependent on intuitive arguments.

References

Antezana, E., Kuiper, M., Mironov, V.: Biological knowledge management: the emerging role of the Semantic Web technologies. Briefings in Bioinformatics (2009)

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: Dbpedia: A Nucleus for a Web of Open Data. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007)

Baluja, S., Seth, R., Sivakumar, D., Jing, Y., Yagnik, J., Kumar, S., Ravichandran, D., Aly, M.: Video suggestion and discovery for YouTube: taking random walks through the view graph. In: Proceeding of the 17th International Conference on World Wide Web, pp. 895–904. ACM (2008)

Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems 30(1-7), 107–117 (1998)

Bundschus, M., Dejori, M., Stetter, M., Tresp, V., Kriegel, H.-P.: Extraction of semantic biomedical relations from text using conditional random fields. BMC Bioinformatics 9(1), 207 (2008)

Dijkstra, E.: A note on two problems in connexion with graphs. Numerische Mathematik 1(1), 269–271 (1959)

Kasneci, G., Suchanek, F., Ifrim, G., Ramanath, M., Weikum, G.: Naga: Searching and ranking knowledge. In: Proc. of ICDE, pp. 1285–1288 (2008)

Klein, D.J., Randić, M.: Resistance distance. Journal of Mathematical Chemistry 12(1), 81–95 (1993)

Lovász, L.: Random walks on graphs: A survey. Combinatorics, Paul Erdos is Eighty 2(1), 1–46 (1993)

Momtchev, V., Peychev, D., Primov, T., Georgiev, G.: Expanding the pathway and interaction knowledge in linked life data. In: Proc. of International Semantic Web Challenge (2009)

Sarkar, P., Moore, A., Prakash, A.: Fast incremental proximity search in large graphs. In: Proceedings of the 25th International Conference on Machine Learning, pp. 896–903. ACM (2008)

Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A Core of Semantic Knowledge. In: 16th International World Wide Web Conference (WWW 2007). ACM Press, New York (2007)

Yen, J.: Finding the k-shortest loopless paths in a network. Management Science 17(11), 712–716 (1971)

Towards BOTTARI: Using Stream Reasoning to Make Sense of Location-Based Micro-posts

Irene Celino[1], Daniele Dell'Aglio[1], Emanuele Della Valle[2,1], Yi Huang[3],
Tony Lee[4], Seon-Ho Kim[4], and Volker Tresp[3]

[1] CEFRIEL – ICT Institute, Politecnico of Milano, Milano, Italy
[2] Dip. di Elettronica e dell'Informazione – Politecnico di Milano, Milano, Italy
[3] SIEMENS AG, Corporate Technology, Muenchen, Germany
[4] Saltlux, Seoul, Korea

Abstract. Consider an urban environment and its semi-public realms (e.g., shops, bars, visitors attractions, means of transportation). Who is the maven of a district? How fast and how broad can such maven influence the opinions of others? These are just few of the questions BOTTARI (our Location-based Social Media Analysis mobile app) is getting ready to answer. In this position paper, we recap our investigation on deductive and inductive stream reasoning for social media analysis, and we show how the results of this research form the underpinning of BOTTARI.

1 Introduction

In the last few years, we have been witnessing the increasing popularity and success of Location-based Services (LBS), especially of those with a Social Networking flavour. Twitter, Facebook Places, foursquare, Gowalla are only a few application examples; those services bring a wide range on useful information about tourist attractions, local businesses and points of interests (POIs) in the physical world. Although these services are enormously popular, users still suffer from a number of shortcomings. The overwhelming information flow coming from those channels often confuses users; it is also very difficult to distinguish between a fair personal opinion and a malicious or opportunistic advice. This might be the reason why users primarily link to people they know personally since, in an on-line social network, there is no clear way to know who to trust.

In this paper, we present our collaborative effort to the design and development of the BOTTARI application, a Location-based Service for mobile users that exploit Social Media Analysis techniques to identify the "mavens" of a specific geographical area, i.e. those people who can be considered as experts of the POIs in this area. BOTTARI was conceived by Saltlux, a Korean Knowledge Communication Company. The application is still under development and it will be made available to Korean users in the Seoul area. BOTTARI exploits hybrid Stream Reasoning both on heterogeneous social network data [1] and geolocation data. The hybrid reasoning engine combines deductive and inductive

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 80–87, 2012.

techniques. Since the input data are huge and change in real-time, the reasoning engine works by processing streaming data. The hybrid reasoning engine is developed on top of the LarKC platform [2], a pluggable architecture to build applications with Semantic Web technologies.

The remainder of the paper is organised as follows. Section 2 explains the concept of stream reasoning and delineates the system architecture. Section 3 describes the BOTTARI app. Section 4 details some user questions in terms of queries to our stream reasoner. Finally, Section 5 concludes the paper.

2 System Architecture

Continuous processing of information flows (i.e. **data streams**) has widely been investigated in the database community. [3]. In contrast, continuous processing of data streams *together with rich background knowledge* requires semantic reasoners, but, so far, semantic technologies are still focusing on rather static data. We strongly believe that there is a need to close this gap between existing solutions for belief update and the actual need of supporting decision making based on data streams and rich background knowledge. We named this little-explored, yet high-impact research area **Stream Reasoning** [4]. The foundation for Stream Reasoning has been investigated by introducing technologies for wrapping and querying streams in the RDF data format (e.g., using C-SPARQL [5]) and by supporting simple forms of reasoning [6] or query rewriting [7].

We are developing the Stream Reasoning vision on top of LarKC [8]. The LarKC platform is aimed to reason on massive heterogeneous information such as social media data. The platform consists of a framework to build workflows, i.e. sequences of connected components (plug-ins) able to consume and process data. Each plug-in exploits techniques and heuristics from diverse areas such as databases, machine learning and the Semantic Web.

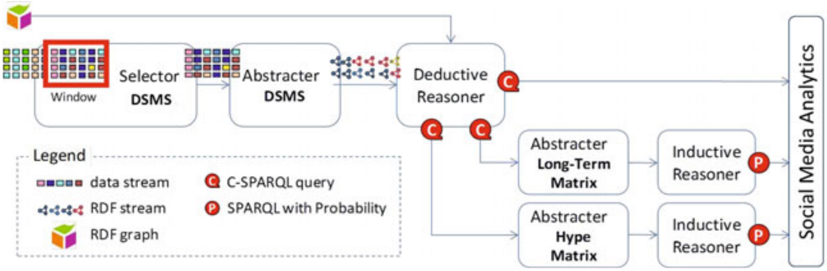
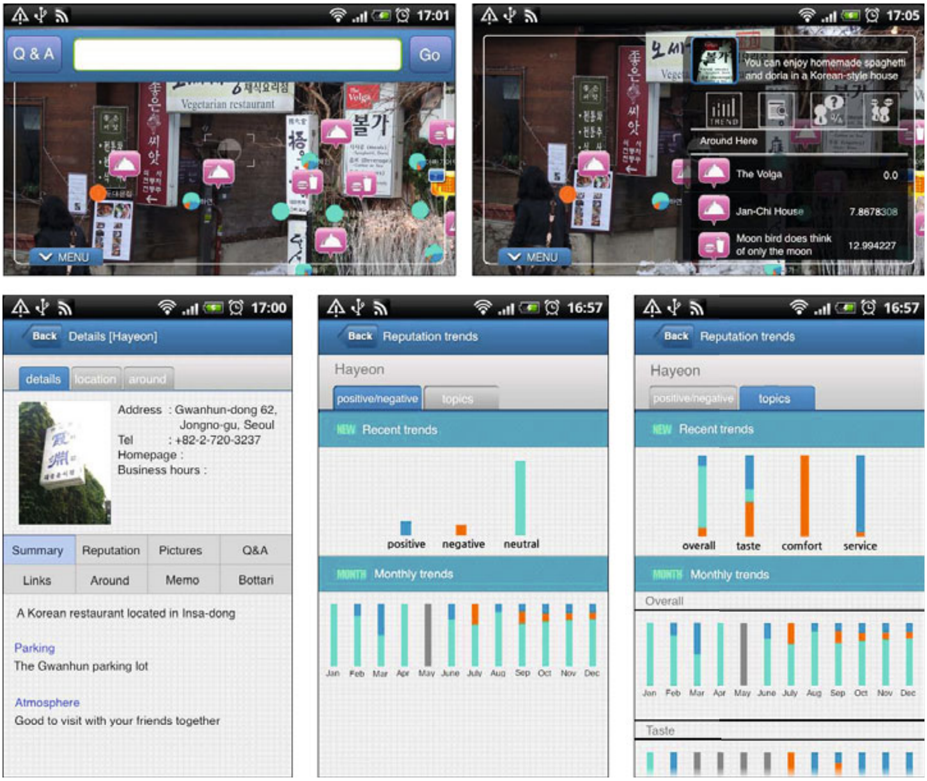


Fig. 1. Architecture of our Stream Reasoner

We built our Stream Reasoning system by embedding a deductive reasoner and an inductive reasoner within the LarKC architecture (see Figure 1). First, BOTTARI pre-processes the micro-posts by extracting information[1] whether a micro-post expresses a positive or a negative feeling of its author about a certain POI.

[1] Those technological details are Saltlux trade secrets.

After BOTTARI data arrives to the stream reasoner as a set of data streams, a selection plug-in extracts the relevant data in each input window of the stream. A second plug-in abstracts the window content from the fine grain data streams into aggregated events and produces RDF streams. Then, a deductive reasoner plug-in is able to register C-SPARQL queries, whose results can be of immediate use (cf. Section 4) or can be processed by other two sub-workflows. Each sub-workflow is constituted by an abstracter and an inductive reasoner, which uses an extended version of SPARQL that supports probabilities [9].



You can enjoy homemade spaghetti and doria in a Korean-style house

The Volga	0.0
Jan-Chi House	7.8678308
Moon bird does think of only the moon	12.994227

Address : Gwanhun-dong 62, Jongno-gu, Seoul
Tel : +82-2-720-3237
Homepage :
Business hours :

A Korean restaurant located in Insa-dong

Parking
The Gwanhun parking lot

Atmosphere
Good to visit with your friends together

Fig. 2. Some screenshots of the BOTTARI Android application

3 The BOTTARI Mobile App

Bottari is a Korean word that refers to a bundle or container made from patterned cloth that is used to transport a one's belongings when travelling. The BOTTARI mobile app is a location-based service that exploits the social context to provide relevant contents to the user in a specific geographic location; as such, BOTTARI lets the user "transport" the location-specific knowledge, derived from the local mavens' expertise, when moving in the physical space.

The purpose of the BOTTARI service is to provide recommendations on local context information to users through an augmented reality interface. BOTTARI gives detailed information on local POIs, including trust or reputation information. In Figure 2, we provide some sample screenshots on how the BOTTARI mobile application will look like once completed. The screenshots in the upper part of Figure 2 show how a user searches for POIs of a given kind (e.g., restaurants or snack bars) around her position and explores them using augmented reality. A small pie graph shows the results of the sentiment analysis for each POI: blue for positive, red for negative, and green for neutral feeling. The screenshots in the bottom part of Figure 2 show how a user visualizes more detailed information about a POI. They are, from left to right, the POI identity card, the global sentiment analysis (again, blue, red and green represent positive, negative and neutral feeling respectively) and the detailed sentiment analysis on different topics (e.g., taste, comfort and service for a restaurant).

The input data for the BOTTARI service come from public social networks and location based services (Twitter, local blogs and Korean news). They are converted into RDF streams and then processed and analysed by the system described in Section 2. The RDF-ized data are modelled with respect to the ontology represented in Figure 3, which is an extension to the SIOC vocabulary [10]. Our model takes into account the specific relations of Twitter (follower/following, reply/retweet); it adds the geographical perspective by modelling the POIs; it includes the "reputation" information by means of positive/negative/neutral reviews.

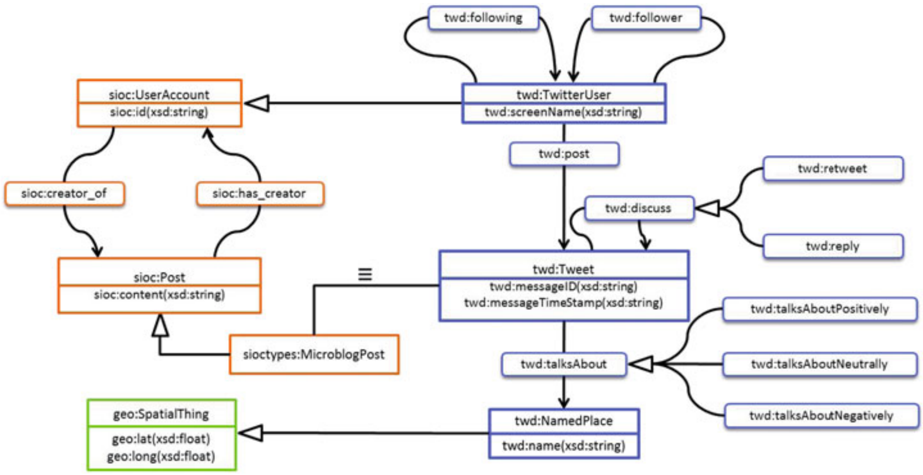


Fig. 3. Ontology modelling of BOTTARI data

4 Computing Answers to User Questions

The hybrid Stream Reasoning solutions we are developing is able to answer questions like: Who are the opinion makers (i.e., the users who are likely to influence the behaviour of their followers with regard to a certain POI)? How fast and how wide are opinions spreading? Who shall I follow to be informed about a given category of POIs in this neighbourhood? Which persons similar to me are nearby at an interesting POI?

In the rest of the section we show how to formulate the four queries above using C-SPARQL and SPARQL with probabilities.

Who are the Opinion Makers?

Lines 1 and 4 of the following listing tell the C-SPARQL engine to register the continuous query on the stream of micro-posts generated by BOTTARI considering a sliding window of 30 minutes that slides every 5 minutes. Lines 2 and 3 tells the engine that it should generate an RDF stream as output reporting the opinion makers for a certain POI.

```
1. REGISTER STREAM OpinionMakers COMPUTED EVERY 5m AS
2. CONSTRUCT { ?opinionMaker a twd:opinionMaker ;
3.                    twd:posts [ twd:talksPositivelyAbout ?poi ] . }
4. FROM STREAM <http://bottari.saltlux.com/posts>  [RANGE 30m STEP 5m]
5. WHERE {
6.        ?opinionMaker a twd:TwitterUser ;
7.                      twd:posts [ twd:talksPositivelyAbout ?poi ] .
8.        ?follower sioc:follows ?opinionMaker;
9.                  twd:posts [ twd:talksPositivelyAbout ?poi ] .
10.      FILTER (cs:timestamp(?follower) > cs:timestamp(?opinionMaker))
11. }
12. HAVING ( COUNT(DISTINCT ?follower) > 10 )
```

The basic graph pattern (BGP) at lines 6–7 matches positive micro-posts of the potential opinion makers about a set of POIs. The BGP at lines 8–9 looks up the followers of the opinion makers who also positively posted about the same set of POIs. The FILTER clause at line 10 checks whether the micro-posts of the followers occur after those from the opinion makers. Finally, at line 12 the clause HAVING promotes to *true* opinion makers those who have at least ten such followers.

How Fast and Wide Opinions are Getting Spread?

Using the RDF stream computed by the previous query, the query in the following listing informs about how wide the micro-posts of an opinion maker are getting spread in half an hour. To do so, it considers the reply and re-tweet relationships among tweets (i.e., tweets linked by the `discuss` property in BOTTARI data model). Being `discuss` a transitive property, the C-SPARQL engine uses the materialization technique presented in [6] to incrementally compute the transitive closure of `discuss`.

```
1.  REGISTER STREAM OpinionSpreading COMPUTED EVERY 30s AS
2.  SELECT ?opinionMaker ?opinionMakerTweet
3.         COUNT(?positiveTweet) COUNT(?negativeTweet)
4.  FROM STREAM <http://bottari.saltlux.com/posts> [RANGE 30m STEP 30s]
5.  FROM STREAM <http://bottari.../OpinionMakers [RANGE 30m STEP 30s]
6.  WHERE {
7.        ?opinionMaker a twd:opinionMaker ;
8.                      twd:post ?opinionMakerTweet .
9.        { ?positiveTweet a twd:Tweet ;
10.                        twd:discuss ?opinionMakerTweet ;
11.                        twd:talksAboutPositively ?poi .  }
12.       UNION
13.       { ?negativeTweet a twd:Tweet ;
14.                        twd:discuss ?opinionMakerTweet ;
15.                        twd:talksAboutNegatively ?poi .  }
```

Lines 1, 4 and 5 tell the C-SPARQL engine to register the continuous query on the stream of micro-posts generated by BOTTARI and on the streaming results of the opinion makers query. In both cases, a sliding window of 30 minutes, which slides every 30 seconds, is considered. The BGP at lines 7–8 matches the micro-posts of the opinion makers. The group pattern at lines 9–11 and the group pattern at lines 13–15 look up other micro-posts that, respectively, positively and negatively discussed those of the opinion makers. Lines 2–3 ask the engine to generate a variable binding reporting how many positive and negative micro-posts are discussing the micro-posts of the current opinion makers.

Who shall I Follow?

Let us consider now a specific BOTTARI user named Giulia. In the following listing we show a query that asks for the mavens Giulia should follow to be informed about attractions for kids, even among people she does not know. The system uses the social network of Giulia and the last window in the stream (generated by the query in the first listing) to determine such predicted probability.

```
1.  SELECT ?user ?prob
2.  FROM STREAM <http://bottari.../OpinionMakers [RANGE 30m STEP 30s]
3.  WHERE{
4.      ?opinionMaker a twd:opinionMaker ;
5.                    twd:posts [ twd:talksAboutPositively ?poi ] .
6.      ?poi skos:subject twd:attractionsForKids .
7.      { :Giulia twd:following ?opinionMaker.
8.        WITH PROBABILITY AS ?prob
9.        ENSURE PROBABILITY [0.8,1] }
10. } ORDER BY DESC(?prob)
```

The BGP at lines 4–6 matches those opinion makers that have recently been expressing positive opinions about attractions for kids. The group pattern at lines 7–9 makes use of SPARQL with probability [9]. The triple pattern at line

7 matches BOTTARI users that Giulia is following. Note that the `following` relationship may have not been asserted yet, the WITH PROBABILITY clause at line 8 extends SPARQL by letting it query an induced model. The variable `?prob` may assume values between 0 and 1, where the value 1 means that she already follows that user. The ENSURE PROBABILITY clause at line 9 accepts only those solutions whose estimated probabilities is larger or equal to 0.8 and less than 1, i.e. those mavens who should be recommended to Giulia. Finally, the ORDER BY clause is used to return users sorted by decreasing probability. The query answer includes pairs of users and predicted likelihoods (e.g. `:Alice` with probability 0.99, `:Bob` with probability 0.87).

What People Similar to Me Are Nearby in an Interesting POI? A more complex example of query soliciting all BOTTARI features is as follows. Let's consider that Giulia is now in a specific location and she is looking for people who share her preferences and who are nearby in an interesting POI. Both physical proximity and recency of micro-posting are to be considered.

```
1.  PREFIX ogc: <http://www.opengis.net/geosparql#>
2.  PREFIX ogcf: <http://www.opengis.net/geosparql/functions#>
3.  SELECT ?poi1 ?user ?prob
4.  FROM STREAM <http://bottari.../streamOftweets> [RANGE 1h STEP 10m]
5.  WHERE {
6.    ?user twd:post [ twd:talksPositivelyAbout ?poi1 ] .
7.    ?poi1 geo:lat ?lat1; geo:long ?long1 ; skos:subject ?category .
8.    :Giulia twd:post [ twd:talksAbout ?poi2 ] .
9.    ?poi2 geo:lat ?lat2; geo:long ?long2 ; skos:subject ?category .
10.    FILTER( ogcf:distance(ogc:Point(?lat1,?long1),
11.                          ogc:Point(?lat2,?long2), ogc:km) < 0.1 )
12.   { :Giulia twd:following ?user.
13.     WITH PROBABILITY AS ?prob
14.     ENSURE PROBABILITY [0.8,1) }
15. }
16. ORDER BY DESC(?prob)
```

Line 4 indicates that activities should be related to the latest period; lines 8–9 determine where Giulia is, while lines 6–7 ask for users who recently tweeted positively about a POI of the same category of the one where Giulia is. Lines 10–11 make sure that the POI is nearby by using GeoSPARQL, a proposal by the Open Geospatial Consortium[2]. The group pattern at lines 12–14 leverages inductive reasoning to ensure at least a 80% similarity between Giulia and the nearby tweeters; this probabilistic value is used to rank results (cf. line 16).

5 Conclusions and Future Works

In this paper we presented BOTTARI, a location-based mobile application which is able to supply contents and personalized suggestions to its users. We explained

[2] Cf. http://www.opengeospatial.org/projects/groups/geosparqlswg

the processing of new recommendations, based on the elaboration of data streams generated by microblogging platforms like Twitter and foursquare. The computation is defined as a workflow combining Semantic Web and machine learning techniques and it is executed on top of the LarKC platform.

Our future work will focus on the development of the first stable version of the BOTTARI application and its release as Android app. The initial release will focus on Korea and will be evaluated by following a user-centered approach: a set of users will try out the application, supplying us feedbacks via a survey with questions about the system and its accuracy in providing suggestions.

Acknowledgments. This work was partially supported by the EU project LarKC (FP7-215535).

References

1. Barbieri, D.F., Braga, D., Ceri, S., Della Valle, E., Huang, Y., Tresp, V., Rettinger, A., Wermser, H.: Deductive and Inductive Stream Reasoning for Semantic Social Media Analytics. IEEE Intelligent Systems 25(6), 32–41 (2010)
2. Cheptsov, A., et al.: Large Knowledge Collider. A Service-oriented Platform for Large-scale Semantic Reasoning. In: Proceedings of WIMS 2011 (2011)
3. Garofalakis, M., Gehrke, J., Rastogi, R.: Data Stream Management: Processing High-Speed Data Streams. Springer-Verlag New York, Inc. (2007)
4. Della Valle, E., Ceri, S., van Harmelen, F., Fensel, D.: It's a Streaming World! Reasoning upon Rapidly Changing Information. IEEE Intelligent Systems 24(6), 83–89 (2009)
5. Barbieri, D.F., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: C-SPARQL: a Continuous Query Language for RDF Data Streams. Int. J. Semantic Computing 4(1), 3–25 (2010)
6. Barbieri, D.F., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: Incremental Reasoning on Streams and Rich Background Knowledge. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010. LNCS, vol. 6088, pp. 1–15. Springer, Heidelberg (2010)
7. Ren, Y., Pan, J.Z., Zhao, Y.: Towards Scalable Reasoning on Ontology Streams via Syntactic Approximation. In: Proc. of IWOD 2010 (2010)
8. Fensel, D., et al.: Towards LarKC: a Platform for Web-scale Reasoning. In: Proc. of ICSC 2008 (2008)
9. Tresp, V., Huang, Y., Bundschus, M., Rettinger, A.: Materializing and querying learned knowledge. In: Proc. of IRMLeS 2009 (2009)
10. Berrueta, D., et al.: SIOC Core Ontology Specification. W3C Member Submission, W3C (2007)

Automatic Detection of Political Opinions in Tweets

Diana Maynard and Adam Funk

Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello Street, Sheffield, UK
diana@dcs.shef.ac.uk

Abstract. In this paper, we discuss a variety of issues related to opinion mining from microposts, and the challenges they impose on an NLP system, along with an example application we have developed to determine political leanings from a set of pre-election tweets. While there are a number of sentiment analysis tools available which summarise positive, negative and neutral tweets about a given keyword or topic, these tools generally produce poor results, and operate in a fairly simplistic way, using only the presence of certain positive and negative adjectives as indicators, or simple learning techniques which do not work well on short microposts. On the other hand, intelligent tools which work well on movie and customer reviews cannot be used on microposts due to their brevity and lack of context. Our methods make use of a variety of sophisticated NLP techniques in order to extract more meaningful and higher quality opinions, and incorporate extra-linguistic contextual information.

Keywords: NLP, opinion mining, social media analysis.

1 Introduction

Social media provides a wealth of information about a user's behaviour and interests, from the explicit "John's interests are tennis, swimming and classical music", to the implicit "people who like skydiving tend to be big risk-takers", to the associative "people who buy Nike products also tend to buy Apple products". While information about individuals is not always useful on its own, finding defined clusters of interests and opinions can be interesting. For example, if many people talk on social media sites about fears in airline security, life insurance companies might consider opportunities to sell a new service. This kind of predictive analysis is all about understanding one's potential audience at a much deeper level, which can lead to improved advertising techniques, such as personalised advertisements to different groups.

It is in the interests of large public knowledge institutions to be able to collect and retrieve all the information related to certain events and their development over time. In this new information age, where thoughts and opinions are shared through social networks, it is vital that, in order to make best use of this information, we can distinguish what is important, and be able to preserve it, in order

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 88–99, 2012.

to provide better understanding and a better snapshot of particular situations. Online social networks can also trigger a chain of reactions to such situations and events which ultimately lead to administrative, political and societal changes.

In this paper, we discuss a variety of issues related to opinion mining from microposts, and the challenges they impose on a Natural Language Processing (NLP) system, along with an example application we have developed to divulge political leanings from a set of pre-election tweets. While knowing that Bob Smith is a Labour supporter is not particularly interesting on its own, when this information is combined with other metadata, and information about various groups of people is combined and analysed, we can begin to get some very useful insights about political leanings and on factors that impact this, such as debates aired on television or political incidents that occur.

We first give in Section 2 some examples of previous work on opinion mining and sentiment analysis, and show why these techniques are either not suitable for microposts, or do not work particularly well when adapted to other domains or when generalised. We then describe the opinion mining process in general (Section 3), the corpus of political tweets we have developed (Section 4), and the application to analyse opinions (Section 5). Finally, we give details of a first evaluation of the application and some discussion about future directions (Sections 6 and 8).

2 Related Work

Sentiment detection has been applied to a variety of different media, typically to reviews of products or services, though it is not limited to these. Boiy and Moens [1], for example, see sentiment detection as a classification problem and apply different feature selections to multilingual collections of digital content including blog entries, reviews and forum postings. Conclusive measures of bias in such content have been elusive, but progress towards obtaining reliable measures of sentiment in text has been made – mapping onto a linear scale related to positive versus negative, emotional versus neutral language, etc.

Sentiment detection techniques can be roughly divided into lexicon-based methods [2] and machine-learning methods [1]. Lexicon-based methods rely on a sentiment lexicon, a collection of known and pre-compiled sentiment terms. A document's polarity is the ratio of positive to negative terms. Machine learning approaches make use of syntactic and/or linguistic features, including sentiment lexicons. Hybrid approaches are very common, and sentiment lexicons play a key role in the majority of methods. However, such approaches are often inflexible regarding the ambiguity of sentiment terms. The context in which a term is used can change its meaning, which is particularly true for adjectives in sentiment lexicons [3]. Several evaluations have shown that sentiment detection methods should not neglect contextual information [4,5], and have identified context words with a high impact on the polarity of ambiguous terms [6]. Besides the ambiguity of human language, another bottleneck for sentiment detection methods is the time-consuming creation of sentiment dictionaries. One solution to this is a crowdsourcing technique to create such dictionaries with minimal effort, such as the Sentiment Quiz Facebook application[1].

However, sentiment dictionaries alone are not enough, and there are major problems in applying such techniques to microposts such as tweets, which typically do not contain much contextual information and which assume much implicit knowledge. They are also less grammatical than longer posts and make frequent use of emoticons and hashtags, which can form an important part of the meaning. This means that typical NLP solutions such as full - or even shallow - parsing are unlikely to work well, and new solutions need to be incorporated for handling extra-linguistic information. Typically, they also contain extensive use of irony and sarcasm, which are also difficult for a machine to detect.

There exists a plethora of tools for performing sentiment analysis of tweets, though most work best on mentions of product brands, where people are clearly expressing opinions about the product. Generally, the user enters a search term and gets back all the positive and negative (and sometimes neutral) tweets that contain the term, along with some graphics such as pie charts or graphs. Typical basic tools are Twitter Sentiment[2], Twends[3] and Twitrratr[4]. Slightly more sophisticated tools such as SocialMention[5] allow search in a variety of social networks and produce other statistics such as percentages of Strength, Passion and Reach, while others allow the user to correct erroneous analyses. While these tools are simple to use and often provide an attractive display, their analysis is very rudimentary, performance is low, and they do not identify the opinion holder or the topic of the opinion, assuming (often wrongly) that the opinion is related to the keyword.

3 Opinion Mining Process

We have developed an initial application for opinion mining using GATE [7], a freely available toolkit for language processing. The first stage in the system is to perform a basic sentiment analysis, i.e., to associate a positive, negative or neutral sentiment with each relevant tweet. This is supplemented by creating triples of the form <*Person*, *Opinion*, *Political Party*>, e.g., <*Bob Smith*, *pro*, *Labour*> to represent the fact that Bob Smith is a Labour supporter. Given the nature of tweets, we have found that it is fairly rare to see more than one sentiment about the same thing expressed in a single tweet: if, however, two opposing opinions about a political party are mentioned, then we simply posit a neutral opinion at this stage.

Once the triples have been extracted, we can then collect all mentions that refer to the same person, and collate the information. For example, John may be equally drawn towards more than one party, not just Labour, but hates

[1] http://apps.facebook.com/sentiment-quiz

[2] http://twittersentiment.appspot.com/

[3] http://twendz.waggeneredstrom.com/

[4] http://twitrratr.com/

[5] http://socialmention.com/

the Conservatives. His opinion may also change over time, especially during the pre-election phase, or since the recent elections. We thus go beyond typical sentiment analysis techniques which only look at a static opinion at a fixed point in time. This is important because it enables us to make much more interesting observations about political opinions and how they are affected by various events.

4 The Pre-election Twitter Corpus

For the development of our application, we used a corpus of political tweets collected over the UK pre-election period in 2010[6]. The Twitter Streaming API[7] was used to collect tweets from this period according to a variety of relevant criteria (use of hash tags such as #election2010, #bbcqt (BBC Question Time), #Labour etc., specific mention of various political parties or words such as "election", and so on). The tweets were collected in JSON format and then converted to xml using the JSON-Lib library[8]. The corpus contains about 5 million tweets; however it contains many duplicates. De-duplication, which formed a part of the conversion process, reduced the corpus size by about 20% to around 4 million tweets.

The corpus contains not only the tweets themselves, but also a large amount of metadata associated with each tweet, such as its date and time, the number of followers of the person tweeting, the location and other information about the person tweeting, and so on. This information is useful for disambiguation and for collating the information later. Figure 1 depicts a tweet loaded in GATE, with the text and some of the metadata (location, author, and author profile) highlighted. We should note that the method for collecting tweets is not perfect, as we find some tweets which are nothing to do with the election, due to ambiguous words (in particular, "Labour" which could also be a common noun, and "Tory" which could also be a person's name). For future work, we plan a more sophisticated method for collecting and pruning relevant tweets; nevertheless, this quick and dirty method enabled us to get the initial experiments underway quickly.

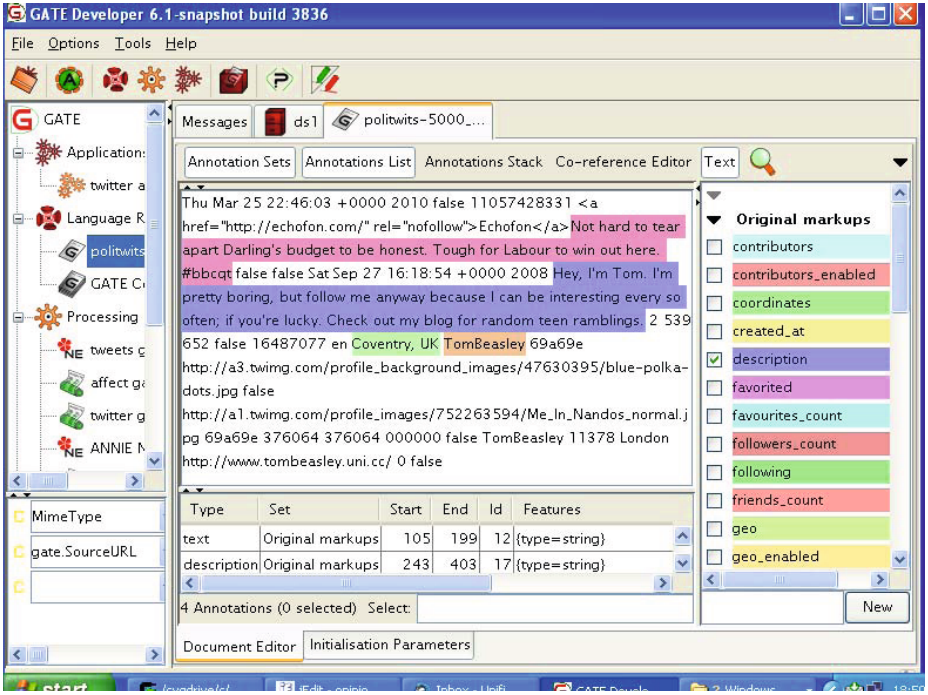
5 Application

The application consists of a number of processing modules combined to form an application pipeline. First, we use a number of linguistic pre-processing components such as tokenisation, part-of-speech tagging, morphological analysis, sentence splitting, and so on. Full parsing is not used because of the nature of the tweets: from past experience, we know it is very unlikely that the quality would be sufficiently high. Second, we apply ANNIE [8], the default named entity recognition system available as part of GATE, in order to recognise named entities in the text (Person, Organisation, Location, Date, Time, Money, Percent).

[6] We are very grateful to Matthew Rowe for allowing us to use this corpus.

[7] http://dev.twitter.com/pages/streaming_api

[8] http://json-lib.sourceforge.net/



Thu Mar 25 22:46:03 +0000 2010 false 11057428331 <a href="http://echofon.com/" rel="nofollow">Echofon</a>Not hard to tear apart Darling's budget to be honest. Tough for Labour to win out here. #bbcqt false false Sat Sep 27 16:18:54 +0000 2008 Hey, I'm Tom. I'm pretty boring, but follow me anyway because I can be interesting every so often; if you're lucky. Check out my blog for random teen ramblings. 2 539 652 false 16487077 en Coventry, UK TomBeasley 69a69e http://a3.twimg.com/profile_background_images/47630395/blue-polka-dots.jpg false http://a1.twimg.com/profile_images/752263594/Me_In_Nandos_normal.jpg 69a69e 376064 376064 000000 false TomBeasley 11378 London http://www.tombeasley.uni.cc/ 0 false

Type	Set	Start	End	Id	Features
text	Original markups	105	199	12	{type=string}
description	Original markups	243	403	17	{type=string}

4 Annotations (0 selected) Select:

MimeType
gate.SourceURL

Original markups
☐ contributors
☐ contributors_enabled
☐ coordinates
☐ created_at
☑ description
☐ favorited
☐ favourites_count
☐ followers_count
☐ following
☐ friends_count
☐ geo
☐ geo_enabled

Fig. 1. Screenshot of a tweet in GATE, with relevant metadata

The named entities are then used in the next stages: first for the identification of opinion holders and targets (i.e., people, political parties, etc.), and second, as contextual information for aiding opinion mining.

The main body of the opinion mining application involves a set of JAPE grammars which create annotations on segments of text. JAPE is a Java-based pattern matching language used in GATE [9]. The grammar rules create a number of temporary annotations which are later combined with existing annotations and converted into final annotations. In addition to the grammars, we use a set of gazetteer lists containing useful clues and context words: for example, we have developed a gazetteer of affect/emotion words from WordNet[10]. These have a feature denoting their part of speech, and information about the original WordNet synset to which they belong. The lists have been modified and extended manually to improve their quality: some words and lists have been deleted (since we considered them irrelevant for our purpose) while others have been added.

As mentioned above, the application aims to find for each relevant tweet, triples denoting three kinds of entity: Person, Opinion and Political Party. The application creates a number of different annotations on the text which are then combined to form these triples.

The detection of the actual opinion (sentiment) is performed via a number of different phases: detecting positive, negative and neutral words (Affect annotations), identifying factual or opinionated versus questions or doubtful statements, identifying negatives, and detecting extra-linguistic clues such as smileys.

Because we only want to process the actual text of the tweet, and not the metadata, we use a special processing resource (the *Segment Processing PR*) to run our application over just the text covered by the XML "text" tag in the tweet. We also use this to access various aspects of the metadata from the tweet, such as the author information, as explained below.

5.1 Affect Annotations

Affect annotations denote the sentiment expressed in the tweet, which could be positive, negative or neutral towards a particular party. These are created primarily by the gazetteer (sentiment dictionary), but the sentiment denoted can then be modified by various contextual factors. First, the gazetteer is used to find mentions of positive and negative words, such as "beneficial" and "awful" respectively. A check is performed to ensure that the part of speech of the gazetteer entry matched and the word in the text are the same, otherwise no match is produced. This ensures disambiguation of the various categories. For example, note the difference between the following pairs of phrases: "Just watched video about awful days of Tory rule" vs "Ah good, the entertainment is here." In the first phrase, "awful" is an adjective and refers to the "days of Tory rule": this would be appropriate as a match for a negative word. In the second phrase, "good" is an adverb and should not be used as a match for a positive sentiment about the entertainment (it does not actually denote that the entertainment itself is good, only that the author is looking forward to the entertainment). Similarly, note the difference between the preposition and verb "like" in the following pair of phrases, which again express very different sentiments about the person in question: "People like her should be shot." vs "People like her."

5.2 Other Clues

Hashtags can also be a source of information about the opinion of the author. Some are fairly explicit, for example #VoteSNP, #Labourfail, while others are more subtle, e.g., #torytombstone, #VoteFodderForTheTories. Currently, we list a number of frequently occurring hashtags of this sort in a gazetteer list, but future work will involve deconstructing some of these hashtags in order to deduce their meaning on the fly (since they are not correctly tokenised, they will not be recognised by our regular gazetteers and grammars). Some hashtags are easier to decipher the meaning of than others: for example, #torytombstone requires some implicit knowledge about the use of the word "tombstone" being used in a derogatory way.

5.3 Opinionated Statements

This phase checks the tweets to see if they are opinionated, or whether they contain questions or doubtful statements. For example, it is hard to tell from the question: "Wont Unite's victory be beneficial to Labour?" whether the author is a supporter of Labour or not, so we posit simply a neutral opinion here. Initially, we match any statement containing an Affect annotation as being opinionated, unless it contains a question, but this could be extended to deal with other cases. We annotate any tweet that contains a question mark (or potentially other distinguishing question-related features) as a Question, and retain it for possible later use, but do not annotate it as an Opinion at this point.

5.4 Negatives

This phase checks the tweet to see if it contains some negative word (as found in a gazetteer list), such as "not", "couldn't", "never" and so on. In most cases, it will reverse the opinion already found: the existing feature value on the Sentiment annotation is changed from "pro" to "anti" or vice versa. More complex negatives include checking for words such as "against", "stop" and so on as part of a noun phrase involving a political party, or as part of a verb phrase followed by a mention of a political party.

5.5 Political Party

Finding the name of the Political Party in the tweet is generally straightforward as there are only a limited number of ways in which they are usually referred to. As mentioned above, however, there is some ambiguity possible. We therefore use other clues in the tweet, where possible, to help resolve these. For example, if "Tory" is part of a person's name (identified by ANNIE), we discard it as a possible political party.

5.6 Identifying the Opinion Holder

Where a Person is identified in the tweet as the holder of the opinion (via another set of grammar rules), we create a Person annotation. If the opinion holder in the pattern matched is a Person or Organization, we create a Person annotation with the text string as the value of an opinion_holder feature on the annotation. If the opinion holder in the pattern matched is a pronoun, we first find the value of the string of the matching proper noun antecedent and use this as the value of the opinion_holder feature. Currently, we only match opinion holders within the same sentence.

However, there may not always be an explicit opinion holder. In many cases, the author of the tweet is the opinion holder, e.g., "I'm also going to vote Tory. Hello new world." Here we can co-refer "I" with the person tweeting. In other cases, there is no opinion holder explicitly mentioned, e.g., "Vote for Labour. Harry Potter would." In this case, we can assume that the opinion is also held by the author. In both cases, therefore, we use "author" as the value of opinion_holder, and get the details of the tweet author from the xml metadata.

5.7 Creating Triples

As described above, we first create temporary annotations for Person, Organization, Vote, Party, Negatives etc. based on gazetteer lookup, named entities and so on. We then use a set of rules to combine these into triples, for example:

<*Person*, *Vote*, *Party*>
"Tory Phip admits he voted LibDem" → <*Phip*, *pro*, *LibDem*>

<*Person*, *Party*, *Affect*>
"When they get a Tory government they'll be sorry." → <*author*, *anti*, *Tory*>

Finally, we create an annotation "Sentiment" which has the following features:

- kind = pro_Labour, anti_LibDem, etc.
- opinion_holder = Bob Smith, author, etc.

Currently, we restrict ourselves to rules which are very likely to be successful, thus achieving high Precision at the expense of Recall. These rules should be eventually expanded in order to get more hits, although Precision may suffer as a result.

6 Evaluation and Discussion

We evaluated the first stage of this work, i.e., the basic opinion finding application, on a sample set of 1000 tweets from the large political corpus (selected randomly by a Python script). We then ran the application over this test set and compared the results. Table 1 gives some examples of the different opinions recognised by the system: it shows the tweet (or the relevant portion of the tweet), the opinion generated by the system (labelled "System") and the opinion generated by the manual annotator (labelled "Key").

Out of 1000 tweets, the system identified 143 as being opinionated (about a political party), i.e., it created a Sentiment annotation for that tweet. We analysed these tweets manually and classified them into the following categories: ProCon, AntiCon, ProLab, AntiLab, ProLib, AntiLib, Unknown and Irrelevant. The first 6 of these categories match the system annotations. Unknown is marked when either a political opinion is not expressed or where it is unclear what the political opinion expressed is, e.g., *"Labour got less this time than John Major did in 1997."* Irrelevant is marked when the tweet is not relevant to politics or the election, e.g., *"i am sooooooooooo bored, want to go into labour just for something to do for a couple of hours :)"*. The distinction between Irrelevant and Unknown is only important in that it tells us which tweets should ideally be excluded from the corpus before analysis: we want to include the Unknown ones in the corpus (even though the system should not annotate them), in order to ensure that the system does not annotate false positives as containing a political sentiment, but not the Irrelevant ones. While only 2 documents out of the 143 were classed as Irrelevant, 29 were classed as Unknown (roughly 20%). This means that roughly

Table 1. Examples of tweets and the opinions generated

Tweet	**System**	**Key**
I just constantly said "Vote Labour" in a tourettes kinda way	pro-Lab	pro-Lab
Daily Mail reveals PM's wife has ugly feet http://bit.ly/b6ZNlK ¡–Eww! Another reason not to vote Labour.	pro-Lab	pro-Lab
Still, can't bring myself to vote tactically for Labour	anti-Lab	anti-Lab
@WilliamJHague If you fancy Interest Rates at 15.8% Vote Tory .... they will throw you out of your house...back to the 80's	pro-Con	anti-Con
Vote Tory to stop them bleating! You know it's worth it.	pro-Con	pro-Con
George Osborne. Reason number 437 not to vote Tory.	anti-Con	anti-Con
Vote Tory or Labour, get Lib Dems. Might as well vote LibDem and have done with it	pro-Lib	pro-Lib
@Simon_Rayner sorry but laughing so much it hurts. Who in their right mind will vote for libdem savage cuts?	anti-Lib	anti-Lib

Table 2. Confusion matrix for evaluation corpus

Key/System	ProCon	AntiCon	ProLab	AntiLab	ProLib	AntiLib	Total
ProCon	**5**	0	0	0	0	0	5
AntiCon	10	**5**	0	2	0	0	17
ProLab	0	0	**69**	2	0	0	70
AntiLab	0	0	4	**4**	0	0	8
ProLib	3	0	1	0	**6**	0	10
AntiLib	0	0	0	0	0	**1**	1
Unknown	10	1	11	5	2	0	29
Irrelevant	0	1	0	1	0	0	2
Total	28	7	85	14	8	1	143

80% of the documents that the system classified with a Sentiment, were in fact opinionated, though not all of them had the correct opinion.

Table 2 shows a confusion matrix for the different sentiments recognised by the system, compared with those in the Key (the manually generated sentiments). This table only depicts the results for those tweets where the system recognised a Sentiment as present: it does not show the Missing annotations (where the system failed to recognise a valid Sentiment). The figures in bold indicate a correct match between System and Key. Overall, the system achieved a Precision of 62.2%, which is promising for work at this early stage.

Unfortunately, it was not feasible in this preliminary evaluation to manually annotate 1000 tweets, so we cannot calculate system Recall easily. However, we can extrapolate some hypothesised Recall based on a smaller sample. We took 150 of the tweets which were not classified as opinionated by the system, and annotated them manually. Of these, 127 (85%) were correct, i.e., were classified as Unknown or Irrelevant by the human annotator. Assuming that this sample is representative, we can predict the Recall. For the task of finding whether a political sentiment exists or not (regardless of its orientation), we get 78%

Precision and predict 47% Recall. Where a document was found to contain a political sentiment, the polarity of this sentiment was correct in 79% of cases. Overall, for the task of both correctly identifying that a document contained a political sentiment, and correctly identifying its polarity, we get 62% Precision and predict 37% Recall.

While the Recall of our system is clearly less than ideal, this is unsurprising at this stage because it has been developed with Precision rather than Recall in mind, i.e., only to produce a result if it is reasonably certain. As we have discussed earlier, there is plenty of scope for improvements to the NLP, in order to improve the Recall of the system. The Precision could also be tightened up further by improving the negation aspect of the rules (most of the errors are related either to not correctly identiying a negative, or by missing out on language nuances such as sarcasm, which are hard for an automated system to deal with). Further evaluation will focus on a larger number of tweets.

It is important also to recognise in the context of evaluation, that performing NLP tasks on social media is in general a harder task than on news texts, for example, because of the style and lack of correct punctuation, grammar etc. Shorter posts such as tweets suffer even more in this respect, and therefore performance of NLP is likely to be lower than for other kinds of text. Also, tweets in particular assume a high level of contextual and world knowledge by the reader, and this information can be very difficult to acquire automatically. For example, one tweet in our dataset likened a politician to Voldemort, a fictional character from the Harry Potter series of books. This kind of world knowledge is unlikely to be readily available in a knowledge base for such an application, and we may have to just accept that this kind of comment cannot be readily understood by auomatic means (unless we have sufficient examples of it occurring).

7 Prospects for Machine Learning

In previous work [11,12] we have obtained good results using SVM-based machine learning (ML) from linguistic features for opinion classification. We plan to experiment with similar data-driven techniques on tweets, although we would probably use the Perceptron algorithm instead, since it is faster and (in our experience) about as accurate for NLP. Our previous experiments were carried out on longer, somewhat more consistently edited texts (film, product and business reviews)—quite unlike the highly abbreviated and inconsistent styles found in tweets—but we obtained good results with unigrams of simple linguistic features, such as tokens and their lemmas, as well as with features derived from SentiWordNet values.

To carry out such experiments successfully on tweets, however, we would need a larger manually annotated corpus—probably 500 tweets annotated with the same 8-way classification used above. This figure is based on our normal use of 5- or 10-fold cross-validation (to stretch the training and test data) for ML as well as on the difficulty of classifying very short texts. We would not necessarily expect the SentiWordNet features to help much with tweets, which are more polysemous and often sarcastic, but it would be interesting to compare them.

8 Conclusions

Typically, opinion mining looks at social media content to analyse people's explicit opinions about an organisation, product or service. However, this backwards-looking approach often aims primarily at dealing with problems, e.g., unflattering comments, while a forwards-looking approach aims at looking ahead to understanding potential new needs from consumers. This is achieved by trying to understand people's needs and interests in a more general way, e.g. drawing conclusions from their opinions about other products, services and interests. It is not sufficient, therefore, to look at specific comments in isolation: non-specific sentiment is also an important part of the overall picture.

One of the difficulties of drawing conclusions from traditional opinion mining techniques is the sparse data issue. Opinions about products and services tend to be based on one very specific thing, such as a particular model of camera or brand of washing powder, but do not necessarily hold for every other model of that brand of camera, or for every other product sold by the company, so a set of very isolated viewpoints is typically identified. The same applies, in some sense, to political viewpoints: a person may not like a particular politician even if they support the party represented by that person, overall. Furthermore, political opinions are often more suject to variations along a timeline than products and brands. A person who prefers Coke to Pepsi is unlikely to change their point of view suddenly one day, but there are many people whose political leanings change frequently, depending on the particular government, the politicians involved and events which may occur. In order to overcome such issues, we need to be able to figure out which statements can be generalised to other models/products/issues, and which are specific. Another solution is to leverage sentiment analysis from more generic expressions of motivation, behaviour, emotions and so on, e.g., what type of person buys what kind of camera, what kind of person is a Labour supporter, and so on. This requires combining the kind of approach to opinion mining described here with additional information about people's likes, dislikes, interests, social groups and so on. Such techniques will form part of our future work.

As discussed earlier, there are many improvements which can be made to the opinion mining application in terms of using further linguistic and contextual clues: the development of the application described here is a first stage towards a more complete system, and also contextualises the work within a wider framework of social media monitoring which can lead to interesting new perspectives when combined with relevant research in related areas such as trust, archiving and digital libraries. In particular, the exploitation of Web 2.0 and the wisdom of crowds can make web archiving a more selective and meaning-based process. Analysis of social media can help archivists select material for inclusion, providing content appraisal via the social web, while social media mining itself can enrich archives, moving towards structured preservation around semantic categories.

Acknowledgements. This research is conducted as part of the EU FP7 project ARCOMEM[9].

References

1. Boiy, E., Moens, M.F.: A machine learning approach to sentiment analysis in multilingual web texts. Information Retrieval 12(5), 526–558 (2009)
2. Scharl, A., Weichselbraun, A.: An automated approach to investigating the online media coverage of US presidential elections. Journal of Information Technology and Politics 5(1), 121–132 (2008)
3. Mullaly, A., Gagné, C., Spalding, T., Marchak, K.: Examining ambiguous adjectives in adjective-noun phrases: Evidence for representation as a shared core-meaning. The Mental Lexicon 5(1), 87–114 (2010)
4. Weichselbraun, A., Gindl, S., Scharl, A.: A context-dependent supervised learning approach to sentiment detection in large textual databases. Journal of Information and Data Management 1(3), 329–342 (2010)
5. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. Computational Linguistics 35(3), 399–433 (2009)
6. Gindl, S., Weichselbraun, A., Scharl, A.: Cross-domain contextualisation of sentiment lexicons. In: Proceedings of 19th European Conference on Artificial Intelligence (ECAI 2010), pp. 771–776 (2010)
7. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics, ACL 2002 (2002)
8. Maynard, D., Tablan, V., Cunningham, H., Ursu, C., Saggion, H., Bontcheva, K., Wilks, Y.: Architectural Elements of Language Engineering Robustness. Journal of Natural Language Engineering – Special Issue on Robust Methods in Analysis of Natural Language Data 8(2/3), 257–274 (2002)
9. Cunningham, H., Maynard, D., Tablan, V.: JAPE: a Java Annotation Patterns Engine 2 edn. Research Memorandum CS–00–10, Department of Computer Science, University of Sheffield (2000)
10. Miller, G.A., Beckwith, R., Felbaum, C., Gross, D., Miller, C., Miller, G.A., Beckwith, R., Felbaum, C., Gross, D., Miller, C., Minsky, M.: Five papers on WordNetk-lines: A theory of memory (1980)
11. Funk, A., Li, Y., Saggion, H., Bontcheva, K., Leibold, C.: Opinion analysis for business intelligence applications. In: First international workshop on Ontology-Supported Business Intelligence (at ISWC). ACM, Karlsruhe (2008)
12. Saggion, H., Funk, A.: Extracting opinions and facts for business intelligence. RNTI Journal E(17), 119–146 (2009)

[9] http://www.arcomem.eu/

The Pragmatics of Political Messages in Twitter Communication

Jurģis Šķilters[1], Monika Kreile[2], Uldis Bojārs[3], Inta Brikše[1],
Jānis Pencis[1], and Laura Uzule[1]

[1] University of Latvia, Department of Communication Studies, Riga, Latvia
[2] University of Oxford, Oxford, United Kingdom
[3] University of Latvia, Advanced Social and Political Research Institute, Riga, Latvia
{jurgisskilters,uldis.bojars,janis.pencis}@gmail.com,
monika.kreile@ling-phil.ox.ac.uk, inta.brikse@lu.lv,
laurauzule@inbox.lv

Abstract. The aim of the current paper is to formulate a conception of pragmatic patterns characterizing the construction of individual and collective identities in virtual communities (in our case: the Twitter community). We have explored several theoretical approaches and frameworks and relevant empirical data to show that the agents building virtual communities are 'extended selves' grounded in a highly dynamic and compressed, linguistically mediated virtual network structure. Our empirical evidence consists of a study of discourse related to the Latvian parliamentary elections of 2010. We used a Twitter corpus (in Latvian) harvested and statistically evaluated using the Pointwise Mutual Information (PMI) algorithm and complemented with qualitative and quantitative content analysis. Special emphasis is given to opinion leaders (journalists, politicians, public relations specialists, academics etc.) in Twitter communication, instead of attempting to cover the entire body of discourse including all Twitter participants.

Keywords: Twitter, virtual identity, social science, political messages.

1 Introduction

In this paper, we explore the pragmatics of political messages in Latvian Twitter communication during the 2010 general election.

The results contain a topical analysis of election discussions as well as an analysis of hashtags and retweeted messages. The fast pragmatic dynamics in Titter communication can be observed through hashtags, showing a rapid reaction of Twitter users to the elections, while top retweets support the findings of content analysis with regard to political sentiment. Content analysis reveals the possibility of significant discrepancies in terms of the cognitive and physical distances between a group and its individual members in their identity generation processes. In view of the results, we propose a hypothesis that reveals correlations between a group and its individual members, the richness of topics, channels of communication, frequency of mention, and connotations and effects of messages.

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 100–111, 2012.

2 Theoretical Background

We assume that the generation of identity takes place through two simultaneous and mutually interdependent social categorization processes – belongingness and differentiation [3,4]. Our study undertakes to examine these two processes in action, constrained by two selection criteria: (1) Twitter messages only, and (2) messages relating directly to national politics. The homogeneity of format and topic draws attention to similarities and differences in content and in discourse strategies.

Twitter is a particularly fruitful resource for this type of analysis because its brevity constraint gives rise to an abundance of shortcut techniques including expressive lexis, the use of abbreviations and hyperlinks for proper names and keywords. Rigid information hierarchies reveal what users presume to be already known and/or shared by their in-group, and are fertile soil for the investigation of presuppositions, cultural common ground, and cultural discrepancies [6,7,15]. This is especially prominent in Twitter discourse about politics, a topic where speakers generally exhibit willingness to report their opinions despite the fact that their perspectives are often conflicting. Although political opinions are usually articulated explicitly, belonging to an identity group[1] may be partly implicit [19].

We focus on mechanisms of self-identification, the formation and maintenance of in-groups and their differentiation from out-groups. The findings attempt to answer the following questions: 1) How are virtual political identities generated and maintained in a condensed public mode of communication? 2) What are the pragmatic instruments that help to engender these processes?

Twitter can also help to understand implicit social categorization. Typically, research on social categorization is conducted using questionnaire or focus group methodologies, mainly addressing explicit political categorization. This study has incorporated some implicit factors of analysis, often crucial in political communication. Approaching human-generated digital content as empirical material for categorization analysis is not new (cp. [9]). Analysis of political messages on Twitter, although not directly focused on categorization, is also provided by several studies (cp. [22]). Several recent studies explore possible correlations between election outcomes and the level of Twitter activity of politicians (US Congress: [14], South Korea: [12]). This study, however, also analyzes political messages created by media organizations and other active users.

2.1 Collocations and Concordance Analysis

Co-occurrence statistics allow to quantitatively project some of a word's semantics grounded in users' categorization performance ([18]). Collocations show the relative most frequent (sometimes stereotypical, implicit) social categories in communication, but the research must be complimented with concordance analysis for semantic complexity. Of course, the output of such a combination of methods concerns the

[1] We define identity as a continuous process where the sense of belongingness to a community interacts with the desire to be a unique individual. A community has an internal and an external structure (relationships within the group and relationships with other groups), and community identity can generate polarization effects.

group (and not individual) patterns of social categorization, and pragmatic effects are related to statistical frequency of language used in communities and not to individual patterns of communication[2]. Importantly, we are not attempting to extrapolate our results to the entire Latvian Twittersphere, but to focus instead on the network of opinion leaders with significant influence on Latvian election discourse in the media.

2.2 Political Messages

Studies show that people frequently have difficulty explicitly articulating their ideology [19]. Thus self-report, focus groups, and questionnaires may often prove inadequate for analyzing political categorization. Ideological labels, moreover, may not correspond to subjective conceptions of beliefs, and undecided voters exhibit a much clearer opinion via implicit tasks than via explicit ones [19]. Political categories are distinguished above all by their extreme polarization (cp. [11,17]). A notable tendency in Twitter communication is that initially informative messages are continually modified to become increasingly polarized [23].

2.3 The Latvian Parliamentary Election 2010

The *Saeima* (the parliament of Latvia) is elected using a proportional multi-partisan representation system for 100 seats. The 2010 election saw 13 competing political parties or their alliances. Candidates from 5 parties were elected: 33 seats for "Unity" (Unity), 29 seats for "Harmony Centre" (HC), 22 seats for the "Union of Greens and Farmers" (UGF), and 8 seats each for the National Association "All For Latvia!"-"TB/LNNK" (NA) and "For a Good Latvia" (FGL). The turnout for the 2010 elections was 63.12%, or approximately 967 000 people.

3 Methodology and Design

The aims of this study are: (1) to build a feasible methodology using content and structural analysis of social media (in particular, Twitter) with respect to political communication; (2) to explore correlations between the election results and the representations of political parties and their candidates in Twitter communication; (3) to explore the identity generation of political actors in pre-election communication on Twitter.

We collected a dataset of tweets covering the election week, performed careful manual extraction work and numerous statistical comparisons. We also created custom tools for analyzing Latvian Twitter content including a concordance tool. We believe that this makes our results, in several respects, even more precise than, e.g., [22], who automatically translated their corpus of empirical data (German tweets) into English and only then processed it with LIWC (Linguistic Inquiry and Word Count).

3.1 Dataset

The dataset consists of one week's collection of Twitter messages (from 28-Sep-2010 to 04-Oct-2010) from a subset of Latvian Twitter users, including 4 days before the

[2] A pragmatic pattern is a typical way of using language in a linguistic community (e.g., in social media).

election, the day of the election (October 2), and 2 days following the election. The total size is 50'032 messages, consisting of: 50% regular tweets; 18% retweets; and 32% replies. There are no publicly available official data about the total number of Twitter participants in Latvia. According to local media experts, the estimate is approximately 40'000 users (November 2010).

In order to choose a topically relevant set of Twitter accounts, we started with a manually selected set that included (1) accounts of political parties and their candidates to the parliament (Saeima); (2) accounts of media organizations, political analysts, and other individuals who write about politics and the election; and (3) accounts of individuals most active in the Latvian Twitter-sphere. This formed an initial set of 179 accounts to follow. We enlarged the set of accounts by (1) retrieving tweets from the current set of accounts; (2) identifying new accounts mentioned in the tweets collected; (3) filtering out accounts not related to Latvia; and (4) repeating this process. The result was a total of 1'377 user accounts to collect tweets from.

We did not choose a random sample to avoid large amounts of redundant data consisting of ordinary discussions unrelated to our research interests - politics, identity generation, and the media. This intentionally selected dataset allows for a more precise analysis of the above research topics.

3.2 Tweet Processing and Analysis

Collected tweets are processed using the NLTK library [1]. The processing of tweets consists of: cleaning the dataset; saving the full tweet data for structure analysis; tokenizing tweets; replacing keywords, where we consolidate the various ways to write the same word or expression and replace it with a single keyword identifier.

Latvian is an inflected language in which the same word may appear in many forms. In the keyword replacement step, we collapse these forms into one keyword. We also replace different ways of writing the same expression (e.g. abbreviations and full names of party names). Since there was no stemming or lemmatizing software for Latvian that we could use, we created our own keyword replacement map for keywords related to elections.

Having processed the tweets, we performed: (1) content analysis in which we examined the textual content of Twitter messages; and (2) structure analysis, in which we examine the metadata in tweets and associated with tweets. The main types of text processing performed in the content analysis phase are concordance lookup, word frequency analysis, and collocation (bigram) analysis. For collocation ranking, we used the Pointwise Mutual Information (PMI) metric [16].

4 Content Analysis

4.1 Representations of the Candidats on Twitter

We made a list of all 1234 candidates competing for seats in the parliament, exploring their representations in selected tweets during the 4 days leading up to the election. Since only a small part of all candidates were represented in Twitter communication (in our dataset) four days before the election, we wished to compare our findings with publicity coverage of the candidates in other mass media in Latvia.

We identified 79 family names of the candidates occurring in collocations in the Twitter dataset, and 170 family names of the candidates occurring in the media monitoring dataset. We distinguish four groups of candidates: (1) those represented both in Twitter and print media and news agencies (44 candidates or 3.56% of all the candidates); (2) those who are represented mostly in Twitter (6.40%); (3) those who are represented mostly in print media and news agencies' releases (7.37%); and (4) those who are mostly not represented in the media we studied (82.67% of all the candidates).

Next, we listed how many personal tweets, collocations and publications occur with each of the family names in various time periods (the average number of collocations of every family name of the candidates four days before the election is 4.68; later, we included only those (9) family names that are statistically significant with respect to their number of collocations ($n \geq 4.68$)). Almost all of these candidates (except one) were elected[3]. They also represent 4 out of the 5 parties elected to the parliament. We analyzed the split of the 100 elected candidates between four previously distinguished groups of candidates. Our calculations show that 32% of elected candidates correspond to the first group (represented in Twitter, print media, and news agencies); 5% correspond to the second group (mostly represented in Twitter); 42% correspond to the third group (mostly represented in print media and news agencies); and 16% correspond to the fourth group (mostly not represented in the media we studied). Based on all of the above, we have formulated a working hypothesis: (1) the more thematically varied and (2) the more frequent the communication, and (3) the more communication channels are used to mention a candidate, the higher the probability that he or she will be elected to parliament.

4.2 Representations of the Parties on Twitter, in Print Media, and by News Agencies Prior to the Election

For names of political parties (Table 1) we listed: (1) how many collocations occur with each name; (2) how many publications from print media and news agencies mention each name; and (3) the results each party has achieved in the election. Every party with an above-average number of collocations in Twitter communication before the election (8.46) was elected to the parliament. An exception is UGF, which was elected despite a below-average number of collocations. We assume that the latter was compensated in the long term by the highest number of publications in print media and news agencies. However, with the high ranking of mention on Twitter before the election (41 collocations), FGL obtained significantly fewer parliament places than "Unity" or other political parties with a lower ranking of mention on Twitter. Initially, it can be assumed that FGL was affected by relatively low publicity rates in print media and news agencies; but in fact, FGL had conducted a more extensive advertising campaign than any other political party). Further investigation points to an important qualitative factor. A review of collocations of FGL and "Unity" in a detailed concordance analysis leads to the observation that the "Unity"

[3] *Election of the 10th Parliament of the Republic of Latvia, October 2, 2010: list and statistics of the candidates. The website of the Central election committee.* Retrieved January 4, 2011 from `http://www.cvk.lv/cgi-bin/wdbcgiw/base/komisijas2010.cvkand10.sak`

collocations feature more positive connotations than the FGL collocations. This allows us to emphasize and modify the above hypothesis regarding the candidates: (1) the more thematically varied and (2) the more frequent the communication, and (3) the more communication channels a political party is mentioned in *positively*, the higher the probability that it will be elected to the parliament.

Table 1. Mentions of political parties (collocations on Twitter, publications in print media and news agencies) and the number of seats in the parliament

Party	Collocations		Publications[4]		Seats
	28.09-01.10	28.09-04.10	27.09-01.10	1 year	
Unity	34	73	276	2799	33 c[5]
UGF	5	20	223	4837	22 c[6]
HC	10	31	232	4674	29
FGL	41	77	230	1647	8
NA	12	32	168	713	8
LP	3	30	0	0	0
FHRUL	3	7	118	1452	0
R	0	6	0	0	0
OPR	2	2	0	0	0
ML	0	1	0	0	0
DL	0	0	0	144	0
LCDU	0	0	0	0	0
PC	0	0	0	0	0

DL = “Daugava for Latvia”; FGL = “For a Good Latvia”; FHRUL = Union "For Human Rights in a United Latvia"; HC = “Harmony Centre”; LCDU = “Latvian Christian Democratic Union”; LP = “The Last Party”; ML = “Made in Latvia”; NA = National Association “All For Latvia!” – “TB/LNNK”; OPR = “For a Presidential Republic”; PC = “People's Control”; R = Social Democratic Alliance “Responsibility”; UGF = “Union of Greens and Farmers”; Unity = Union “Unity”.

4.3 Identity-Generation Processes for Political Parties and Individuals in Twitter Communication

Two political parties – FGL and “Unity” - have significantly higher rankings of mention than other parties. Moreover, their candidates for the post of Prime Minister (Ainārs Šlesers (FGL) and Valdis Dombrovskis (Unity)) have similar rankings of mention. In spite of these similarities, the two have strikingly different election results (“Unity” won the election and got 33 seats in the parliament, with Valdis Dombrovskis approved to the post of Prime Minister, while FGL got only 8 seats in the parliament). This led us to investigate more closely the identity generation of these individuals and organizations through political categorization in pre-election tweets. First, we identified 10 collocations of significantly high ratings for the four name keywords. Secondly, we used concordance analysis to examine the semantics in each collocation.

[4] Publications in print media and news agencies for (1) the election week (27-Sep – 03-Oct); (2) one year (28-Sep-2009 - 03.10.2010). Dates differ from those in tweet collocations due to the source of press data.

[5] c = Formed the ruling coalition.

Table 2. Connotations of keyword topics

	Positive	Neutral	Negative	No of topics
Politician (party)				
Dombrovskis (Unity)	27.59%	68.97%	3.45%	29
Šlesers (FGL)	25.00%	41.67%	33.33%	12
Political party				
Unity	0.00%	54.55%	45.45%	11
FGL	21.43%	21.43%	57.14%	14

We have listed in Table 2 what percentage of the topics bear positive, neutral or negative connotations, and how many topics are covered by each of the keywords. As Table 2 demonstrates, the individual and the organization are categorized similarly in the case of Šlesers and his political party, FGL: both are more related to negative topics than positive ones. The case of Valdis Dombrovskis and his political party "Unity" is different: the individual is mostly categorized by positive or neutral topics, while the political party is categorized by negative or neutral ones. This demonstrates an important effect: the identity generation of an organization, and that of its individual members, may involve significant discrepancies in terms of cognitive versus physical distances[6]. In this case, the cognitive distance between Dombrovskis and "Unity" is bigger than the 'physical' one. This may be in part due to the fact that the "Unity" election campaign focused exclusively on Dombrovskis, promoting him as the principal benefit to the voters. Thus the individual became more cognitively important (prominent) than the whole (an organization). As this study aims to show, cognitive prominence is different for individuals as opposed to communities.

This allows us to expand our hypothesis regarding politicians and political parties as follows: (1) the more thematically varied and (2) the more frequent the communication, and (3) the more communication channels are used to mention a *member* of an organization (in this case, a politician) *positively*[7], the higher the probability that he or she will become cognitively more important than the organization (in this case, the political party) and cause a shift in the perception of the significance of the organization.

5 Structural Analysis

In this section, we analyze Twitter messages by examining implicit and explicit metadata and structural information contained in tweets.

[6] In Spreading-Activation Theory, assuming a correlation between the collocational structure of the corpus and the mental models of its users, collocational structure reflects the cognitive distance between conceptual entities such as political parties and individuals. Indirectly connected nodes are more distant than directly connected ones.

[7] Using manual concordance analysis, connotations are determined and generalized according to three categories (positive, neutral, and negative), determined individually for each tweet. Examples include: "… I shall vote for Dombrovskis, because I trust his professionalism …" (positive); "Šlesers doubts the objectivity of social media …" (neutral); "Dombrovskis: a protégé of corruption or a racketeer?" (negative).

5.1 Hashtag Analysis

Hashtags were used in 2'238 tweets (4.47% of all tweets). In total, 750 different hashtags were used 2'668 times. Most hashtags were used only once. 29.06% of hashtags (218) were used more than once and 2.26% (17) were used at least 20 times.

The most popular hashtag was #velesanas (“election”), used in 459 tweets (17.2% of tweets containing hashtags). Other election-related hashtags that were used at least 20 times include #nobalsoju ("I voted"), #politsports (political sport), #pietiek ("enough!"), #vēlēšanas (#velesanas with Latvian diacritics), #cieti (“solid” – a slogan of FGL), #twibbon (twibbons were used to show party support).

Table 3. Dynamics of top hashtags related to politics (30-Sep-2011 – 04-Oct-2011)

Hashtag	30-Sep	01-Oct	02-Oct	03-Oct	04-Oct
#ir	27	34	21	32	
#pietiek	7	10	16	10	5
#pll	5				
#politika	5				
#politsports	5				
#velesanas		12	346	94	5
#cieti		8	16		
#fail		9	5		9
#sleptareklama		5			
#nobalsoju			71		
#twibbon			60		
#vēlēšanas			35	11	
#velesanas2010			7		

For the purposes of this paper, we limited Table 3 to hashtags related to politics. Most of the top 10 hashtags on election day were related to politics (9 out of 10) and appear in the table. Other days had less election-related tags, but also lower hashtag activity in general. The #velesanas (“election”) hashtag appeared the day before the election and had a remarkable spike in its usage on election day and the day following it, receding back to background level the day after that.

Hashtags that retained popularity for at least 4 days in this 5 day period were the journalism tags #pietiek and #ir. Both refer to publications seen by top Twitter users as prestigeous and integral organizations for investigative journalism. The hashtag #sleptareklama ("hidden advertising") coincided with the appearance of controversial hockey-related advertisements that were suspected of containing hidden political advertising. Hashtags were sometimes used creatively, as when syntactically integrated into a sentence: a notable example is using #ir, the magazine whose name literally means “is”, as a verb: e.g., “There #is still time to form a new coalition”.

Apart from the obvious purpose of attracting attention to major topics, hashtags carry the connotation of familiarity with the object of the tag, be it a topic, an individual, or an organization – at the very least, one must know what is worth tagging. Tags help to define group identity in two recursive ways: by highlighting issues considered important by the group, and by presenting the group as the kind of community where such issues are considered important.

5.2 Analysis of Retweeting

We considered a retweet any Twitter message that contains the string "RT @nickname" (17.68% of the selected dataset). Most retweets start with "RT @nickname", i.e. are marked as such and point to the original message. These results shows more uniformity of retweet formats than reported in [2], possibly a result of more officialized retweet functionality. For further analysis, we used retweets which contained information about the original tweet (i.e. 90.46% of all retweets). An analysis of the top 20 most retweeted posts reveals that 70% of these posts are directly related to the elections; 10% are loosely related; 20% are unrelated.

There were 14 election-related messages among the 20 most retweeted messages. Of these, the majority (8 out of 14) were satirical tweets criticizing a political party or a politician. Seven refer to FGL or to its prominent members Ainārs Šlesers and Andris Šķēle. Other parties mentioned in these retweets were HC and FHRUL (one tweet each). The two most retweeted messages are related to the election.

5.3 Opinion Leaders and In-Group Demarcation Mechanisms

The content of top retweets and hashtags reveals that the opinion leaders in the Latvian Twitter-sphere, the in-group that enjoys the highest popularity and prestige, can be vaguely defined as a group of centrists who see themselves as positioned between two perceived polarities. The cognitive space, as regarded by the in-group, can be characterized thus: to the left are *krievi* ("the Russians"), the parties and their supporters commonly perceived as pro-Muscovite and representing the interests of the Russian-speaking population (HC, FHRUL). To the right are *nēģi* ("the parasites" – an imprecise translation of the word taken from a popular tweet criticizing this group), the nationalist alliance (FGL, NA) that the Twitter opinion leaders see as outdated and highly corrupt, and exploiting their privilege for personal gain. The in-group supports the political alliance "Unity" and particularly its leader, Valdis Dombrovskis, who was subsequently elected Prime Minister.

The fact that the in-group appears to take a centrist position is significant: their output is less polarizing than could be expected of a highly politicized group. Still, there is a clear demarcation of the in-group from both out-groups described above. This is achieved by the opinion leaders of the in-group through several group-identity-generating mechanisms and strengthened by the heightened emphasis on the social self [4], typical of both online communities and political discourse.

Manipulating cognitive distances is relatively easy in the dematerialized virtual space, which facilitates impressions of togetherness and mutual identification within the in-group, on the one hand, but also the distancing of the in-group from out-groups. Perhaps surprisingly, the brevity constraint of Twitter messaging, rather than complicating political categorization, can assist it: the format is well-suited to the in-group's simplified tripartite presentation of the political space. Thus, through repeated tweeting of negative content containing the letters "PLL" or "PCTVL" (acronyms of the names of political parties on the two sides of the perceived spectrum), it is soon enough to write "PLL" or "PCTVL" to evoke a cognitive frame [8] associated with negative content. Clearly, the details of this content will be unique for each user; but as long as there is a basic understanding of a commonality of reference – in this case, of the negativity of the referents – a mention of a party acronym will effectively serve as an invitation to 'fill in the gaps' with each reader's own meaning [13].

Political jokes, abundant in top retweets, work in a similar manner. Provided that humourous effect is usually achieved by inviting the audience to *frame-shift* through an unexpected element [7], political jokes on Twitter are doubly rewarding because they give the audience a feeling of belonging through having understood the frame shift without surrounding linguistic context and through a very limited number of signs. Similarly to a hashtag, a retweet works recursively by simultaneously flaunting an individual's understanding (and hence his belonging to the in-group) and helping to define his individual identity through the content of what is understood and retweeted.

Our corpus shows that power and control are very much the preoccupation of Twitter users, and the independently formed, 'grass-roots' community of top tweeters quickly forms its own behaviour canons. This is typical of online communities, where a myriad of rules and expectations underlie seemingly free, chaotic communication [10]. A popular political message on Twitter is at once an expression of individual and group identity, an invitation to the in-group members to share the opinion expressed, and a warning about the consequence of deviating from the group's norms. By way of illustration, a message retweeted 15 times reads: "I heard that Šlesers won't vote for PLL *either*, because they're said to be thiefs" (our emphasis). In addition to cleverly poking fun at the politician by suggesting he will not vote for his own party, the message succeeds in conveying that the author will not vote for Šlesers, that he assumes that his in-group members will not do so, and that anyone who does vote for Šlesers will be seen as voting for a thief and undermining his or her in-group membership. In short, Twitter conformity mechanisms are just as compact as the medium itself.

Yet without a conforming audience, such successful steering toward a rigidified, formal categorisation would not be possible (we may well judge the above message as successful, since it is on the list of top retweets). The tension between individual opinion and in-group identification (the personal vs. the interpersonal/social self) is resolved through a balance of stereotyping processes: just as the political parties and actors are stereotyped to fit into one of the few cognitive categories carved out for the occasion of the election, so the individual members engage in a certain degree of *self-stereotyping* [20]. Members will be more willing to overlook differences of opinion and concentrate on their commonalities (real or imagined) when membership is seen as beneficial, and particularly if the group is seen as working toward a common goal of some sort – in this case, victory in the parliamentary elections [4]. Because intra-group attraction on Twitter in the run-up to parliamentary election is ideational rather than interpersonal, the in-group *achieves* a high degree of political cohesion in part simply through *perceiving itself* as a cohesive unit.

6 Results and Conclusion

We have formulated a correlation according to which three factors contribute to the efficiency of political messages in the electoral discourse – in particular, *for a given collocation bigram*: (a) the variety of thematic contexts of occurrence, (b) the frequency of mention, (c) positive connotations. (While there are other factors determining efficiency, this study has focused on popularity-oriented facets.) We have therefore extended the results stated by [12,14] regarding the correlation between minority parties, Twitter activity, and election results. The dynamics of Twitter users' interest in the event (the election) can be observed through hashtag usage and top retweets. The latter also convey user sentiment toward political parties and individuals.

We have noted instances of discrepancy between attitudes toward individual politicians as opposed to attitudes toward political groups, and observed that frequent positive mention of an individual can lead to a heightened cognitive significance of this individual, causing the perception of the significance of the relevant organization to recede into the background.

We envision possible applications of this work in analysis tools correlating Twitter dynamics with the structure generated from the following parameters: (a) the variety of occurrence contexts, (b) the frequency of mention, (c) positive connotations (generated semi-automatically). The items which rank highest in such analysis results can be further analyzed manually and a variety of pragmatic effects (stereotyping, presupposition generation, etc.) might be observed.

Finally, we can hypothesize that the user of a microblogging resource such as Twitter extends the sphere of his or her cognitive processing by involving additional interactive structures of communication. Thus, if we assume that the social categorization in a community consists of (a) self-categorization as the most crucial and basic level of identity building, (b) interpersonal communities of individuals, and (c) large-scale social communities (e.g., national identity communities) including sub-communities [4], we could argue that modern-day self-categorization involves a substantial amount of extended cognitive processing offloaded onto the digital environment (in our case, Twitter). In this sense, the results provided by our study can complement research on the extented mind [5,21]. A more detailed analysis of the extended self and offloading effects in cognitive processing is an appealing topic for a future study.

Acknowledgments. This work has been supported by the European Social Fund project «Support for Doctoral Studies at the University of Latvia». The paper is partly funded by the Latvian State-endowed project *Nacionālā identitāte (National identity).*

References

1. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python. O'Reilly (2009)
2. Boyd, D., Golder, S., Lotan, G.: Tweet, Tweet, Retweet: Conversational Aspects of Retweeting on Twitter. In: HICSS-43, January 6. IEEE, Kauai (2010)
3. Brewer, M.B.: The social self: On being the same and different at the same time. Personality and Social Psychology Bulletin 17, 475–482 (1991)
4. Brewer, M.B., Gardner, W.: Who is this "we"? Levels of collective identity and self representations. Journal of Personality and Social Psychology 71, 83–93 (1996)
5. Clark, A.: Pressing the flesh: Exploring a tension in the study of the embodied, embedded mind. Philosophy and Phenomenological Research (2006)

6. Clark, H.H., et al.: Common ground and the understanding of demonstrative reference. Journal of Verbal Learning and Verbal Behaviour 22, 245–258 (1983)
7. Coulson, S., et al.: Looking back: Joke comprehension and the space structuring model. Humor. 19, 229–250 (2006)
8. Fillmore, C.J.: Frame semantics and the nature of language. In: Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech, vol. 280, pp. 20–32 (1976)
9. Glushko, R.J., Maglio, P.P., Matlock, T., Barsalou, L.W.: Categorization in the wild. Trends in Cognitive Sciences 12, 129–135 (2008)
10. Haythornthwaite, C.: Social networks and online community. In: Joinson, A., McKenna, K., Postmes, T., Reips, U.-D. (eds.) The Oxford Handbook of Internet Psychology, pp. 121–137. Oxford University Press, Oxford (2007)
11. Heit, E., Nicholson, S.P.: The opposite of republican: polarization and political categorization. Cognitive Science 34, 1503–1516 (2010)
12. Hsu, C.-L., Park, H.W.: Sociology of hyperlink networks of Web 1.0, Web 2.0, and Twitter: a case study of South Korea. Social Science Computer Review (2010), doi:0894439310382517
13. Langacker, R.W.: Grammar and Conceptualization. Mouton de Gruyter, Berlin (2000)
14. Lassen, D.S., Brown, A.S.: Twitter: the electoral connection? Social Science Computer Review (September 23, 2010), doi:0894439310382749
15. Laurent, J.P., et al.: On understanding idiomatic language: the salience hypothesis assessed by ERP's. Brain Research 1068, 151–160 (2006)
16. Manning, C.D., Schutze, H.: Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge (2003)
17. Medin, D.L., Lynch, E.B., Solomon, K.E.: Are there kinds of concepts? Annual Review of Psychology 51, 121–147 (2000)
18. Mitchell, J., Lapata, M.: Composition in distributional models of semantics. Cognitive Science 34, 1388–1429 (2010)
19. Nosek, B.A., Graham, J., Hawkins, C.B.: Implicit political cognition. In: Gawronski, B., Payne, B.K. (eds.) Handbook of Implicit Social Cognition: Measurement, Theory, and Applications, pp. 548–564. The Guilford Press, New York (2010)
20. Smith, E.R., Henry, S.: An in-group becomes part of the self:Response time evidence. Personality and Social Psychology Bulletin 22, 635–642 (1996)
21. Spivey, M., Richardson, D., Fitneva, S.: Thinking outside the brain: Spatial indices to linguistic and visual information. In: Henderson, J., Ferreira, F. (eds.) The Interface of Vision, Language, and Action, pp. 161–189. Psychology Press, New York (2004)
22. Tumasjan, A., Sprenger, T.O., Sandner, P.G., Welpe, I.M.: Predicting elections with twitter: what 140 charcters reveal about political sentiment. In: Proceedings of the 4th International AAAI Conference on Weblogs and Social Media, pp. 178–185 (2010)
23. Yardi, S., Boyd, D.: Dynamic debates: an analysis of group polarization over time on Twitter. Bulletin of Science, Technology & Society 30(5), 316–327 (2010)

A Knowledge Dashboard for Manufacturing Industries

Suvodeep Mazumdar[1], Andrea Varga[2], Vita Lanfranchi[2],
Daniela Petrelli[3], and Fabio Ciravegna[2]

[1] Information School
[2] OAK Group, Department of Computer Science
University of Sheffield, Regent Court – 211 Portobello Street,
S1 4DP Sheffield, UK
s.mazumdar@sheffield.ac.uk,
{a.varga,v.lanfranchi,f.ciravegna}@dcs.shef.ac.uk
[3] Art & Design Research Centre, C3RI, Sheffield Hallam University,
Furnival Building, 153 Arundel St., S1 2NU, Sheffield, UK
d.petrelli@shu.ac.uk

Abstract. The manufacturing industry offers a huge range of opportunities and challenges for exploiting semantic web technologies. Collating heterogeneous data into semantic knowledge repositories can provide immense benefits to companies, however the power of such knowledge can only be realised if end users are provided visual means to explore and analyse their datasets in a flexible and efficient way. This paper presents a high level approach to unify, structure and visualise document collections using semantic web and information extraction technologies.

Keywords: Semantic Web, Information Visualisation, User Interaction.

1 Introduction

Modern manufacturing is a complex domain where productivity and efficiency are strongly affected by a broad range of factors such as site locations, cultural values, management decisions and communication capabilities. For example, large manufacturing organizations are usually globalised, with facilities geographically distributed, making use of multiple manufacturing machines, interacting with several suppliers and warehouses. Also, a recent trend in large organisations has been the presence of dynamic, interdisciplinary working groups and communities of practice who require rapid, flexible customisation of information to their specific needs [1]. At the same time, the information they generate needs to be shared with the rest of the organisation, and hence, must be presented to other communities in ways that can be easily understood (and correctly interpreted) and reused [2].

The underlying commonality between these phenomena is information availability: if information is captured, stored and shared between different departments and locations then efficient communication can be reached and stronger support for managerial decisions can be provided. Unfortunately this information is often collected in a wide variety of formats (e.g., text files, images, PDF documents) and dispersed in independent repositories, including shared directories, local and

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 112–124, 2012.

company-wide databases, ad hoc information systems, etc. Critical knowledge may be hidden in the huge amount of manufacturing data, and the cost of exhaustively identifying, retrieving and reusing information across this fragmentation is very high and often a near impossible task.

This paper presents how Semantic Web and Information Extraction (IE) technologies can be adopted to unify such collections of documents and formalize their knowledge content, bringing together information from different domains, which can feed into organisational knowledge. Visualisation techniques can then be applied on top of the semantically structured data to explore, contextualise and aggregate it, offering multiple perspectives on the information space and provide analytic tools that could support users in spotting trends and identifying patterns and relationships. In order to achieve this goal two steps are required:

- Knowledge Acquisition: acquiring information from different documents and corpora and semantically structuring it in a semi-supervised manner.
- Knowledge Visualisation: creating multiple views over the semantic knowledge space.

Our methodology is innovative compared to previous literature (analysed in Section 2) as it defines the Knowledge Acquisition and Visualisation steps at an abstract level: the use of ontologies to extract, structure and visualise information make our approach flexible, reusable and extensible.

The Knowledge Acquisition and Visualisation steps will be described in details in Section 3, before providing implementation details (Section 4) and discussing conclusions and future work (Section 5).

The following scenario (taken from SAMULET[1], an existing research project on advanced manufacturing in the aerospace industry in which the authors are involved) has been considered as a foundation for the work: in a manufacturing industry a huge number of components are produced every day based on design data provided by Design departments, and are reused in other divisions of the company. When these components are produced manufacturing data is collected such as manufacturing time, location of the plant and of the manufacturing machine, type of component and details (possibly linked to design data). Additional information includes the person and machine responsible for the production, manufacturing costs and so on. This data is collected in a wide variety of formats (e.g. Excel spreadsheets, images, Word Documents), stored in independent repositories and often distributed using personal channels (such as e-mails, or shared network drives).

Manufacturing data are essential to resolving any issue that may arise on a component, in order to be able to clearly identify the driving factors behind the issue and to discover any significant trends or patterns related to individual manufacturing units/machines/personnel. Identifying non-obvious patterns in the data is fundamental to increasing productivity and efficiency: for example, a consistently poorly performing machine may be over-shadowed by a well performing manufacturing unit – data analysis and visualisation would help in spotting such trends and support putting corrective measures in place.

[1] SAMULET project, `http://www.rolls-royce.com/investors/news/2009/280709_research_factories.jsp` Last Accessed 14/04/2011.

2 Related Work

Our approach aims to provide a consistent and coherent environment for knowledge exploration in the manufacturing domain, encompassing knowledge acquisition and knowledge visualisation techniques. Related work in both these areas is now analysed, with particular emphasis on the adoption in the manufacturing domain.

2.1 Knowledge Acquisition

Traditional machine learning (ML) approaches for knowledge acquisition in manufacturing started to gain much attention only in recent years [3-10], mostly because the majority of the ML algorithms and tools require skilled individuals to understand the output of ML process [3]. However there has been some work on using traditional ML techniques for specific areas (such as fault detection, quality control, maintenance, engineering design, etc.) employing classification [6,7], clustering [8] and association rule mining [9,10] algorithms [3-5]. Classification algorithms were used for categorising data into different classes, for example classifying defects in the semi-conductor industry [5]. [6] employed a hybrid approach combining neural networks and decision tree classification algorithms for recognising false classifications in control chart pattern recognition (CCPR) thus facilitating quality control. [7] used decision tree algorithms for producing classification rules which were then saved in the competitive decision selector (CDS) knowledge bases enabling efficient job shop scheduling. Clustering algorithms were also used to group similar data into clusters, for example clustering the orders into batches for speeding up the product movement within a warehouse [5]. [8] applied fuzzy c-means clustering algorithm for identifying changes in traffic states thus improving the traffic management systems. Association rule mining algorithms were used to identify relationships among the attributes describing the data. [9] used association rule mining for detecting the source of assembly faults, thus improving the quality of assembly operations. [10] extracted association rules from historical product data to identify the limitations of the manufacturing processes. This information can then be used to improve the quality of the product and identify the requirements for design change.

Despite the increased interest, most of these approaches still lack portability and require a large amount of annotated data to achieve high performance, which is usually tedious and costly [13] to obtain. Furthermore recent advances in domain adaptation show that traditional machine learning (ML) approaches for IE are no longer the best choices [11,12]. These algorithms work only well when the format, writing style in which the data (e.g. manufacturing time, location of the plant and the machine) is presented is similar across different corpora [11,12]. In dynamic and heterogeneous corpora, these ML based systems need to be rebuilt for each corpus or format, making them impractical in many scenarios [11], such as the one presented in this paper. To enable effective knowledge capture in manufacturing our approach employs an adaptable IE framework based on domain adaptation techniques, as presented in Section 3.

2.2 Knowledge Visualisation

Information visualisation techniques have been extensively adopted in the manufacturing domain to display and illustrate different processes such as simulation of model verification and validation, planning, decision making purposes and so on [14, 20]. Though most simulation results are based on data models, visualisations are essential to efficiently communicate information to end-users [15]. For example visualising CAD (Computer Aided Design) models enriched with performance scores provides analysts insights into the performances of different manufacturing units; alternative techniques provide ways for manufacturing units to validate their products against software models [14] (to evaluate compliance of manufacturing units to design).

Commercial tools generally focus on 3D visualisations of manufacturing models, factories, machines and so on. Examples of such commercially available tools used in the manufacturing industry include Rockwell's FactoryTalk[2] (remote monitoring of manufacturing processes); Autodesk's 3ds Max[3] and Maya[4] (modelling of product designs, animation, virtual environments); VSG's OpenInventor[5] (3D Graphics toolkit for developing interactive applications); DeskArtes ViewExpert[6] (viewing, verifying, measuring CAD data); Oracle's AutoVue[7] (Collaboration tool to annotate 3D or 2D models). These 3D commercial tools are also adopted in other industries like gaming, animation and so on [17]. However the high cost of 3D hardware and software makes this option unfeasible for smaller companies [16].

3D visualisation techniques have also been investigated in academic works, such as Cyberbikes, a tool for interaction with and exploration using head-mounted displays. [21] presents another example of 3D visualisation, providing factory floor maps which use animations to convey real-time events.

Using visualisations to communicate high-quality data in manufacturing scenarios can greatly reduce the amount of time and effort taken by engineers to resolve an issue: in a study by [18], engineers provided with animated visualisations combining several steps of a simulation could substantially reduce their analysis time. [22] discusses how factory map visualisation based navigation can often provide means to significantly reduce the cognitive load on analysts monitoring a typical manufacturing factory, when compared to list-based navigation of factory machines and their performances. Our approach takes inspiration from this work in aiming to provide efficient visualisation techniques that will reduce engineers cognitive workload and facilitate knowledge analysis.

[2] FactoryTalk, `http://www.rockwellautomation.com/rockwellsoftware/factorytalk/` Last Accessed 14/04/2011.

[3] AutoDesk 3ds Max, `http://usa.autodesk.com/3ds-max/` Last Accessed 14/04/2011.

[4] AutoDesk Maya, `http://usa.autodesk.com/maya/` Last Accessed 14/03/2011.

[5] VSG OpenInventor, `http://www.vsg3d.com/open-inventor/`sdk Last Accessed 14/04/2011.

[6] DeskArtes ViewExpert, `http://www.deskartes.com/` Last Accessed 14/04/2011.

[7] Oracle AutoVue, `http://www.oracle.com/us/products/applications/autoVue/index.html` Last Accessed 14/04/2011.

3 Adding Semantics to the Manufacturing Domain

Given the large scale and the heterogeneity both in data types and data formats, automatic techniques are required to process the data, unifying the document collections and formalising their knowledge content. In the following we distinguish between data, information and knowledge as proposed in [27]. Namely, data refers to the basic raw unit without any implicit meaning, information refers to data enhanced with context and perspective, and knowledge is information connected by patterns and relations. In our case the outcome of our Information Extraction framework is considered knowledge as it extracts entities and relations and assigns semantic meaning to them.

Our approach (shown in Figure 1) is therefore based on the use of a common knowledge representation in the form of ontologies describing the manufacturing domain. The ontologies are created manually so that the high-level ontology covers the generic manufacturing scope (common concepts and relationships between them), and the local ontologies (interlinked by the over-arching high-level ontology) capture the information specific to the different corpora. An adaptable Information Extraction framework considering the high-level ontology then extracts the common concepts across the corpora, thus avoiding ontology mapping and integration (see Section 3.1).

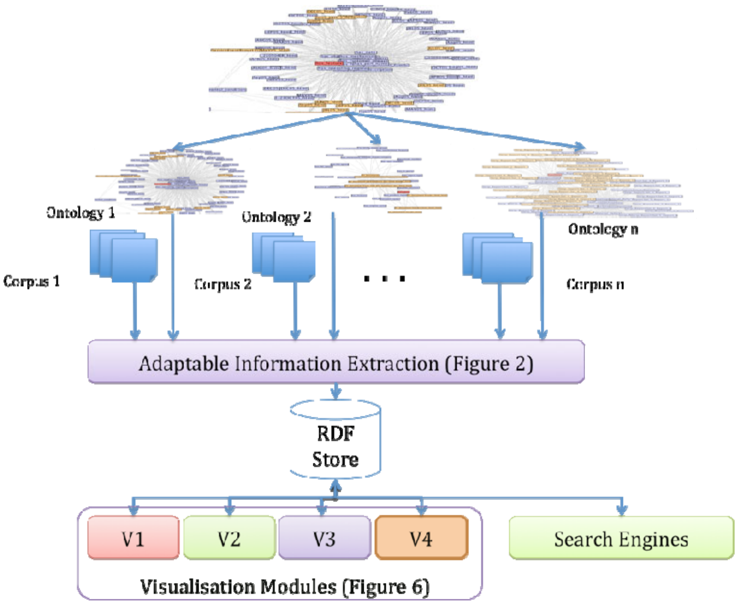


Fig. 1. The knowledge acquisition and visualisation process

The extracted information is then stored in a RDF store and available for query and visualisation (see Section 3.2).

3.1 Adaptable Information Extraction Framework

The adaptable Information Extraction (IE) framework runs in a semi-supervised manner over the (automatically converted) textual versions of the documents in each corpus, extracting the relevant entities and relations and mapping them to the ontological concepts. The IE process is composed of two steps:

- Manual annotation of a subset of data by domain experts for training purposes.
- Unsupervised domain adaptation and annotation of the remaining documents using a Support Vector Machine (SVM)[8] [25] classifier.

Whilst this approach is common in literature [11,12] the novelty is in the portability of the classifier between different corpora with minimal supervision (using only a small amount of human annotations). For each new corpus (and document type) the initial classifier is augmented applying a feature representation approach [11,12] inspired on [29]. That is, the words from all the corpora are first clustered into semantic topics using Latent Dirichlet Allocation [28] topic model. Then new semantic features consisting of a set of most probable topics for each word are added to the classifier. This approach makes our IE system flexible and adaptable, enabling efficient knowledge acquisition across corpora.

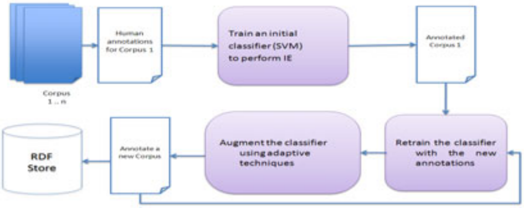

Human annotations for Corpus 1
Train an initial classifier (SVM) to perform IE
Annotated Corpus 1
Corpus 1 .. n
RDF Store
Annotate a new Corpus
Augment the classifier using adaptive techniques
Retrain the classifier with the new annotations

Fig. 2. The adaptable information extraction process

The extracted knowledge (ontology-based annotations) is then stored in a triple store in the form of RDF triples and used later for semantic visualisation. The current implementation of the IE framework also applies a terminology recognition [26] module for domain specific information extraction (e.g. type of component) within the SAMULET project, however the scope of the IE system is more generic and allows extracting domain independent entities and relations too (e.g. person, time, location).

3.2 Knowledge Dashboard

Our approach focuses on providing multiple knowledge visualisations at different granularity levels, using a semantic knowledge dashboard to support users in quickly

[8] LibSVM tool: `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/` Last Accessed 14/04/2011.

gathering a broad insight of their datasets from differing perspectives. This approach is based on a set of interlinked ontologies (as explained in Section 3), which structure the knowledge from the different corpora and define relations between found entities. For each semantic entity type in the knowledge space a set of possible visualisations is defined: automatic inferences are then made on the type of entities and relations stored in the knowledge space to create the visualisation widgets. These visualisations can be customised to suit the user task, needs and preferences.

A dashboard interaction paradigm has been chosen as it provides large amounts of information in one interface, without compromising on clarity [23] and it is an increasingly common visualisation paradigm thanks to its adoption by several well-known websites like igoogle[9] and BBC[10]. Such an approach offers the possibility of dynamically choosing the best visualisation tool for the task in hand, as differently represented data can reveal different insights.

A detailed scenario is now presented to highlight the features of the knowledge dashboard and the interaction possibilities. In our hypothetical scenario, a manufacturing engineer (Bruce) working at a large aerospace organisation has access to six types of documents from different departments:

- Machine Performance Reports - describing operational performances of machines at manufacturing sites;
- Site Performance Reports - describing the overall performance of manufacturing sites;
- People Pages – websites of various individuals and authors of the reports;
- Machine Testing Reports – describing the findings of laboratory testing on machines at manufacturing sites;
- Quality Documents – reports discussing the outcome of various quality tests on manufactured products.
- Service Event Reports – reports discussing various service and maintenance operations conducted on engines over their lifetime.

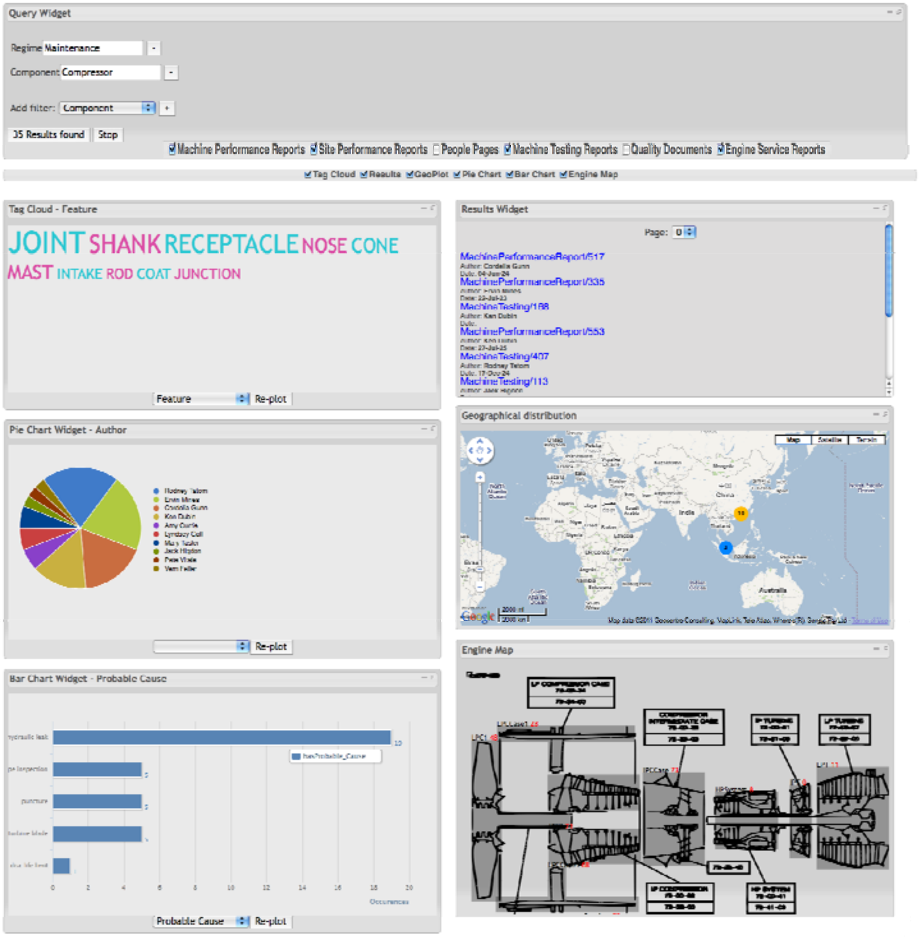
These different report types have been analysed using our adaptable IE framework and semantic knowledge has been extracted and stored in a unified knowledge base. Visualisation ontologies have been defined for the different entities and relations and for the user preferences. These ontologies are used by the knowledge dashboard to automatically build the knowledge space and visualisation widgets. The selections of the visualisations are based on various features such as user preferences, usage history, current task, scale of retrieved datasets and types of data. The visualisation ontologies are essentially classifications of existing visualisations based on these parameters. Once a dataset is retrieved, the visualisation ontologies are used to infer the most effective visualisations for the dataset and users.

In our scenario, Bruce is investigating a condition where a lot of enquiries have been made to the manufacturing teams while service engineers were inspecting compressors of several engines during maintenance. Bruce first selects the relevant

[9] iGoogle interface, `http://www.google.com/ig`, Last Accessed 04/03/2011.

[10] BBC interface, `http://www.bbc.co.uk/`, Last Accessed 04/03/2011.

document sets from the combo boxes provided in the query interface. He then selects the filters ‘Regime’ and ‘Component’ and enters his query (‘maintenance’ and ‘compressor’ respectively).



Query Widget

Regime: Maintenance

Component: Compressor

Add filter: Component

35 Results found

☑ Machine Performance Reports ☑ Site Performance Reports □ People Pages ☑ Machine Testing Reports □ Quality Documents ☑ Engine Service Reports

☑ Tag Cloud ☑ Results ☑ GeoPlot ☑ Pie Chart ☑ Bar Chart ☑ Engine Map

Results Widget

Page: 0

MachinePerformanceReport/517
Author: Cordelia Gunn
Date: 04-Jan-24

MachinePerformanceReport/335
Author: Erwin Minns
Date: 23-Jul-23

MachineTesting/188
Author: Ken Dubin
Date:

MachinePerformanceReport/553
Author: Ken Dubin
Date: 27-Jul-25

MachineTesting/407
Author: Rodney Telom
Date: 17-Dec-24

MachineTesting/113
Author: Jack Higdon

Fig. 3. Knowledge Dashboard

This initiates several queries to be sent to the backend from the interface. Bruce is then provided with several widgets (Figure 3), each of which present different facets (powered by different relations in the underlying semantic knowledge): the tag cloud informs Bruce that out of all the documents retrieved, most of the discussion has been related to features ‘joint’, ‘Shank’ and ‘Receptacle’ – this could indicate which manufacturing machines might be responsible. The bar chart indicates that most of the documents discussing engine service events have also discussed ‘hydraulic leaks’. The engine map groups the documents by the components they discuss – this shows how documents discussing ‘compressors’ also refer to other related components.

These components are then displayed as grey areas, along with counts of how many documents have been found for each component. The pie chart provides a plotting by document authors – this enables Bruce to contact authors for further information and advice. The geographical plot provides the locations of manufacturing sites that are responsible for producing the components being described in the datasets. Using such visualisations, Bruce can now answer several common questions often asked during investigations: Where are the manufacturing machines located? What parts of an engine have the machines manufactured? What are the features of the parts that are being manufactured? Who are responsible for the manufacturing sites? From the multiple visualisation layers a summation of the knowledge emerges that can highlight previously unseen trends, patterns and issues/relations.

Thanks to the semantic knowledge and the background ontologies, the document collections can be visualised at different levels of granularity. For example an encompassing visualisation is achieved by displaying the whole document collections and comparing them, to show a high level view on the available facets without having to look at the individual document instances. The widgets are interactive, allowing zooming and selecting the preferred granularity level, from document to instance level. This follows the well-known principle of "overview first, zoom and filter, then details-on-demand" [24].

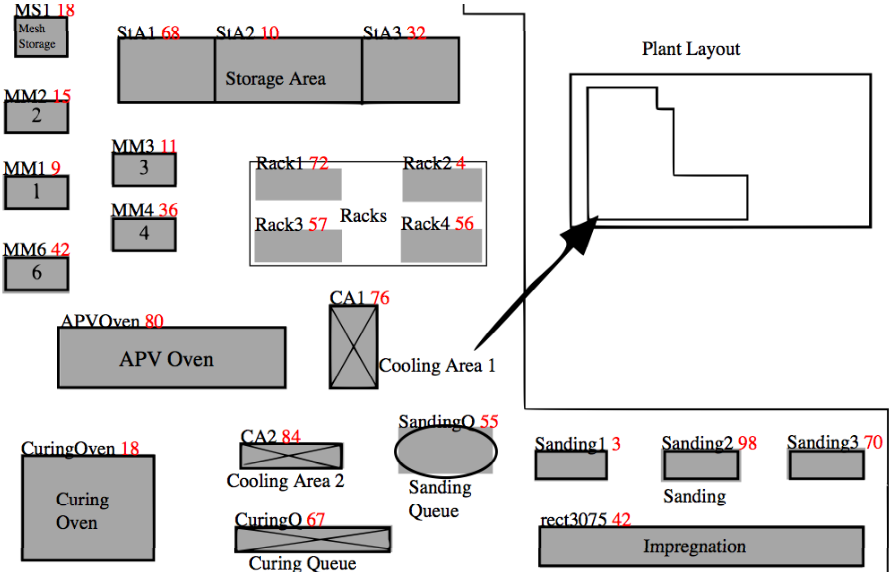


Fig. 4. Floor plan visualisation of knowledge instances

In our scenario if Bruce needs to analyse the performances of the organization from a manufacturing unit point-of-view, he can explore the knowledge space using a geographic view, then zooming in on an individual manufacturing site to reveal the site's floor plan along with the positions of the manufacturing machines. This floor plan is then enriched with performance statistics of the machines, extracted from the Site and Machine performance reports, as shown in Figure 4.

Users can also choose to look at the information from the product point-of-view, by clicking on sensitive areas of the engine, which loads a detailed view of the area of interest, enriched with instances from the documents returned as shown in figure 5. The documents are now grouped into different sections, which are shown as shaded areas- the numbers beside each section indicate the number of retrieved documents related to that section.

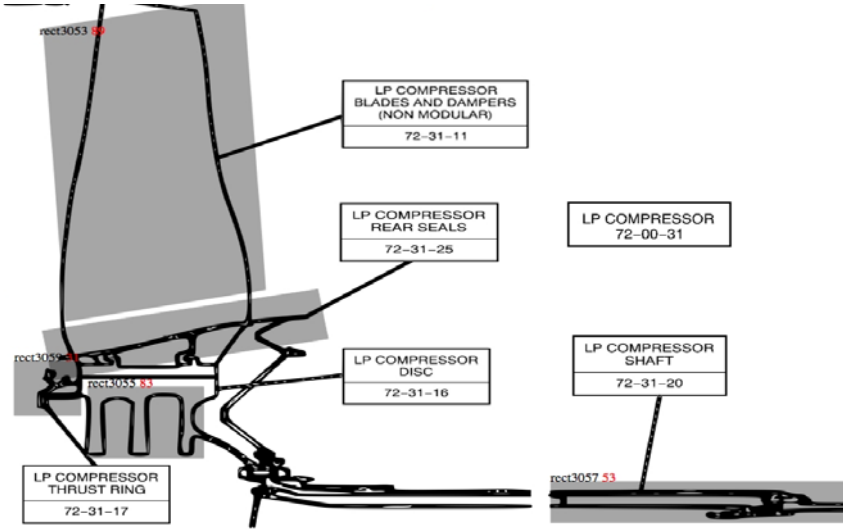


Fig. 5. Detailed view of engine, enriched with knowledge instances

4 Implementation

The system implementation is segmented into a knowledge acquisition and a knowledge exploration system. The knowledge acquisition system is an off-line process implemented in Java. The knowledge visualisation is a web-based dynamic and real-time application, consisting of a javascript frontend and a php backend that communicate using SPARQL queries over a semantic triplestore. The frontend is in charge of interpreting the user interactions and transforming them into corresponding SPARQL queries. For example, clicking on a section of a pie chart would be interpreted as a SPARQL SELECT query. These queries are then transmitted to the backend, which forwards the queries to triplestores. The results from the triplestores are then received by the backend and converted to JSON objects for visualisation in the interface. The system architecture is described in the Figure 6. The block in the right side of the figure shows the front end, while the left side shows the backend processes.

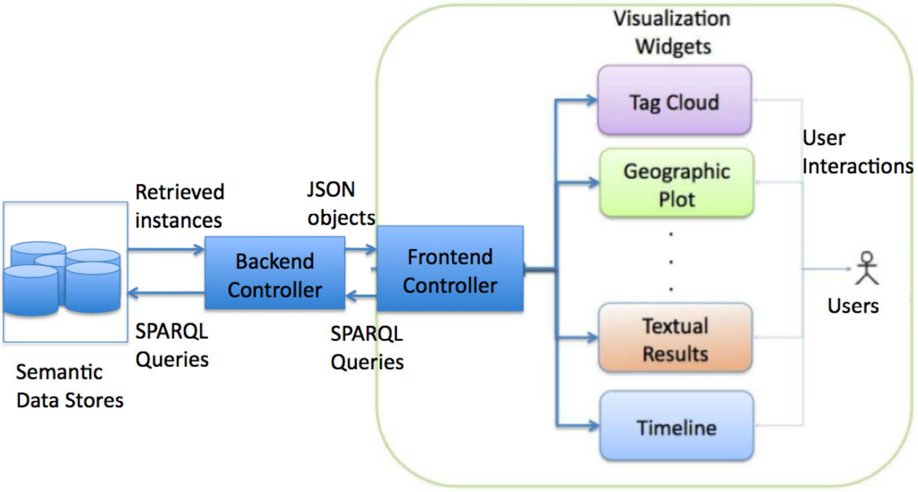


Fig. 6. Visualisation System Architecture

5 Discussion and Conclusions

This paper presented the approach developed during ongoing research work for a project about knowledge management in the manufacturing industry, focusing on how Semantic Web and Information Extraction technologies can be adopted to acquire knowledge from heterogeneous and disparate data whilst providing visualisations to explore, contextualise and aggregate the data, offering multiple perspectives on the knowledge space.

The developed approach is high level and domain independent as it is based on ontologies to structure and visualise knowledge it can be easily applied to a wider context than the manufacturing one. For example it could be applied to any business unit inside a large organisation (i.e. design, service and manufacturing). Expanding the domain will enable organisations to create a large integrated knowledge space available for sharing and reuse.

Future work will concentrate on extending our methodology to different corpora and in enriching the visualisation techniques to better match the user needs. As the project adopts a participatory design paradigm, real users are constantly providing feedbacks on mock-ups and vision demonstrators, to make sure the final prototype will be meeting their needs. This will be complemented by a comparative study of the developed prototype and the current software search systems being used by engineers. Moreover a final user evaluation will be carried out in a real-life scenario to assert the user satisfaction and acceptance of the new technology and a separate in-vitro evaluation will be conducted to test the efficiency and efficacy of the Adaptable IE framework in terms of precision, recall and F-Measure.

Acknowledgments. This research has been supported by the SAMULET project, co-funded by Technology Strategy Board and Rolls-Royce plc. For privacy and security reasons the dataset under analysis is private and all images and data samples have been anonymised.

References

1. Wenger, E.: Communities of Practice: Learning, Meaning, and Identity. Cambridge University Press (1998)
2. Bhagdev, R., Chakravarthy, A., Chapman, S., Ciravegna, F., Lanfranchi, V.: Creating and Using Organisational Semantic Webs in Large Networked Organisations. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 723–736. Springer, Heidelberg (2008)
3. Harding, J.A., Shahbaz, M., Srinivas, Kusiak, A.: Datamining in manufacturing: A review. American Society of Mechanical Engineers (ASME). Journal of Manufacturing Science and Engineering 128(4), 969–976 (2006)
4. Wang, K.: Applying data mining to manufacturing: the nature and implications. Journal of Intelligent Manufacturing of Intelligent Manufacturing (2007)
5. Choudhary, A., Harding, J., Tiwari, M.: Data mining in manufacturing: a review based on the kind of knowledge. Journal of Intelligent Manufacturing 20(5), 501–521 (2009)
6. Guh, R.S.: Real time pattern recognition in statistical process control: A hybrid neural network/decision tree-based approach. Proceedings of the Institution of Mechanical Engineers. Journal of Engineering Manufacture (2005)
7. Kwak, C., Yih, Y.: Data mining approach to production control in the computer integrated testing cell. IEEE Transactions on Robotics and Automation (2004)
8. Crespo, F., Webere, R.: A methodology for dynamic datamining based on fuzzy clustering. Fuzzy Sets and Systems (2005)
9. Cunha, D., Agard, B., Kusiak, A.: Data mining for improvement of product quality. International Journal of Production Research (2006)
10. Shahbaz, M., Srinivas, Harding, J.A., Turner, M.: Product design and manufacturing process improvement using association rules. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture (2006)
11. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 22(10), 1345–1359 (2010)
12. Jing, J.: A literature survey on domain adaptation of statistical classifiers. Technical report (2008)
13. Ciravegna, F., Dingli, A., Petrelli, D., Wilks, Y.: Timely and nonintrusive active document annotation via adaptive information extraction. In: Proc. Workshop Semantic Authoring Annotation and Knowledge Management (2002)
14. Rohrer, R.: Visualization and its importance in manufacturing simulation. Industrial Management (1996)
15. Rohrer, M.W.: Seeing is believing: the importance of visualization in manufacturing simulation. In: Proceedings of the 2000 Winter Simulation Conference (2000)
16. Kamath, R.S., Kamat, R.K.: Development of cost effective 3D stereo visualization software suite for manufacturing industries. Indian Journal of Science and Technology (2010)
17. Agrusa, R., Mazza, V.G., Penso, R.: Advanced 3D Visualization for Manufacturing and Facility Controls. Human System Interactions (2009)

18. Edgar, G.W.: Visualization for non-linear engineering FEM analysis in manufacturing. In: Proceedings of the 1st Conference on Visualization 1990 (1990)
19. Gausemeier, J., Ebbesmeyer, P., Grafe, M., Bohuszewicz, O.v.: Cyberbikes - Interactive Visualization of Manufacturing Processes in a Virtual Environment. In: Proceedings of the Tenth International IFIP WG5.2/WG5.3 Conference on Globalization of Manufacturing in the Digital Communications Era of the 21st Century: Innovation, Agility, and the Virtual Enterprise (1999)
20. Greif, M.: The visual factory: building participation through shared information (1989)
21. Zhong, Y., Shirinzadeh, B.: Virtual factory for manufacturing process visualization. Complexity International (2008)
22. Stowasser, S.: Hybrid Visualization of Manufacturing Management Information for the Shop Floor. In: Human-Computer Interaction: Theory and Practice (Part 2), vol. 2 (2008)
23. Few, S.: Information Dashboard Design: The Effective Visual Communication of Data. 3900693099. O'Reilly Media (2006)
24. Shneiderman, B.: The eyes have it: A task by data type taxonomy of information visualization. In: Bederson, B., Shneiderman, B. (eds.) The Craft of Information Visualization. Morgan Kaufman, San Francisco (2003)
25. Joachims, T.: Estimating the generalization performance of a SVM efficiently. In: Proceedings of International Conference on Machine Learning (2000)
26. Butters, J., Ciravegna, F.: Authoring Technical Documents for Effective Retrieval. In: Cimiano, P., Pinto, H.S. (eds.) EKAW 2010. LNCS, vol. 6317, pp. 287–300. Springer, Heidelberg (2010)
27. Ackoff, R.L.: From Data to Wisdom. Journal of Applied Systems Analysis 16 (1989)
28. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet Allocation. Journal of Machine Learning Research (2003)
29. Guo, H., Zhu, H., Guo, Z., Zhang, X., Wu, X., Su, Z.: Domain adaptation with latent semantic association for named entity recognition. In: Proc. HTL-NAACL, pp. 281–289 (June 2009)

FREyA: An Interactive Way of Querying Linked Data Using Natural Language

Danica Damljanovic[1], Milan Agatonovic[2], and Hamish Cunningham[1]

[1] Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello Street, Sheffield, UK
{d.damljanovic,h.cunningham}@dcs.shef.ac.uk
[2] Fizzback, London, United Kingdom
magatonovic@fizzback.com

Abstract. Natural Language Interfaces are increasingly relevant for information systems fronting rich structured data stores such as RDF and OWL repositories, mainly because of the conception of them being intuitive for human. In the previous work, we developed FREyA, an interactive Natural Language Interface for querying ontologies. It uses syntactic parsing in combination with the ontology-based lookup in order to interpret the question, and involves the user if necessary. The user's choices are used for training the system in order to improve its performance over time. In this paper, we discuss the suitability of FREyA to query the Linked Open Data. We report its performance in terms of precision and recall using the MusicBrainz and DBpedia datasets.

Keywords: Natural language interfaces, ontologies, question-answering, learning, clarification dialogs.

1 Introduction

With the rapid growth of the Linked Open Data (LOD) cloud[1] the effective exploitation becomes an issue largely because of the complexity and syntactic unfamiliarity of the underlying triple models and the query languages built on top of them. Natural Language Interface (NLI) systems become increasingly relevant for information systems fronting rich structured data stores such as RDF and OWL repositories, mainly because of the conception of them being intuitive for human.

According to [8], a major challenge when building NLIs is to provide the information the system needs to bridge the gap between the way *the user* thinks about the domain of discourse and the way *the domain knowledge is structured* for computer processing. This implies that in the context of NLIs to ontologies, it is very important to consider the ontology structure and content. Two ontologies describing identical domains (e.g., music) can use different modelling conventions. For example, while one ontology can use a datatype property *artistName* of class *Artist*, the other one might use instances of a special class to model

[1] http://linkeddata.org

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 125–138, 2012.

the artist's name[2]. A *portable* NLI system would have to support both types of conventions without sacrificing performance. *Portable* NLIs are those that can be adapted easily to new domains (or new ontologies covering the same domains). Constructing such systems poses a number of technical and theoretical problems because many of the techniques developed for specialised systems preclude automatic adaptation to new domains [8].

Ontologies can be constructed to include sufficient lexical information to support a domain-independent query analysis engine. However, due to different processes used to generate ontologies, the lexicon might be of varying quality. In addition, some words might have different meanings in two different domains or context. For example, *How big* might refer to *height*, but also to *length*, *area*, or *population* – depending on the question context, but also on the ontology structure. This kind of adjustments – or mappings from words or phrases to ontology concepts/relations, is usually performed during the *customisation* of NLIs.

Many NLIs for querying ontologies have been developed in recent years. Challenges related to Natural Language understanding such as *ambiguity* and *expressiveness* are balanced by constraining the supported language, e.g. by using a Controlled Natural Language, such as in AquaLog [14], or ORAKEL [1]. While NLI systems with a good performance require customisation (such as in the case of ORAKEL), several systems have been developed for which the customisation is not mandatory (e.g., AquaLog, PANTO [17], Querix [10], NLP-Reduce [10], QuestIO [5]). However, as reported in [14] the customisation usually improves recall. On the other hand, the complexity of supported questions differs from one system to another. While systems such as NLP-Reduce or QuestIO process queries without deep grammar analysis, the other systems (such as ORAKEL) support compositional semantic constructions such as quantification and negation.

With regards to *portability*, most of these systems are tested in the *closed-domain* scenario with ontologies which cover different, but narrow domains, with the exception of PowerAqua [15], the system that evolved from AquaLog aiming to serve as a Question-Answering system for the Semantic Web. PowerAqua was evaluated in the *open-domain* scenario [12] (e.g. through querying the ontologies indexed by Watson [6]). Portability of the majority of other NLIs to ontologies is tested by demonstrating that all that is required to port the system is the ontology URI – the system automatically generates the *domain lexicon* by reading and processing ontology lexicalisations.

With the availability of Linked Open Data, *portability* gained a new dimension bringing up the open-domain scenario where the context is multiple domains/ontologies on the contrary to the previously considered closed-domain. Having more than one ontology describing exactly the same domain, or hundreds of domains in one huge dataset requires support for *heterogeneity, redundancy* and *incompleteness* which comes with this multi-billion dataset. In other words, the system now needs to deal not only with how to map certain terms to the ontology concepts but it also needs to disambiguate and decide

[2] See for example how class *Alias* is used in the Proton System Module ontology: `http://proton.semanticweb.org/`

which ontology should provide the best answer (should *Where* be mapped to *http://purl.org/dc/terms/Location*, *http://dbpedia.org/ontology/locationCity* or any other). On the other hand, availability of such enormous knowledge base gives the possibility to merge experience which has been collected for decades by researching open-domain Question-Answering systems, NLIs to databases, and dialog systems in order to successfully accomplish what has been a great challenge for such a long time: answering questions automatically using the distributed sources on the Web. This has not been possible with databases as they are distributed and not interoperable, while Question-Answering systems use methods from Information Retrieval to locate documents in which the answer may appear. Information Retrieval methods although scale well, do not often capture enough semantics - relevant documents could be easily disregarded if the answer is hidden in a form which is not in-line with the expected patterns.

In this paper, we discuss requirements and suitability for querying the Linked Open Data using the system called FREyA. FREyA is named after **F**eedback, **R**efinement and **E**xtended Vocabular**Y** **A**ggregation [4], as it aims to investigate whether *user interaction* coupled with deeper syntactic analysis and usability methods such as *feedback* and *clarification dialogs* can be used in combination to improve the performance of NLIs to ontologies. FREyA has previously shown a good performance (recall and precision reaching 92.4% [4]) on the Mooney GeoQuery dataset which is extensively used for the evaluation of NLIs in recent years. We report the performance of FREyA using the MusicBrainz and DBpedia datasets provided within the QALD-1 challenge[3] and discuss how we begin to address the problem of querying the linked data in the open-domain scenario.

2 FREyA

FREyA is an interactive Natural Language Interface for querying ontologies which combines usability enhancement methods such as *feedback* and *clarification dialogs* in an attempt to: 1) **improve recall** by enriching the domain lexicon from the user's vocabulary (see [2]) 2) **improve precision** by resolving *ambiguities* more effectively through the dialog. The suggestions shown to the user are found through ontology reasoning and are initially ranked using the combination of string similarity and synonym detection (using WordNet[7]). The system then learns from the user's selections, and improves its performance over time. In what follows we give a brief overview of FREyA, followed by the requirements for using the system with different datasets and challenges raised by using it with the linked data.

Figure 1 shows the workflow starting with a Natural Language (NL) question (or its fragment), and ending when the answer is found. **The syntactic parsing and analysis** generates a parse tree using Stanford Parser [11] and then uses several heuristic rules in order to identify *Potential Ontology Concepts (POCs)*. POCs refer to question terms/phrases which can but not necessarily have to be linked to Ontology Concepts (OCs). POCs are chosen based on the

[3] `http://www.sc.cit-ec.uni-bielefeld.de/qald-1`

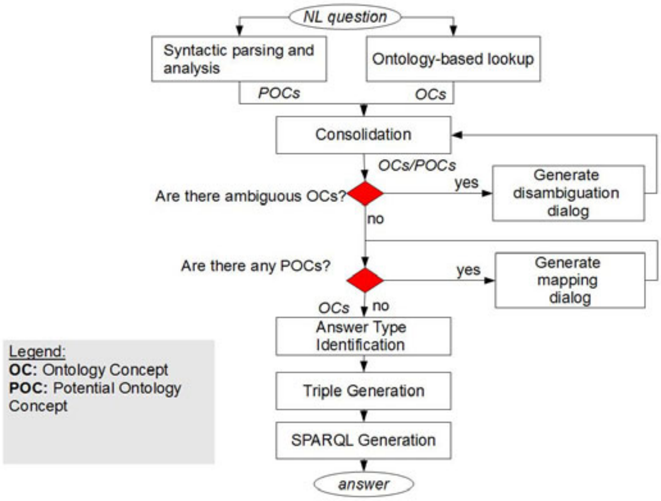


Fig. 1. FREyA Workflow

analysis of the syntactic parse tree, however this analysis does not require strict adherence to syntax and works on ill-formed questions and question fragments as well as on the grammatically correct ones. For example, nouns, verbs, or WH-phrases such as *Where*, *Who*, *When*, *How many* are expected to be found by our **POC identification algorithm**. This algorithm is based on the identification of prepreterminals and preterminals in the parsed tree, and also on their part-of-speech tags (see [4]).

The ontology-based lookup links question terms to logical forms in the ontology which we call Ontology Concepts (OCs) without considering any context or grammar used in the question (apart from morphological analysis, see [5]). Ontology Concepts refer to *instances/individuals*, *classes*, *properties*, or *datatype property values* such as string literals. By default, the system assumes that *rdfs:label* property is used to name the specific Ontology Concept. However, for ontologies which use different naming conventions (such as using *dc:title* inside the MusicBrainz dataset), it is possible to predefine which properties are used for names. This will enable the system to make the distinction between making a *datatype property value element* and an *instance element*. This distinction is important as different *elements* are used differently during the *Triple Generation* and *SPARQL generation* steps.

The consolidation algorithm aims at mapping existing POCs to OCs automatically. If it fails, the user will be engaged in the dialog. In the query *Give me all former members of the Berliner Philharmoniker.*, the **POC identification algorithm** will find that *the Berliner Philharmoniker* is a POC, while the **ontology-based lookup** will find that *Berliner Philharmoniker* is an OC, referring to an instance of *mm:Artist*. As the only difference in the POC and the OC text is a determiner (*the*), the consolidation algorithm will resolve this POC automatically by removing it, and by verifying that this noun phrase refers to the OC with *dc:title Berliner Philharmoniker*.

When the system fails to automatically generate the answer (or when it is configured to work in the *forceDialog* mode, see Section 2.1) it will prompt the user with a dialog. There are two kinds of dialogs in FREyA. The **disambiguation dialog** involves the user to resolve identified ambiguities. The **mapping dialog** involves the user to map a POC to the one of the suggested OCs. While the two types of dialogs look identical from the user's point of view, there are differences which we will highlight here. Firstly, we give a higher priority to the disambiguation dialog in comparison to the mapping dialog. This is because our assumption is that the question terms which exist in the graph (OCs) should be interpreted before those which do not (POCs). Note that FREyA does not attempt to interpret the whole question at once, but it does it for one pair of OCs at the time. In other words, one resolved dialog can be seen as a pair of two OCs: an OC to which a question term is mapped, and the neighbouring OC (context). Secondly, the way the suggestions are generated for the two types of dialogs differ. The disambiguation dialog includes only the suggestions with Ontology Concepts that are the result of the ontology-based lookup (unless it is extended using the *forceDialog* mode, see Section 2.1). The mapping dialog, in contrast, shows the suggestions that are found through the ontology reasoning by looking at the closest Ontology Concepts to the POC (the distance is calculated by walking through the parsed tree). For the closest OC X, we identify its neighbouring concepts which are shown to the user as suggestions. *Neighbouring concepts* include the defined properties for X, and also its neighbouring classes. *Neighbouring classes* of class X are those that are defined to be 1) the domain of the property P where range(P)=X, and 2) the range of the property P where domain(P)=X. Finally, the sequence of disambiguation and mapping dialogs themselves controlled differently for these two kinds of dialogs:

- *The disambiguation dialogs* are driven by the question *focus* or the *answer type*, whichever is available first: the closer the OC to be disambiguated to the question focus/answer type, the higher the chance that it will be disambiguated before any other. The question focus is the term/phrase which identifies *what the question is about*, while the answer type identifies the type of the question such as *Person* in the query *Who owns the biggest department store in England?*. The focus of this question would be *the biggest department store* (details of the algorithm for identifying the focus and the answer type are described in [3]). After all ambiguities are resolved the FREyA workflow continues to resolve all POCs through the mapping dialogs.
- *The mapping dialogs* are driven by the availability of the OCs in the neighbourhood. We calculate the distance between each POC and the nearest OC inside the parsed tree, and the one with the minimum distance is the one to be used for the dialog before any other.

After all OCs are disambiguated and no POCs remain to be resolved, the system proceeds to finding the answer. First, it identifies the *answer type*, and then combines OCs into triples, which are then used to generate the SPARQL query. Unlike other approaches which start by identifying the question type followed by the identification of the answer type, our approach tries to interpret the majority of the question before it identifies the answer type. The reason for this is that in FREyA there is no strict adherence to syntax, and the approach heavily relies on the ontology-based lookup and the definitions in the RDF structure. Hence, it can only identify the answer type after all relevant mappings and disambiguations are performed. Note however, that there are cases when the answer type is identified before the whole question is interpreted, and in this case it is used to drive the remaining mappings, if any (as described above).

An important part of FREyA is its **learning mechanism**. Our goal is to *learn the ranking* of the suggestions shown to the user so that after sufficient training the system can automatically generate the answer by selecting the best ranked options. In order to make the model as generic as possible, we do not update our learning model per question, but per combination of a POC/OC and the neighbouring OC (context). We also preserve a function over the selected suggestion such as minimum, maximum, or sum (applicable to datatype property values). This way we may extract several learning rules from a single question, so that if the same POC/OC appears in the same context, we can reuse it. The algorithm has previously shown a good performance on the Mooney GeoQuery dataset improving the initial suggestion rankings by 6% on a random sample of 103 questions (see [4]).

An Example Figure 2 shows the syntax tree for the query *what is the population of New York*. As *New York* is identified as referring to both *geo:State* and *geo:City*, we first ask the user to disambiguate (see Figure 2 a.)). If he selects for example *geo:City*, we start iterating through the list of remaining POCs. The next one (*population*) is used, together with the closest OC *geo:City*, to generate suggestions for the mapping dialog. Among them there will be *geo:cityPopulation* and after the user select this from the list of available options, *population* is mapped to the datatype property *geo:cityPopulation* (see Figure 2 b.)). Note that if the user selected that *New York* refers to *geo:State*, suggestions would be different, and following his selection, *population* would probably be mapped to refer to *geo:statePopulation* as the closest OC would be *geo:State*.

2.1 Querying Linked Data with FREyA

FREyA can be easily ported to work with a different ontology, or a set of ontologies. It can either preload the ontologies into its own repository which is based on OWLIM[4], or connect to an already existing repository, which can be local or remote.

In order to perform the ontology-based lookup at the query processing time, FREyA requires extracting ontology lexicalisations, processing them, and adding them to an index. The extraction of ontology lexicalisations requires reading the whole repository through a set of SPARQL queries. The number of SPARQL

[4] http://ontotext.com/owlim

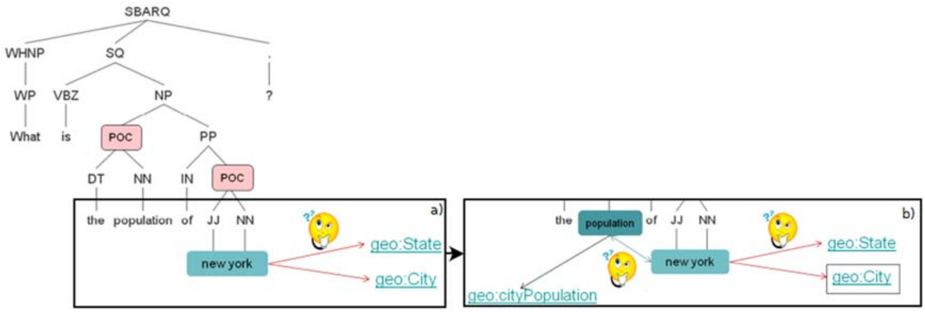


Fig. 2. Validation of potential ontology concepts through the user interaction

queries depends on the size of the schema which describes the dataset. Nowadays, the data are distributed over various types of servers, which often allow access through SPARQL endpoints. However, depending on the repository which is used underneath, some SPARQL queries can be highly unoptimised and slow. Alternative solution is to use services such as Watson [6] or Sindice [16] which index ontologies on the Web, in order to remove the burden of the system initialisation which can be large for large datasets. However, the downside of this approach is the lack of control over these services. As pointed out in [12], the resources on the open Web that can be accessed through Watson seem to have quality issues: there are many redundant, noisy and incomplete data (for example, the schema could be missing or ontologies might not be populated). These problems are partially addressed by approaches for assessing the quality through tracking the provenance (e.g.,[9]), however, much more needs to be done in the years to come in order to use and query the Web of Data effectively.

Most of the recently developed NLIs to ontologies (including our own QuestIO [5]) are built with the assumption that *ontologies are perfect*:

- each concept/relation in the ontology has the human lexicalisation which describes it – not necessarily a definition, but rather a term which a human would use to refer to this concept/relation;
- each concept/relation is positioned carefully in the taxonomy: super-concepts and super-relations are more generic, and sub-concepts and sub-relations are more specific.

The availability of Linked Open Data changed this assumption as encouraging people to publish their data resulted in the large amount of RDF graphs being made available and interlinked with each other. None of these are perfect – lexicalisations do exist, but not often they reflect “a term which a human would use to refer to a concept”. In addition, the flat structure is dominant. One of the reasons for this is scalability: tractable reasoners do not scale well if the structure of the ontology is complex. The assumptions based on which NLIs to ontologies have been developed had to adapt to the new challenges, the main one being that *ontologies are not perfect*, and that tools which work with them must take this into consideration. In addition, the *scale* becomes an issue, and also *incompleteness, heterogeneity, and noise* inherent in these data. A huge number of ontologies interlinked with each other means a high probability that there is a redundant information, which needs to be filtered out by the systems querying these data.

FREyA does not require a strict adherence to syntax, however, it relies on the ontology-based lookup. Trying a sample query *What is the capital of France?* with FREyA initialised with a superset of DBpedia (accessed through `http://www.factforge.net/sparql` repository) revealed that according to the extracted lexicon, **each word in the question refers to at least one Ontology Concept**. If there were no automatic disambiguation nor heavy grammar analysis, the system would model the first dialog asking: *What is 'what'? Is 'what' related to: LIST OF URIs*. A similar dialog would be modelled for 'is': the system would ask the user whether *is* is related to: *be*, *was*, or *were*. And so on, for each word in the question. These situations must be resolved either by performing automatic disambiguation (which might be expensive for datasets with billions of triples) or by constraining the supported language and allowing the user to type in only a limited set of question types. In case of the system failing to automatically interpret the question, it can prompt the user with the dialog as is the case with FREyA. The fine balance is in the combination of these approaches: disambiguate as much as possible and use the ranking mechanisms (e.g., those that exist in FREyA, or any other methods for effective ranking such as [13]), and correct them if necessary using the interactive features of FREyA.

When trying any dataset with FREyA for the first time, it is advisable to use the dialog as much as possible in order to check the system interpretations and correct them if necessary. In that regard, there are several *modes* that can be used, among which the most important are:

- **Automatic mode.** The system will simulate selection of the best ranked option(s) for each attempt to map a question term to an OC. This mode is used when the confidence is high that the ranking is effective, or the system has been trained enough and can make the decisions correctly. For the example previously described in Figure 2, the automatic mode would return both *statePopulation of new york state* and *cityPopulation of new york city* in the results initially as the initial ranking would assign the equal score to both*new york city* and *new york state*.
- **ForceDialog mode** generates the dialog for each attempt to map a question term to an OC.

The mode can be changed easily and without the need to reinitialise the system hence if the user uses FREyA in the automatic mode and discovers non-satisfying results, he can immediately switch to the *force dialog mode* in order to investigate the mappings. His input will then improve the system for the next user. Note that for the true ambiguities the automatic mode might not be the best choice even in the perfectly trained system. For instance, if somebody asks about *How big is new york state?* we might be unable to decide whether *How big* refers to *state area* or *state population* automatically. In this situation, as the system learns from the user's suggestions, the automatic mode would work in favour of majority of the users. However, if the majority of users refer to *state area* when talking about size, the minority still have chance to get the correct answer by using FREyA in the *forceDialog* mode and mapping *big* to *state population*.

3 Evaluation

In this section we report the performance of FREyA using the MusicBrainz and DBpedia datasets provided within the QALD-1 challenge. We preloaded the data into our local repository (BigOWLIM 3.4, on the top of Sesame[5]) and then initialised the system using the SPARQL queries. Another option was to connect to the SPARQL endpoint provided by the QALD-1 challenge organisers[6], however, this was a difficult path due to the limited server timeout, which was not sufficient for executing all required queries.

Generating the index which is required for performing the ontology-based lookup is a mandatory step but is done once per dataset, although it might be time-consuming depending on the size of the data. Table 1 shows the statistics of loading the two datasets into the OWLIM repository and generating the index.

Table 1. Initialisation of the system and the size of datasets

	MusicBrainz	**DBpedia**
#explicit statements	14 926 841	328 318 709
#statements	19 202 664	372 110 845
#entities	5 490 237	96 515 478
#SPARQL queries executed	30	361623
initialisation time	1380s (0.38h)	182779s (50.77h)

After the index is generated, it is used at the query execution time. We first ran 50 training queries for both datasets and measured the overall precision, recall and f-measure. We then repeat the process with 50 test questions for each dataset. This experiment was conducted with FREyA in the *forceDialog* mode. Results are shown in Table 2[7]. MusicBrainz was a challenging dataset due to the existence of properties *beginDate* and *endDate*, which do not have any domain defined, and moreover, which are used extensively throughout the ontology and especially in the combination with the blank nodes. Several failures were due to the misfunction of the *Triple Generator* when these two properties were mapped to the wrong entity. For example, *Since when is Tom Araya a member of Slayer?* resulted in generating the following triples:

```
?joker1 - beginDate - Tom Araya (Artist)
Tom Araya (Artist) - member of band - ?joker2
?joker2 - toArtist - Slayer (Artist)
```

[5] `http://openrdf.org`

[6] `http://greententacle.techfak.uni-bielefeld.de:5171/sparql`

[7] Demos showing FREyA answering the QALD-1 challenge questions are available from `http://gate.ac.uk/sale/dd/`

Table 2. Performance of FREyA using QALD-1 datasets: the left figures exclude while the right figures include the questions correctly answered after reformulation. The number of dialogs per question includes only the questions that could be answered *correctly* with or without reformulation. *Not supported* questions include those that could not be correctly mapped to the correct SPARQL query due to the limited language coverage. For example, questions requiring negation, temporal reasoning such as *Which bands were founded in 2010?* or quantification such as in *Which locations have more than two caves?*. Partially correct questions are those that have returned a portion or a superset of the correct results.

	MusicBrainz		**DBpedia**	
	Training	*Testing*	*Training*	*Testing*
Precision	0.75/0.77	0.66/0.8	0.74/0.85	0.49/0.63
Recall	0.66/0.68	0.54/0.66	0.58/0.66	0.42/0.54
F-measure	0.70/0.74	0.59/0.71	0.67/0.72	0.45/0.58
# questions not supported	6	9	11	7
# reformulated questions	1	6	4	6
avg.#dialogs per question	3.4	3.65	2.7	2.85
# partially correct questions	1	1	3	12

and the corresponding SPARQL resulted in retrieving the birthday of Tom Araya, and not the date when he joined the group which is the correct answer.

Other challenges related to the ontology design in MusicBrainz include existence of the property *trackList* which has a container of type *rdf:Seq* as range. In addition, the statements with *releaseType* property use subclasses of class *Type* and not instances of that class which caused several failures. For example, the question *Who is the creator of the audiobook the Hobbit?* requires retrieving instances with lexicalisation *the Hobbit*, which are at the same time related to the class *TypeAudiobook* using the *releaseType* property, while FREyA expects that they are related using the *rdf:type* relation.

The main challenge with DBpedia was a selection of the property to use, due to the large number of suggestions that have always been present. For example, *Who created English Wikipedia?* could be mapped to *?joker dbp:created dbpedia:English_Wikipedia* while the correct answer is returned only after using *dbo:author* relation, instead of *dbp:created*[8]. In addition, there are many quality issues such as in the question *Who designed the Brooklyn Bridge?* where *designed* was mapped to *dbp:architect* instead of *dbp:designer* which resulted in retrieving `http://dbpedia.org/resource/John_Augustus_Roebling`, while using *dbp:designer* the result is `http://dbpedia.org/page/John_A._Roebling`. However, as no mapping exist between the two URIs, the former URI is not the same as the latter, and hence this is marked as an incorrect answer. Interestingly, the former URL is redirected to the latter, which indicates that the two URIs should also be connected using the property *sameAs* in the dataset.

[8] We use *dbp for* `http://dbpedia.org/property` and *dbo* for `http://dbpedia.org/ontology` namespaces.

Another challenge specific to DBpedia was the lack of the domain and range classes for properties. Therefore, some questions could not be correctly mapped to the underlying Ontology Concepts. In some cases, the reformulation of queries could help (such as using *spouse* instead of *married to*). However, reformulation was not always sufficient. For example, in *Which states border Utah?*, *border* needs to be mapped to the eight properties: *dbp:north*, *dbp:south*, *dbp:east*, *dbp:west*, *dbp:northwest*, *dbp:northeast*, *dbp:southwest*, and *dbp:southeast.* As none of these have any domain or range, they did not appear in the suggestions and hence the only way to answer the question using FREyA is to ask eight questions such as *Which states are north of Utah?*, *Which states are south of Utah*, and so on for each property. It is interesting to observe that 12 incorrectly answered questions using the DBpedia test questions were indeed partially correct. The correct mappings could only be placed if we were more familiar with the knowledge structure inherent in the dataset. This also explains the difference in the performance of FREyA using the training and the testing set of DBpedia.

Failures that were common for both datasets are related to the equal treatment of the datatype property values. For example, the question *How many jazz compilations are there?* failed to be answered correctly due to FREyA finding all compilations that had the user defined tag ‘jazz’ which is case insensitive (using FILTER REGEX(str(?var), “`^jazz$`”,“i”). Therefore, it included also ‘Jazz’ which lead to the incorrect answer. On the other hand, some entries were missed when the fuzzy matching was necessary such as in *Which companies are in the computer software industry?* that requires finding not only companies with the property *industry* ‘computer software’ but also ‘computer hardware, software’, ‘computer software and engineering’, and the like. At the moment, the datatype property values in FREyA are supported by including the exact match (case insensitive) only. In future, we might extend our approach to support more sophisticated treatment of strings so that the treatment differs depending on the context.

Several reformulations for both datasets resulted in a significant increase of the precision and recall, e.g. adding quotes such as in *Which artists performed the song “Over the Rainbow?”*. Without quotes, *Over* was parsed as a preposition, and the whole question failed to be answered, while with quotes this was a part of the Noun Phrase which lead to the correctly answered question.

Learning. To measure the effect of the learning mechanism, we run the experiment in two iterations: we first answered 50 testing questions using an empty learning model and then using the system trained with 50 training questions. Results are shown in Table 3.

The learning mechanism improved the overall ranking of suggestions for 0.05 for MusicBrainz, and only 0.02 for DBpedia. The reason is the size of the datasets and the relatively small number of the training questions. However, improvement of 0.02 is still an achievement considering that DBpedia has almost 100 million entities.

Table 3. Mean Reciprocal Rank for the testing set with and without learning

	MusicBrainz		**DBpedia**	
	untrained system	*trained system*	*untrained system*	*trained system*
MRR	0.63	0.68	0.52	0.54

Execution time for queries that could be answered correctly fluctuates based on the complexity of questions (e.g. number of the required dialogs). This is due to our on fly mechanism for finding suggestions which requires executing a large number of SPARQL queries in order to generate a dialog. Long execution is also affected by the complexity of the final generated SPARQL which is used to retrieve the answer. For example, queries which include FILTER statements over literal strings such as FILTER (regex(?var, “`^jazz$`”, “i”)) currently can take more than ten minutes to be executed[9]. The size of the dataset influences the execution time as well. For MusicBrainz, the average time per dialog was in the range from 0.073 to 11.4 seconds, or 8.5 seconds on average per question. For DBpedia, the execution time was much longer: from 5 to 232 seconds per dialog, and 36 seconds on average per question. This is quite slow, however, it can be optimised (e.g. by using the caching mechanisms for suggestions).

4 Conclusion and Future Work

We discussed the requirements and suitability of the Natural Language Interface – FREyA, to be used with different ontologies and for querying the Linked Open Data. The evaluation using the DBpedia and MusicBrainz testing datasets leads to the f-measure of 0.58 and 0.71 respectively which favourably compares to the other tested systems that participated in the QALD-1 challenge (PowerAqua 0.5 using DBpedia, SWIP 0.66 using MusicBrainz). More importantly, FREyA was the only system that is tested with both MusicBrainz and DBpedia datasets which demonstrates the portability. The learning mechanism improved the results for 5% and 2% for the MusicBrainz and DBpedia datasets respectively.

Acknowledgments. Authors would like to thank Ivan Peikov from Ontotext who helped with the configuration of OWLIM which was necessary for performing the experiments reported in this paper. This research is partially supported by the EU-funded LarKC (FP7-215535) project.

References

1. Cimiano, P., Haase, P., Heizmann, J.: Porting natural language interfaces between domains: an experimental user study with the orakel system. In: IUI 2007: Proceedings of the 12th International Conference on Intelligent User Interfaces, pp. 180–189. ACM, New York (2007)

[9] Experiments are run using the CentOS 5.2 Linux virtual machine running on a AMD Opteron 2431 2.40GHz CPU with 2 cores and 20G RAM.

2. Damljanovic, D.: Towards Portable Controlled Natural Languages for Querying Ontologies. In: Rosner, M., Fuchs, N. (eds.) Second Workshop on Controlled Natural Language. CEUR Workshop Pre-Proceedings, Marettimo Island, Italy, September 13-15, vol. 622 (2010) ISSN 1613-0073, http://ceur-ws.org
3. Damljanovic, D., Agatonovic, M., Cunningham, H.: Identification of the Question Focus: Combining Syntactic Analysis and Ontology-based Lookup through the User Interaction. In: 7th Language Resources and Evaluation Conference (LREC), ELRA, La Valletta, Malta (May 2010)
4. Damljanovic, D., Agatonovic, M., Cunningham, H.: Natural Language Interfaces to Ontologies: Combining Syntactic Analysis and Ontology-based Lookup through the User Interaction. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010. LNCS, vol. 6088, pp. 106–120. Springer, Heidelberg (2010)
5. Damljanovic, D., Tablan, V., Bontcheva, K.: A text-based query interface to owl ontologies. In: 6th Language Resources and Evaluation Conference (LREC). ELRA, Marrakech (2008)
6. d'Aquin, M., Baldassarre, C., Gridinoc, L., Angeletou, S., Sabou, M., Motta, E.: Watson: A Gateway for Next Generation Semantic Web Applications. In: Poster, ISWC 2007 (2007), http://iswc2007.semanticweb.org/papers/Paper366-Watson-poster-ISWC07.pdf
7. Fellbaum, C. (ed.): WordNet - An Electronic Lexical Database. MIT Press (1998)
8. Grosz, B.J., Appelt, D.E., Martin, P.A., Pereira, F.C.N.: TEAM: An experiment in the design of transportable natural-language interfaces. Artificial Intelligence 32(2), 173–243 (1987)
9. Hartig, O., Zhao, J.: Using Web Data Provenance for Quality Assessment. In: Proceedings of the First International Workshop on the Role of Semantic Web in Provenance Management (SWPM 2009) at the International Semantic Web Conference (ISWC 2009), Washington D.C., USA (2009)
10. Kaufmann, E., Bernstein, A., Fischer, L.: NLP-Reduce: A Naive but Domain-independent Natural Language Interface for Querying Ontologies. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 1–2. Springer, Heidelberg (2007)
11. Klein, D., Manning, C.D.: Fast exact inference with a factored model for natural language parsing. In: Becker, S., Thrun, S., Obermayer, K. (eds.) Advances in Neural Information Processing Systems, NIPS 2002, vol. 15, pp. 3–10. MIT Press (2002), http://books.nips.cc/papers/files/nips15/CS01.pdf
12. Lopez, V., Fernández, M., Motta, E., Stieler, N.: PowerAqua: Supporting Users in Querying and Exploring the Semantic Web Content. Semantic Web Journal (2011)
13. Lopez, V., Nikolov, A., Fernandez, M., Sabou, M., Uren, V., Motta, E.: Merging and Ranking Answers in the Semantic Web: The Wisdom of Crowds. In: Gómez-Pérez, A., Yu, Y., Ding, Y. (eds.) ASWC 2009. LNCS, vol. 5926, pp. 135–152. Springer, Heidelberg (2009), http://dx.doi.org/10.1007/978-3-642-10871-6_10
14. Lopez, V., Uren, V., Motta, E., Pasin, M.: Aqualog: An ontology-driven question answering system for organizational semantic intranets. Web Semantics: Science, Services and Agents on the World Wide Web 5(2), 72–105 (2007)
15. Lopez, V., Uren, V.S., Sabou, M., Motta, E.: Cross Ontology Query Answering on the semantic Web: an Initial Evaluation. In: Gil, Y., Noy, N.F. (eds.) K-CAP, pp. 17–24. ACM (2009)

16. Tummarello, G., Delbru, R., Oren, E.: Sindice.com: Weaving the Open Linked Data. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 552–565. Springer, Heidelberg (2007)
17. Wang, C., Xiong, M., Zhou, Q., Yu, Y.: Panto: A Portable Natural Language Interface to Ontologies. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 473–487. Springer, Heidelberg (2007)

Ontology-Based User-Defined Rules and Context-Aware Service Composition System

Victoria Beltran[1], Knarig Arabshian[2], and Henning Schulzrinne[3]

[1] Dept of Telematics, Universitat Politècnica de Catalonia/Fundació I2Cat, Barcelona, Spain
[2] Alcatel-Lucent Bell Labs, New Jersey, USA
[3] Dept of Computer Science, Columbia University, New York, USA

Abstract. The World Wide Web is becoming increasingly personalized as users provide more of their information on the Web. Thus, Web service functionality is becoming reliant on user profile information and context in order to provide user-specific data. In this paper, we discuss enhancements to SECE (Sense Everything, Control Everything), a platform for context-aware service composition based on user-defined rules. We have enhanced SECE to interpret ontology descriptions of services. With this enhancement, SECE can now create user-defined rules based on the ontology description of the service and interoperate within any service domain that has an ontology description. Additionally, it can use an ontology-based service discovery system like GloServ as its service discovery back-end in order to issue more complex queries for service discovery and composition. This paper discusses the design and implementation of these improvements.

Keywords: context-aware systems, ontologies, semantic web, rule-based systems, service discovery, service composition, web services.

1 Introduction

In recent years, the World Wide Web has been advancing towards greater personalization. Services on the Web such as, social networking, e-commerce or search sites, store user information in order to profile the user and target specific products or ads of interest. Since web service functionality is increasingly relying on user information, a user's context is becoming more crucial towards creating a personalized set of services within the Web.

As these types of services proliferate, a framework is needed where multiple services can be discovered and composed for a particular user within a certain context. With this in mind, we have developed SECE (Sense Everything, Control Everything), a platform for context-aware service composition based on user-defined rules. The contributions to SECE are two-fold: a user-friendly rule language and the design and implementation of a context-aware service composition framework.

SECE differs from other rule-based systems in that it provides an interface for creating rules in natural English-like language commands. The main drawback

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 139–155, 2012.

of rule-based systems is that the rule languages involve complex formulaic or XML descriptions. Lay people are not as inclined to use these systems as the learning curve for these languages may be steep. Thus, we have defined a formal rule language which resembles English. With a simplified English-like interface to creating rules, users will be more prone to incorporate rule-based systems into their lives, making context-aware computing a seamless part of everyday life.

Additionally, SECE provides a platform for context-aware service composition for a number of services, such as, presence, telecommunication, sensors and location-aware services. Users can subscribe to various services by formulating simple rules that create a composition of services. The rules trigger event-based service discovery and composition depending on the user's context, such as her location, time, and communication requests. Traditional rule-based systems are mostly designed to handle a single service domain. SECE, on the other hand, interacts with a few service domains. For more information on the SECE architecture and rule language, we encourage the readers to refer to the following paper [1].

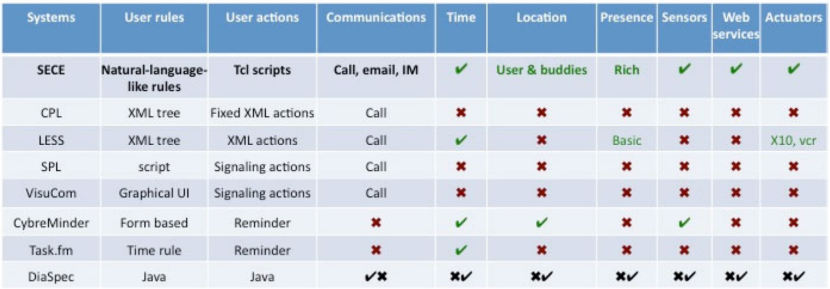
In this paper, we discuss enhancements to both aspects of SECE: its rule language and back-end architecture. Whereas previously SECE had a hard-coded rule language for a limited number of event-based service domains, we have now improved SECE to use the Web Ontology Language (OWL) description of a service domain to dynamically create a rule language for that service domain. Additionally, SECE's architectural platform has been modified to integrate with a back-end ontology-based global service discovery system, GloServ, to access any type of service domain within the GloServ directory [2] [3]. GloServ classifies services in an ontology and provides ontology descriptions of different service domains. It also has an ontology-based query interface for service discovery and composition.

With these improvements, SECE can now be generalized to include all types of service domains, described in an ontology, as well as issue more complex ontology-based queries for service discovery and composition. Having the ability to adapt a rule language to new service domains makes SECE into a powerful front-end context-aware system. Additionally, by using GloServ as its back-end, SECE can now interoperate with any type of service that has an OWL description, broadening its scope drastically. We envision that SECE will enable services to seamlessly integrate into people's lives. A person can now create rules with ease and be notified of services at the right time and place. This will create a profound impact in how people interact with services. There will now be a closer connection between a person and services available, establishing a personalized network of services.

The organization of this paper is as follows: Section 2 describes current work in the field of context-aware computing and service composition; Section 3 gives an overview of the original SECE architecture and functionality; we discuss the enhancements to SECE and its implementation in Section 4; Section 5 discusses future work; finally, Section 6 summarizes the main contributions of this paper.

2 Related Work

Several solutions for user created services have been proposed; some of these solutions are compared to SECE in Figure 1. The second column indicates the user language for defining events and conditions that trigger action scripts. The third column indicates the language for action scripts. The fourth column shows the kinds of communication services that the users can use. The following columns show the types of information handled by the systems. CPL [4], LESS [5], SPL [6], VisuCom [7] and DiaSpec [8] are attempts to allow end users to create services, but they are all limited to controlling call routing. Also, CPL and LESS use XML and, hence, even simple services require long programs. Moreover, XML-based languages are difficult to read and write for non-technical end-users. DiaSpec is very low level. Writing a specification in DiaSpec and then developing a service using the generated framework is definitely not suitable for non-technical end users. The authors of DiaSpec extended [9] their initial work to support services beyond telephony, which include sensors and actuators. However, it is still only suitable for advanced developers. SPL is a scripting language which is suitable for end-users but only for telephony events. VisuCom has the same functionality as SPL, but allows users to create services visually via GUI components.



Systems	User rules	User actions	Communications	Time	Location	Presence	Sensors	Web services	Actuators
SECE	**Natural-language-like rules**	**Tcl scripts**	**Call, email, IM**	✓	User & buddies	Rich	✓	✓	✓
CPL	XML tree	Fixed XML actions	Call	✗	✗	✗	✗	✗	✗
LESS	XML tree	XML actions	Call	✓	✗	Basic	✗	✗	X10, vcr
SPL	script	Signaling actions	Call	✗	✗	✗	✗	✗	✗
VisuCom	Graphical UI	Signaling actions	Call	✗	✗	✗	✗	✗	✗
CybreMinder	Form based	Reminder	✗	✓	✓	✗	✓	✗	✗
Task.fm	Time rule	Reminder	✗	✓	✗	✗	✗	✗	✗
DiaSpec	Java	Java	✓✗	✗✓	✗✓	✗✓	✗✓	✗✓	✗✓

Fig. 1. Comparison to related work

CybreMinder [10] is a context-aware tool which allows users to setup email, SMS, print out and on-screen reminders based not only on time but also location and presence status of other users. It uses local sensors to detect a user's location. It does not take any actions, but rather displays reminders to the end user. Also it is not as powerful as scripting-based systems due to its form-based nature. Task.fm [11] is a similar SMS and email remainder system which uses natural language to describe time instants when email or SMS reminders will be sent. However, Task.fm only supports time-based rules and does not include information from sensors. This tool does not take actions other than reminding users via SMS, email or phone call.

Regarding composition of web services, SWORD [12] was one of the first prototypes. However, this offers a quite limited composition that is not automatic and its scripting language is targeted at developers. Ezweb [13] is a graphical tool by which users can connect web services manually. However, this does not provide automatic web service discovery or a language for composing services. Moreover, service composition is not context-aware and proactive. Yahoo Pipes [14] is other graphical tool for web service composition. However, it presents the same limitations as Ezweb and its graphical interface is not really easy-to-use and intuitive, which makes it very difficult for non-technical users. We only found a prototype described in a research paper [15] that offers event-based web service composition. This means that service composition is triggered by events, such as changes in the user's context, instead of end users. However, this work does not provide any language or tool for specifying the web service compositions and events that trigger them. The authors seem to implement low-level compositions that may be personalized according to user preferences. Thus, this does not offer end users control of service composition. Moreover, this prototype seems not to be available in the Internet.

To the best of our knowledge, there is no implemented platform for allowing end users to compose services of different kind based on events. The current solutions are not proactive because the end-user is who triggers the composite services or only provides template-based compositions (i.e., the user is not who defines the compositions). There is neither a platform for event-based web service discovery. The composition tools that take user context into account, only consider a limited set of context. The graphical interfaces of the studied tools are quite limited and not flexible for non-technical users. The scripting languages provided by some tools are neither suitable for non-technical users and only support a limited set of context information. Moreover, none of the studied tools proactively discover web services based on the user preferences.

3 SECE

SECE is a rule-based context-aware system that connects services, that may have otherwise been disconnected, to create a personalized environment of services for a user. It has two fully-integrated components: user-defined rules in a natural English-like formal language and a supporting software architecture. Users are not required to continually interact with the system in order to query for or compose a set of services. They need to only define rules of what they want to accomplish and SECE does the rest by keeping track of the user's context, as well as information from external entities such as sensors, buddies, or social events in order to notify the user about a service. It accomplishes this by communicating with several third party applications and web services such as Google services (e.g., GMail, GContacts and GCalendar), Social Media services (e.g., Facebook or Twitter), VoIP proxy servers, presence servers, sensors and actuators. Figure 2 gives an overview of the overall SECE architecture and how it interacts with its environment. We will discuss these two components of SECE in this section.

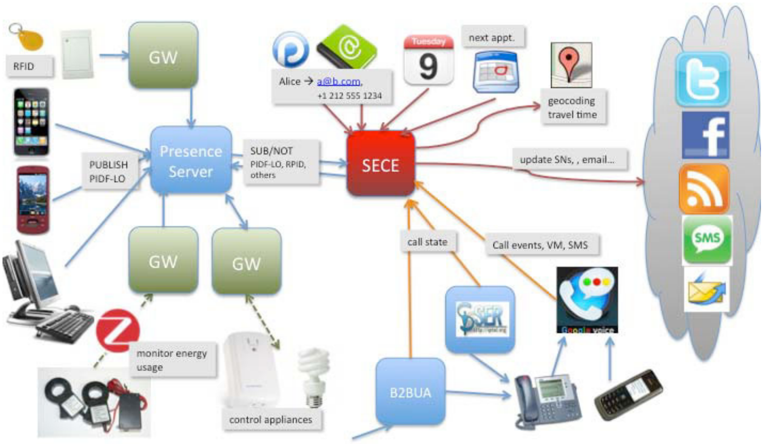


Fig. 2. SECE and its external components

3.1 SECE Architecture

As Figure 2 depicts, SECE is a web service that interacts with other web services, namely Google Services and Social Media services such as Twitter, Flickr and Facebook. The rules that are running on SECE and the rule actions that will potentially be executed determine the services with which SECE needs to interact. Thus, based on the kinds of rule that the user wishes to create and the actions that she wishes to compose, the user will need to configure the proper third-party services in her SECE account. Section 3.2 explains the SECE rules and actions, and their required services in more detail.

We are developing two services that tightly collaborate with SECE: the presence server and the VoIP proxy server. The presence server is built on the Mobicents Presence Service [16], which is compliant with SIMPLE (SIP for Instant Messaging and Presence Leveraging Extensions) [17]. It is responsible for collecting and aggregating context from different sources and sending it to SECE. It accomplishes this by receiving presence publications from context sources that contain the latest information about a user and, in turn, notifying SECE of the context changes. In the SECE framework, context sources include user devices' presence applications and gateways that control sensor networks, energy consumption and user location via RFID. To use the presence service, the end user needs to create an account from the SECE website in order to obtain the SECE presence server's access information. Thus, the user can configure the SIMPLE-compliant presence applications (e.g. SIP Communicator and Pidgin) that run on her mobile devices or desktop computers to use the SECE presence server. In the future, the presence server will interact with the home gateway for obtaining information from sensor networks and changing the state of actuators.

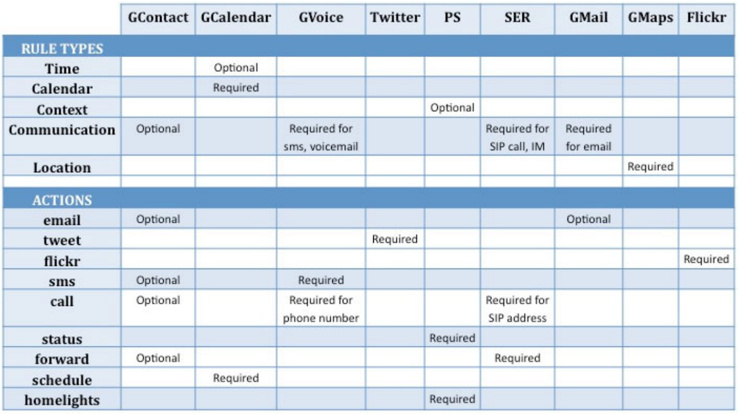
The VoIP proxy server is a SIP (Session Initiation Protocol) Express Router (SER) [18] extended to interact with SECE for handling users' SIP communications. This server and SECE implement an efficient binary protocol that lets SER inform SECE of a SIP event and lets SECE notify SER of the action to take for this event. Basically, SER informs SECE of users' incoming and outgoing calls and instant messages (IM). If an event of this kind matches a rule, the rule is triggered and, therefore, decides to either forward, reject, or modify the call by invoking an action. Then, SECE will let SER know about the action to take. As the presence service, the user needs to create a SER account through her SECE account for using the VoIP proxy service. The user also needs to set her SIP-compliant multimedia applications to use the SECE VoIP proxy server as outbound/inbound SIP proxy. A first prototype of SECE has already been developed as a web service and is being tested by members of the Internet Real Time (IRT) group at Columbia University. For a more detailed description of the SECE architecture, we refer the readers to the following paper [1].

3.2 SECE Rules

The SECE language supports five types of rules: time, calendar, location, context and communication. As a formal language, it states the valid combinations of keywords and variables for each kind of event and provides a set of commands, such as "sms", "email", "tweet" or "call". SECE rules and actions interact with different third party services based on their subscribed events and functions. Thus, SECE users need to learn about the services needed by the rule types and actions that they wish to use and configure their SECE accounts for such services. SECE will provide online documentation that gives users information about each rule's and action's syntax and required services. This documentation will also contain example rules to help users build rules for specific events and goals, and get familiarized with SECE rules. Figure 3 summarizes the required and optional services for the SECE rules and some actions.

Any SECE rule has the structure "*event* { *actions* }". *Event* defines the conditions that need to be satisfied to execute the *actions* that are delimited by braces. The SECE language for describing events is a formal language similar to English that has been designed to be easy to use and remember by end-users. This language is generated by an ANTLR grammar [19]. We use the Tcl language [20] as the syntax for the rule actions. This choice is due to Tcl's extensibility that allows adding new commands to its core with relative ease. Tcl provides a command that receives the name, arguments and code of a new command as parameters, constructs the corresponding Tcl command and incorporates it into the Tcl interpreter. Below, we describe the types of SECE rules and their involved services. In order to clearly display the structure of the rule and action language, the variables that are set by the user are highlighted in bold and the language keywords are italicized.

Time Rules: Below are two types of rules: single time events and recurrent time events. The former starts with the keyword *on* and the latter starts



	GContact	GCalendar	GVoice	Twitter	PS	SER	GMail	GMaps	Flickr
RULE TYPES									
Time		Optional							
Calendar		Required							
Context					Optional				
Communication	Optional		Required for sms, voicemail			Required for SIP call, IM	Required for email		
Location								Required	
ACTIONS									
email	Optional						Optional		
tweet				Required					
flickr									Required
sms	Optional		Required						
call	Optional		Required for phone number			Required for SIP address			
status					Required				
forward	Optional					Required			
schedule		Required							
homelights					Required				

Fig. 3. Third party services of SECE rules and some actions

with the keyword *every*. Both are fully-compliant with the Internet Calendar (ICal) standard [21]. The *on*, *until*, *except* and *including* keywords are always followed by a date expression that can have different formats (e.g., "December 31, 2011", "31st day of December, 2011" and "12/31/2011") or can be an entry in the user's GCalendar. In the first example below, the user defined an entry named "Anne's birthday" in her 2011 GCalendar.

```
on Anne's birthday, 2011 at 12:00 in Europe/Zurich {
   sms Anne "Happy Birthday!!! John";
}
every week on WE at 6:00 PM from 1/1/11 until May 10, 2011
except 3rd WE of Feb, 2011 including first day of June, 2011 {
   email irt-list "reminder: weekly meeting today at 6:00 PM";
}
```

Calendar Rules: These rules specify events that are defined in the user's GCalendar and always start with the keyword *when*. Thus, the user needs to configure his GCalendar in his SECE account before entering rules of this kind.

```
when 30 minutes before "weekly meeting" {
      email [event participants] "The weekly meeting will start in 30 minutes";
      if {{ ! my location within 3 miles of campus } {
         email [status bob.email] "I'm away" "Please, head the conference room and
         prepare everything for the weekly meeting. Not sure if I will be on time.";
      }
}
```

Location Rules: A location rule starts with the keyword *me*, if it is about the user that is entering the rule, or an identifier of one of his friends such as a nickname, email and SIP address. Five types of location information are supported: geospatial coordinates, civic information, well-known places, user-specified places and user locations. Different location-related operators can be used, such as *near*, *within*, *in*, *outside of* or *moved*. Below we show a location rule using the *near* operator. *Within* means that the user is within a radius of the reference point. *Near* means the same but the radius is a default distance that the user defines in his SECE account.*Outside of* and *in* means that the user is outside of and inside the reference point, which must be represented as a polygonal structure. We are working on a location database that allows users to predefine polygonal locations through a GMaps-based graphical interface. *Moved* means that the user moved the given distance from where he was located when the rule was entered or triggered for the last time.

```
Bob near "Columbia University" {
    if{ my status is idle } { call bob; }
}
```

Context Rules: These specify the action to execute when some context information changes, such as presence or sensor state. These rules always start with the keyword *if*. If the rule is about the user that is entering the rule, this keyword if followed by *my*. Otherwise, the *if* keyword is followed by the friend's identifier. Below, we show an example of a context rule about a friend.

```
if Bob's status is available { alarm me; }
```

Communication Rules: These specify the actions to execute in response to incoming, outgoing or missed communication requests. A request rule can start with the keyword *incoming*, *outgoing* or *missed*, followed by the type of event. The following rule is an example of incoming call handling.

```
incoming call to me.phone.work {
    if { [ my location is not office] } {
        autoanswer audio no_office.au;
        email me "[incoming caller] tried to reach you on your work phone at
        [incoming time]";
    }
}
```

4 Enhancing SECE toward Ontology-Based User-Defined Rules for Automatic Service Discovery

As it stands, SECE has no way of automatically discovering a new type of service, generating a rule language for it and incorporating it in its system. The set of services that are supported in SECE are hard-coded. Thus, we have enhanced SECE to support ontology-based user-defined rules for automatic service discovery. The simple but illustrative example below emails the user whenever a new restaurant that satisfies the given conditions is found.

```
Any japanese restaurant that is cheaper than 20$ and whose location contains Manhattan {
    email me "new restaurant found" "Details: [event description]";
}
```

We have incorporated GloServ, an ontology-based service discovery system, within SECE's back-end architecture. GloServ provides an API whereby service ontology descriptions, for a number of domains, can be downloaded and queried for with an ontology query. GloServ uses the OWL DL ontology to describe its services. Thus, SECE can access these OWL specifications in order to dynamically define rules for the specific service domain. Users are made aware of these services by a front-end application to SECE that displays the discoverable services' descriptions. For each service domain, SECE will provide documentation on how to create rules. Currently, users will still need to learn how the rules are constructed, however, for the future, we plan on building a GUI that will use the ontology description to aid the user in constructing the rules. This section will describe the design and implementation of these enhancements.

4.1 SECE Architecture

Design. Figure 4 outlines the main interactions between SECE, GloServ, front-end applications and web services. Although SECE is a standalone web service, we are enhancing it toward a more flexible architecture. We envision SECE as a common layer on which advanced front-end applications can be built. In this mode of operation, end users are connected to front-end applications that provide more functionality or fancy graphical interfaces. Users that are not comfortable with scripts will therefore be able to use more sophisticated graphical tools with probably advanced online guides. On the other hand, the SECE web service provides a lightweight solution for the sake of simplicity and efficiency. Users can enter rules into the SECE web service quickly without any resource-demanding graphical application, which is very convenient for mobile user devices with limited resources.

From the moment at which a web service rule is entered in SECE on, SECE will periodically communicate with GloServ for discovering the web services that match the rule. A GloServ request specifies the web service of interest as a SPARQL query [22] and matched services' profiles, if any, are sent to SECE into a GloServ response. If a new web service matches a rule, SECE executes the rule's body.

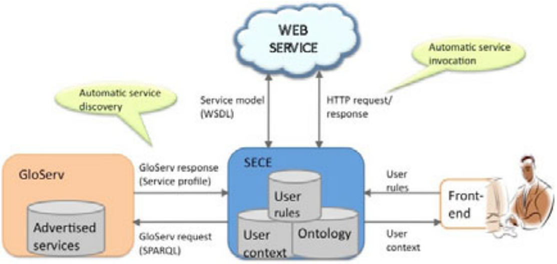


Fig. 4. SECE, GloServ, front-end applications and web services

SECE has a layered architecture, as Figure 5 shows. For details of each of the components, we encourage the reader to refer to the original SECE paper [1]. We will discuss the components that have been added to the enhanced SECE architecture in this section.

The new components that have been added to the SECE architecture are: 1) *WBRL* rule, which implements the web service rules; 2) Jena Ontology Model, which contains the necessary ontologies' schemes; 3) GloServ Context Mediator, which periodically pulls GloServ for checking out new web services of interest.

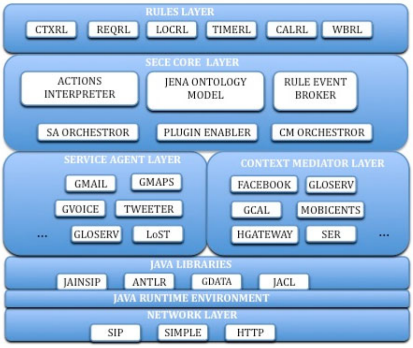


Fig. 5. SECE architecture

Implementation. SECE stores the OWL specifications of web services in an ontology database that is built upon the Jena Framework [23]. When a web service rule is entered into SECE, it has to go through the following steps: 1) parse the rule (i.e., syntactic checking); 2) verify that the described kind of web service exists (i.e., semantic checking); 3) subscribe to the described web service event; and 4) take the rule's actions whenever this event occurs. Figure 6 outlines the main interactions for creating a web service subscription.

The SECE core coordinates the software components in SECE. First, the SECE parser checks that the input rule is consistent with the SECE language, which is generated by an ANTLR grammar [19]. As a result, the parser creates a *WSRule* object that encapsulates information about the rule, namely a web service event and the actions that will be taken if this event occurs. The web service event is defined by the service name and optionally a set of property constraints in the form of *(propertyName, operator, value)*. If the rule parsing is successful, the SECE core verifies that the rule's web service description corresponds to a web service's ontology. To do it, this interacts with the SECE Ontology Model (i.e., *SECEOntModel* in Figure 6). The SECE Ontology Model encapsulates the Jena database that contains the web services' ontologies and provides convenient functions for searching and retrieving information about them. A web service description is semantically correct if there exists a web service's ontology that describes a service that is named as the described web service and can be associated with the described properties and constraints. Thus, SECE will ask the SECE Ontology Model for the namespace URI of the web service and its properties. If this web service does not correspond to any ontology, the SECE ontology Model returns null values. This means that the rule's web service event is semantically incorrect, which results in aborting rule creation and warning the user. Otherwise, the rule's web service event is semantically correct and the SECE core proceeds to create the corresponding subscription (i.e., *WSSubs* in Figure 6).

The SECE core then retrieves an event monitor from the Event Monitor Broker (*OntEM* and *EMBroker* in Figure 6). An event monitor is the agent that watches a particular service and generates an event whenever a new instance of this service is discovered. The Event Monitor Broker maintains a list of the event monitors that are actually monitoring a web service. Thus, if an event monitor for the web service event already exists, the Event Monitor Broker returns it. Otherwise, the Event Monitor Broker creates a new one, appends it to the list of monitors and returns it. Then, the SECE core associates the event subscription with the event monitor and starts the subscription.

Starting and pausing an event subscription makes it subscribe and unsubscribe to the associated event monitor, respectively. When an event monitor receives a subscription request and there are no other subscribers, it creates the corresponding SPARQL query that describes the web service event. This also starts up a recursive timer to query the GloServ Context Mediator (i.e., *GloServCM* in Figure 6) at fixed intervals with the SPARQL query. If this query results in any matched service, the event monitor creates an OntEvent object that describes the discovered service and notifies the subscriber of this event. Note that the outbound messages between GloServCM and GloServ are omitted in Figure 6 because of lack of space. When an event monitor is associated with more than one subscriber, the SPARQL query represents the least restrictive subscription. When a web service matches this subscription, the event monitor checks out whether the service matches any of the other subscriptions. Figure 6 only shows this check on the web service subscription *wss* through the *matchedServ* method. Furthermore, the event monitor maintains a cache of discovered events. When a new subscription is created, this cache is checked out and the matching web services are notified.

4.2 SECE Ontology-Based Sublanguage

SECE provides a simple and generic ontology-based language for end-users to define web service rules. In line with SECE's philosophy, this language looks like natural English and is easy to learn. Its basic structure is "any *service* whose *prop rel value*" given that *service* is a web service class, *prop* is one of this service class' properties and *rel* and *value* represent a restriction on the property. *Rel* is a relational operator that depends on the property's type: *contains* and *is* for strings and $=, <, >, \leq$ and $\geq$ for numbers.

Multiple property constraints can be added by the *and* and *or* boolean operators as for example "any shopping offer whose type contains "ski boots" and whose price is cheaper than 150$". Equality on numeric properties can be

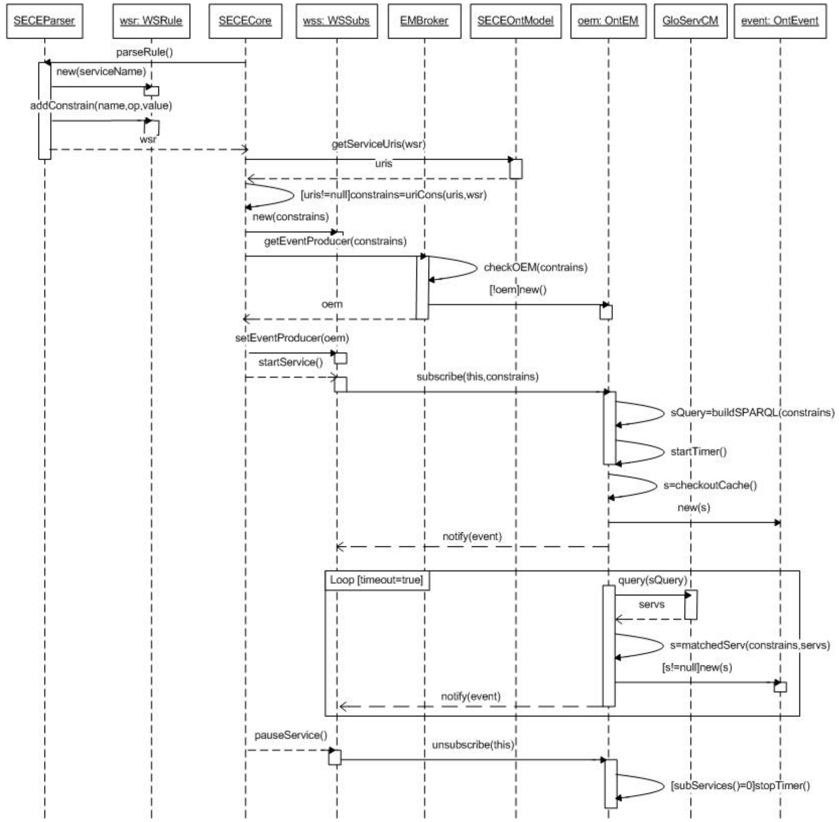


Fig. 6. Sequence diagram from entering a web service rule to querying GloServ

expressed by the verb *has* followed by a number and the property name as in "any happy hour and inexpensive bar that has 20 free seats". Users can place property values before the class name when the property works as adjective. In the previous example, the *bar* class has the boolean properties *happyHour* and *inexpensive*. Boolean constraints can also be expressed by the operators *that has (no)* and *that is (not)* as in "any restaurant that has delivery", "any restaurant that is open 24 hours" and "any cultural exhibition that is free and is not crowded".

Boolean constraints can be applied to class properties or types, which depends on the ontology's structure and is transparent for end-users. An example of boolean property is the above-mentioned *delivery* property whose domain is the *restaurant* class. Boolean constraints on class types restrict inherited types as for example "any restaurant that is southamerican" subscribes to restaurants that are subclasses of the *southamericanRestaurant* class.

5 Future Work

Integrating web service rules into SECE brings out many possibilities in the Semantic Web. Sections 5.1, 5.2 and 5.3 briefly introduce some of these possibilities.

5.1 Automatic Learning of SECE Rules

Automatic suggestions about service composition, events and actions are essential for beginner users, whom can feel lost when creating rules. We are growing SECE towards automation, and hence we find that suggestion systems for SECE rules should be automated too. Thus, in the near future, we will provide a mechanism for front-end applications to build suggestions dynamically. We will model the semantics and syntax of SECE rules ontologically. Front-ends applications will therefore be able to obtain the rules' ontologies and reason about them, and hence they will dynamically create suggestions on rule construction. This mechanism will provide front-end applications with the ability to learn rules' semantics and syntax automatically.

Although SECE offers a set of in-built rules, front-ends may want to offer more sophisticated rules, for example, by combining multiple kinds of events or using an event syntax other than SECE's. To allow front-ends to add new event syntax dynamically into SECE, we will provide proper interfaces to subscribe to events, obtain user context and interact with SECE core functions. Ontologies for rules, events and context as well as semantic paths will provide developers with a high-level interface to SECE data, independent from any underlying data structure. This mechanism may be considered as a plugging system whereby third-parties can insert customized event descriptions into SECE, and SECE will learn these descriptions automatically. To this end, the SECE web service will provide an API.

5.2 Error Detection

Users can create multiple rules of a variety of kinds, which are running parallel on SECE. It is therefore possible that a user creates a rule that involves unexpected results, specially if the user has little experience in creating rules. Although unexpected results are very hard to detect, we can address this issue in several ways. SECE provides online documentation about events and how to create correct compositions. Nevertheless, we can not assume that users enter the correct rules that will execute exactly what they intended. Thus, we are planning to add a dry-run mode that allows users to know what SECE would do when rules are triggered. Actions are just logged, rather than executed. A functionality to test rules before inserting them into SECE will also be provided. This will show the sequence of rule actions, and possibly the value of rule variables, that will be executed when triggering the rule. Moreover, we will avoid infinite loops by simple preventions such as limitations on the times a particular action is invoked and the CPU time taken by a rule.

5.3 Event-Based Context-Aware Web Service Composition System

SECE provides a set of actions for users to build up compositions. Some actions interact with web services, such as *tweet*, *publish* and *email*; other actions send protocol-specific requests, such as *call* (i.e., SIP INVITE); and others are supportive routines. The set of web services with which SECE communicate is static and the communication is hard-coded. Therefore, SECE compositions are static in the sense that, once a composition is created, it will not change. We are planning to incorporate dynamic compositions to SECE through automatic web service discovery and composition. Two new SECE actions will add this functionality: *find* and *plan* for discovery and composition, respectively. An example rule is shown below, in which the *plan* and *find* commands are pseudo-code because they have not been implemented yet. In this example, whenever a new flight is found, other web services are discovered (i.e., hostels, car rentals and restaurants) and composed (i.e., trip planning). Note that the *plan* action could invoke *find* to discovery web services that are necessary for the composition. As the discovered web services and the communication with them can be different each time the composition is executed, we say that this composition is dynamic.

```
Any domestic flight that is cheaper than 200$ and whose date is after June 1, 2011 {
    p=plan flight with hostel and car rental;
    r=find good restaurants according to $p;
    email me “new plan found” “Details: $p $r”;
    sms me “New Plan discovered. See email inbox for details!”;
}
```

With these two new actions, SECE could perform semantic web service discovery and composition that does not need user interaction to be executed; it is automatically triggered by events. In addition, this would also allow combining static and dynamic composition. For example, the rule above provides dynamic composition through the *plan* and *find* actions and static composition through the *email* and *sms* actions. Besides web service discovery events, semantic compositions could be triggered by any SECE event, such as location, context, calendar, communication and time. For instance, the example below discovers web services based on time events. Web services of kind “brunch offer” are found according to the user’s location and are emailed to him or her.

```
Every Sunday at 12:00 {
    offer=find brunch offer whose location is near me”;
    email me “Brunch offer” $offer;
}
```

As the Semantic Web is not widely adopted yet, hybrids platforms like SECE are necessary to offer users flexible and powerful composition tools. Table 1 indicates the types of composition that SECE already supports (white column) and will support in the future (gray columns). Rows define the events that trigger the compositions and columns the types of web service communication in the compositions.

For dynamic compositions, SECE will interact with web services automatically, by retrieving their models and, according to their WSDL specifications, constructing HTTP requests.

Table 1. Types of SECE composition

	Semantic service communication	**Hard-coded service communication**	**Both kinds of communication**
Web service events	Dynamic composition triggered by discovered web services	Static composition triggered by discovered web services *(current contribution)*	Mixed composition triggered by discovered web services
Other events	Dynamic composition triggered by real-world events	Static composition triggered by real-world events *(typical SECE composition)*	Mixed composition triggered by real-world events

6 Conclusions

The Semantic Web is investing much effort in developing standards for providing automatic web service discovery and composition. Although many authors have been interested in this exciting topic in the last decade, complete solutions do not yet exist. Most authors describe or propose theoretical work. The few that present real implementations are partial solutions, domain-specific or lack some desired feature. Thus, there is a strong need for general-purpose platforms for automatic web service discovery and composition. Such platforms should provide intuitive and user-friendly interfaces that do not require engineering or technical skills. Besides template-based composition, end users should be able to orchestrate service composition. Service discovery and composition should be user-centric, context-aware and proactive. To face all these needs, we present a context-aware, event-based platform for service discovery and composition. This platform results from integrating two existing solutions: SECE and GloServ. SECE is a user-centric, context-aware platform for service composition that provides a natural-English-like language for creating event-based rules. GloServ is a scalable network for web service discovery. We implemented the communication between GloServ and SECE. We extended SECE with an ontology database that stores the web services' schemes that are downloaded from GloServ. We also developed an ontology-based language to create rules that work as subscriptions to web service discovery events. This language is independent from any particular web service description, and hence new kinds of service supported by GloServ can be added transparently. We described the whole platform and the advantages it can bring to the Semantic Web. This allows subscribing to web service discovery events by creating rules in a user-friendly language that looks like natural English. SECE also allows creating service compositions that can be triggered by discovered web services and real-world events such as context changes, location, or time. This permits end-users to define and personalize context-aware web service discovery, invocation and composition based on a variety of events. SECE can be decoupled from front-end applications so that more fancy graphical interfaces can be built on top of it. Modeling SECE rules ontologically can provide front-ends with the means of understanding and learning new SECE rules automatically.

Acknowledgments. Victoria Beltran was supported by the Spanish Government through the CICYT project TIC2009-11453 and the FPU grant AP2006-02846.

References

1. Boyaci, O., Beltran, V., Schulzrinne, H.: Bridging communications and the physical world: Sense everything, control everything. In: Proceedings on the IEEE Globecom (UbiCoNet Workshop) (December 2010)
2. Arabshian, K., Schulzrinne, H.: An ontology-based hierarchical peer-to-peer global service discovery system. Journal of Ubiquitous Computing and Intelligence 1(2), 133
3. Arabshian, K., Dickmann, C., Schulzrinne, H.: Service composition in an ontology-based global service discovery system. tech. rep. Columbia University, New York, NY (September 2007)
4. Rosenberg, J., Lennox, J., Schulzrinne, H.: Programming Internet telephony services. IEEE Internet Computing 3, 63–72 (1999)
5. Wu, X., Schulzrinne, H.: Programmable End System Services Using SIP. In: Conference Record of the International Conference on Communications (ICC) (May 2003)
6. Burgy, L., Consel, C., Latry, F., Lawall, J., Palix, N., Reveillere, L.: Language Technology for Internet-Telephony Service Creation. In: IEEE International Conference on Communications, ICC 2006, vol. 4, pp. 1795–1800 (June 2006)
7. Latry, F., Mercadal, J., Consel, C.: Staging telephony service creation: a language approach. In: IPTComm 2007: Proceedings of the 1st International Conference on Principles, Systems and Applications of IP Telecommunications, pp. 99–110. ACM, New York (2007)
8. Jouve, W., Palix, N., Blum, A., Kadionik, P.: A SIP-Based Programming Framework for Advanced Telephony Applications. In: Schulzrinne, H., State, R., Niccolini, S. (eds.) IPTComm 2008. LNCS, vol. 5310, pp. 1–20. Springer, Heidelberg (2008)
9. Cassou, D., Bertran, B., Loriant, N., Consel, C.: A generative programming approach to developing pervasive computing systems. In: GPCE 2009: Proceedings of the Eighth International Conference on Generative Programming and Component Engineering, pp. 137–146. ACM, New York (2009)
10. Dey, A.K., Abowd, G.D.: CybreMinder: A Context-Aware System for Supporting Reminders. In: Thomas, P., Gellersen, H.-W. (eds.) HUC 2000. LNCS, vol. 1927, pp. 172–186. Springer, Heidelberg (2000)
11. task.fm Free SMS and Email Reminders, http://task.fm
12. Ponnekanti, S., Fox, A.: Sword: A developer toolkit for web service composition. In: Proc. of the Eleventh International World Wide Web Conference, Honolulu, HI (2002)
13. Soriano, J., Lizcano, D., Hierro, J., Reyes, M., Schroth, C., Janner, T.: Enhancing user-service interaction through a global user-centric approach to SOA. In: Fourth International Conference on Networking and Services, ICNS 2008, pp. 194–203. IEEE (2008)
14. Yahoo pipes, http://pipes.yahoo.com/pipes/

15. Kazhamiakin, R., Bertoli, P., Paolucci, M., Pistore, M., Wagner, M.: Having Services "YourWay!": Towards User-Centric Composition of Mobile Services. In: Domingue, J., Fensel, D., Traverso, P. (eds.) FIS 2008. LNCS, vol. 5468, pp. 94–106. Springer, Heidelberg (2009)
16. Mobicents, http://www.mobicents.org/
17. SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE), http://datatracker.ietf.org/wg/simple/charter/
18. About SIP Express Router, http://www.iptel.org/ser/
19. Parr, T.: The Definitive ANTLR Reference: Building Domain-Specific Languages. Pragmatic Bookshelf (2007)
20. Ousterhout, J.K., Jones, K.: Tcl and the Tk Toolkit, 2nd edn. Addison-Wesley, Upper Saddle River (2009)
21. Desruisseaux, B.: Internet Calendaring and Scheduling Core Object Specification (iCalendar). RFC 5545 (Proposed Standard), Updated by RFC 5546 (September 2009)
22. W3C, SPARQL Query Language for RDF. Website (January 2008), http://www.w3.org/TR/rdf-sparql-query/
23. Jena - A Semantic Web Framework for Java, Website, http://jena.sourceforge.net/index.html

Random Indexing for Finding Similar Nodes within Large RDF Graphs

Danica Damljanovic[1], Johann Petrak[2], Mihai Lupu[3], Hamish Cunningham[1],
Mats Carlsson[4], Gunnar Engstrom[4], and Bo Andersson[4]

[1] Department of Computer Science, University of Sheffield, United Kingdom
{d.damljanovic,h.cunningham}@dcs.shef.ac.uk
[2] Austrian Research Institute for Artificial Intelligence, Vienna, Austria
johann.petrak@ofai.at
[3] Information Retrieval Facility (IRF), Vienna, Austria
m.lupu@ir-facility.org
[4] AstraZeneca, Lund, Sweden
{Mats.Carlsson,Gunnar.Engstrom,Bo.H.Andersson}@astrazeneca.com

Abstract. We propose an approach for searching large RDF graphs, using advanced vector space models, and in particular, Random Indexing (RI). We first generate documents from an RDF Graph, and then index them using RI in order to generate a semantic index, which is then used to find similarities between graph nodes. We have experimented with large RDF graphs in the domain of life sciences and engaged the domain experts in two stages: firstly, to generate a set of keywords of interest to them, and secondly to judge on the quality of the output of the Random Indexing method, which generated a set of similar terms (literals and URIs) for each keyword of interest.

Keywords: random indexing, vectors space models, information retrieval, RDF graphs, ontologies.

1 Introduction

Recent years have seen a massive increase of highly structured data being made available in the form of RDF triple representations. Both legacy data and new data have been made available in RDF triple format and this representation has also made it worthwhile and feasible to create mappings between RDF data that originates from different legacy sources, leading to potentially very large RDF repositories. Initiatives such as Linked Open Data[1] are working on creation, publication and interlinking of huge RDF graphs.

Traditionally, RDF spaces are being searched using an RDF query language such as SPARQL [15] which allows the formulation of fine-grained queries by their ability to match whole graphs and to create complex conditions on the variables to be bound in the query. This level of complexity and flexibility is

[1] http://linkeddata.org/

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 156–171, 2012.

very useful in many situations, especially when the query is created automatically in the context of an application. However, for end-users who want to explore the knowledge represented in an RDF store, this level of detail is often more of a hindrance: querying the repository is not possible without a detailed knowledge of its structure and the names and semantics of all the properties and classes involved. This is especially the case for large RDF graphs which may have thousands of classes and properties, for example Linked Life Data[2] (5 billion statements), or FactForge[3] (2 billion statements).

In this paper we investigate whether advanced Information Retrieval (IR) methods can bring a new dimension to the task of searching huge RDF graphs. We propose a complementary approach based on word space model, more concretely Random Indexing (RI) [14], for building a *semantic index* for a large RDF graph. Traditionally, a *semantic index* captures the similarity of *terms* based on their contextual distribution in a large document collection, and the similarity between *documents* based on the similarities of the terms contained within. By creating a semantic index for an RDF graph, we are able to determine contextual similarities between graph nodes (e.g., URIs and literals) and based on these, between arbitrary subgraphs. These similarities can be used for finding a ranked list of *similar* URIs/literals for any given input term (a literal or a URI), which can then be used for exploring the repository or enriching SPARQL queries.

We evaluate our approach on subsets of the Linked Life Data (LLD) repository – a large integrated repository which contains 5 billion RDF statements from the biomedical domain, including UniProt[4], PubMed[5], and many more[6]. Our evaluation is based on human judgment by clinical research scientists (from AstraZeneca pharmaceutical company) who were involved in two stages: firstly, to generate a set of keywords of interest to them, and secondly to judge on the quality of the output of the Random Indexing method, which generated a set of similar terms (literals and URIs) for each topic of interest.

2 Related Work

A considerable amount of work has been done in the area of using Information Retrieval methods for the task of selecting and retrieving RDF triples. However, most of these approaches do not take advantage of the latent semantics included in an RDF Graph, as their primary intention is finding the RDF files on the Web relevant to the given keyword and/or a URI. These systems are semantic search engines such as Swoogle [8] or Sindice [19]. They collect the Semantic Web resources from the Web and then index the keywords and URIs against the RDF files containing those keywords and URIs, using the inverted index scheme. These search engines use traditional weighting mechanisms such as TF-IDF.

[2] www.linkedlifedata.com

[3] www.factforge.net

[4] www.uniprot.org/

[5] www.ncbi.nlm.nih.gov/PubMed/

[6] See the full list at: www.linkedlifedata.com/sources

In [11] the authors introduce the *ReConRank* algorithm, which adapts the well-known PageRank algorithm to the Semantic Web data. This method ranks the nodes in a topical subgraph that is selected based on keyword matching from the RDF files. In other words, it ranks the results of a query based on the RDF links in the results. The subgraph that the algorithm identifies includes both the subject nodes related to the query, and also the context of the subject nodes (i.e. the provenances or sources of the subjects), in order to improve the quality of ranking. In comparison to these approaches we use the neighbouring nodes as semantic context for each node in an RDF graph. The nodes and their contexts are used as *virtual documents* for Random Indexing.

In [16], the authors describe an approach for generating a *virtual document* for each URI reference in an RDF triple store (or, equivalently, each node in an RDF graph). The virtual document contains the local name and labels of the URI reference, other associated literals such as those in *rdfs:comment*, and the names of neighbouring nodes in the RDF graph. These virtual documents are then used for ontology matching and also for generating object recommendations for users of Falcons [2]. In comparison to our approach, their neighbouring operations involve only one-step neighbours without including properties. Our approach includes properties, and parts of the TBox, and also can operate on an arbitrarily large graph of neighbouring nodes.

While using latent semantics in the context of large RDF graphs is not widespread, some recent approaches include the TripleRank [10] algorithm, which uses 3-dimensional tensors to represent RDF properties. This approach is evaluated in the context of faceted browsing and shows promising results. Another similar work, described in [17], applies Random Indexing to Wikipedia articles and then links them to URIs in DBpedia.org.

3 Semantic Index

Latent Semantic Analysis (LSA) [7] is one of the pioneer methods to automatically find synonyms. The assumption behind this and other statistical semantics methods is that words which appear in the similar context (with the same set of other words) are synonyms. Synonyms tend not to co-occur with one another directly, so indirect inference is required to draw associations between words used to express the same idea [3]. This method has been shown to approximate human performance in many cognitive tasks such as the Test of English as a Foreign Language (TOEFL) synonym test, the grading of content-based essays and the categorisation of groups of concepts (see [3]). However, one problem with this method is scalability: it starts by generating a $term * document$ matrix which grows with the number of terms and the number of documents and will thus become very large for large corpora. For finding the final LSA model, Singular Value Decomposition (SVD) and subsequent dimensionality reduction is commonly used. This technique requires the factorization of the term-document matrix which is computationally costly and does not scale well. Also, calculating the LSA model is not easily end efficiently doable in an incremental or out-of-memory fashion. The Random Indexing method [18] circumvents these problems by avoiding the need of matrix factorization in the first place.

RI can be seen as an approximation to LSA which is shown to be able to reach similar results (see [14] and [4]). RI can be incrementally updated and also, the $term * document$ matrix does not have to be loaded in memory at once – loading one row at the time is enough for computing context vectors. Instead of starting with the full term-document matrix and then reducing the dimensionality, RI starts by creating almost orthogonal random vectors (index vectors) for each document. This random vector is created by setting a certain number of randomly selected dimensions to either +1 or -1. Each term is represented by a vector (term vector) which is a combination of all index vectors of the document in which it appears. For an object consisting of multiple terms (e.g. a document or a search query with several terms), the vector of the object is the combination of the term vectors of its terms.

Random Indexing relies on the Johnson-Lindenstrauss lemma:

Lemma 1. *Given* $0 < \epsilon < 1$*, a set* X *of* m *points in* R^N*, and a number* $n > n_0 = O(\frac{log(m)}{\epsilon^2})$*, there exists a mapping* $f : R^N \rightarrow R^n$ *such that* $(1-\epsilon)||u-v|| \leq ||f(u) - f(v)|| \leq (1+\epsilon)||u-v||$*, for all* $u, v \in X$*.*

And particularly on the proof provided by Johnson and Lindenstrauss in their 1984 article [13], where they show that if one chooses at random a rank n orthogonal projection, then, with positive probability, the projection restricted to X will satisfy the condition in the Lemma. RI relies on the observation that, in a high dimensional space, a random set of vectors is always almost orthogonal.

In order to apply RI to an RDF graph we first generate a set of documents which represent this graph, by generating one *virtual document* for each URI in the graph (Section 3.1). Then, we generate a semantic index from the virtual documents (Section 3.2). This semantic index is then being searched in order to retrieve similar literals/URIs (Section 3.3).

3.1 Generating Virtual Documents

The task of deriving a set of documents from a huge RDF graph starts with generating a *representative subgraph* for each URI of interest. We shall refer to such an URI as a *representative URI.*

A representative subgraph represents the context of a URI i.e. the set of other URIs and literals directly or indirectly connected to that URI. For a representative URI S, the representative subgraph of order N is a set of all paths of triples $(S, P_1, O_1; O_1, P_2, O_2; \cdots ; O_{N-1}, P_N, O_N)$. If O_N is not a literal we also include all triples O_N, P_{N+1}, L_J where L_J is a literal. In other words, we apply the breadth-first search starting with the representative node, and the depth being defined by N. In addition, we include or exclude certain parts of the TBox: direct classes for instances are excluded ($P_N != rdf : type$), while other annotation properties such as $rdfs : label$ are included. In the experiments reported in this paper, the representative subgraphs are of order 1 ($N = 1$).

We create *virtual documents* by including all paths from representative subgraphs where:

- all URIs of nodes or appearing inside literals are included unchanged;
- for literals we remove punctuation and stop words, and then lowercase the text; we also remove number literals, gene and protein sequences, complex names, and HTML tags[7].

3.2 Generating Semantic Index

There are several parameters which can influence the process of generating semantic index, or vectors using the RI method:

- **Seed length.** Number of +1 and -1 entries in a sparse random vector.
- **Dimensionality.** Dimension of the semantic vector space – predefined number of dimensions to use for the sparse random vectors.
- **Minimum term frequency.** Minimum frequency of a term to get included in the index.

Our experiments study how variations of these parameters influence the quality of the results and how sensitive the method is to that variation.

3.3 Search

Once the semantic index has been created, it can be used to find similarities between URIs, literals, and RDF subgraphs. We use the cosine function to calculate the similarity between the input term (literal or URI) vector and the existing vectors in the generated vector space model. We can perform the following kinds of searches:

1. *finding similarities between two terms*: given a keyword, find similar literals and URIs; this can be used in several ways for example for refinement of SPARQL queries (see [6]); also, it can be used as an alternative way of browsing and finding URIs or literals related to a topic of interest (expressed through a keyword or a set of keywords)
2. *finding documents related to a specific term*: this task would be useful for suggesting a set of representative URIs related to a given keyword.
3. *finding documents related to a document*: this task would be useful for suggesting a set of representative URIs related to a set of URIs.
4. *finding terms related to the specific documents*: this can be used for describing a representative URIs through a set of literals and URIs.

While in the context of large RDF graphs (e.g., LLD) we find all these searches useful, in the experiments we present next we focus on term-term search only.

[7] Although the method described in this paper is domain agnostic, carefully choosing the preprocessing strategies to suit the domain covered by an RDF graph can significantly improve results.

4 Experiments

Our goal in using the Random Indexing method is to investigate whether it can offer an alternative way of searching large RDF spaces, by suggesting literals or URIs which are similar to the topic of interest. As the LLD dataset covers the life sciences domain, we conduct an evaluation experiment with clinical research scientists from AstraZeneca, with the aim to assess this.

4.1 Dataset

Linked Life Data is a dataset covering the life sciences domain, and the latest version 0.6 contains 5,052,047,661 statements in total (for a comparison, one year ago it contained 4,179,999,703 statements). Advanced IR methods based on Vector Space Model (VSM) are computationally expensive, and therefore, before we apply the Random Indexing method on the whole dataset, we evaluate it on two smaller subsets of LLD.

We have generated the two subsets as follows. For 1528 seed URIs (the URIs representing all MEDLINE articles from December 2009) we retrieve neighbouring subgraphs (of order 1) recursively until we reach certain predefined limit of statements, and we refer to these as LLD1 and LLD2. Table 1 shows the sizes of LLD1 and LLD2.

Table 1. Sizes of LLD1 and LLD2 datasets

	LLD 1	LLD2
number of statements	595798	4573668
number of virtual documents	64644	473742
number of terms	417753	1713349

4.2 Evaluation Measures

In order to calculate the correctness of the retrieved terms, there are standard Information Retrieval measures such as *precision*, *recall* and *Mean Average Precision (MAP)*. Precision is defined as the number of relevant documents retrieved divided by the total number of documents retrieved and is usually calculated for certain number of retrieved documents (e.g., Precision@10, Precision@20). Recall is the number of relevant documents retrieved divided by the total number of existing relevant documents (which should have been retrieved).

Mean Average Precision (MAP) is by far one of the most popular measures in IR evaluation because, for each system and set of topics, it provides a single value to measure its performance [5]. Average Precision (AP) is computed for each topic by first calculating precision for each relevant document that is retrieved and then averaging these values. Mean Average Precision is then the mean of these values for all keywords. Furthermore, by the nature of the averaging process, MAP is more sensitive to ranking than *precision* at a specific point, favouring systems which return more relevant documents at the top of the list than at the bottom, whereas *precision* does not make this distinction as long as the results are within the cut-off range.

As our task is to retrieve most relevant literals and URIs first, we used MAP@10. Recall is extremely difficult to measure due to the number of terms in our datasets (see Table 1). In addition, our task is to help domain experts explore large RDF graphs, which is similar to Web search in the sense that there is a vast amount of terms to be searched through, and also a significant number which is relevant for each input term. Hence, for these kinds of tasks, users care more about precision than about recall. Indeed, they care most about the top ranked results, which is exactly what is captured by MAP.

For each query keyword, two clinical research scientists evaluated the relevance of the retrieved terms. All scientists looked at all retrieved terms. *Relevant* were considered only those terms which *both* scientists evaluating a particular query marked as relevant. In order to measure the agreement between scientists on this particular task, we measured the Inter Annotator Agreement (IAA) between the two clinicians, based on the words which both of them marked as relevant/irrelevant.

IAA has been used mainly in classification tasks, where two or more annotators are given a set of instances and are asked to classify those instances into some pre-defined categories. The two commonly used IAA measures are *observed agreement* and *Kappa (κ)* [12].

Observed agreement is the portion of the instances on which the annotators agree. For our case, with the two annotators and two categories (relevant and irrelevant), it is defined as

$$A_o = \frac{a+d}{a+b+c+d} \tag{1}$$

where a refers to the number of terms *both annotators agreed as relevant*, d refers to the number of terms *both agreed as irrelevant*, b refers to the number of terms *annotator 1 marked as relevant, and annotator 2 as irrelevant*, c refer to the number of terms *annotator 1 marked as irrelevant, and annotator 2 as relevant.*

A certain amount of agreement is expected by chance which is not captured by the observed agreement. **Kappa** is defined as the observed agreements A_o minus the agreement expected by chance A_e and is normalized as a number between -1 and 1.

$$k = \frac{A_o - A_e}{1 - A_e} \tag{2}$$

$k = 1$ means perfect agreement, $k = 0$ means the agreement is equal to chance, $k = -1$ means 'perfect' disagreement.

There are two different methods for estimating A_e: in **Cohen's Kappa**, each annotator has a personal distribution, based on his distribution of categories. In **Siegel & Castellan's Kappa**, there is one distribution for all annotators, derived from the total proportion of categories assigned to all annotators (see [9] for more details and for the comparison of the two). We used Cohen's Kappa.

4.3 Experimental Setup

We have performed our experiment through the following steps:

1. **Extracting topics of interest** represented as query terms present in both LLD1 and LLD2. In order to avoid exposing the scientists to learning SPARQL, we have formed a team of one computer scientist and one clinical research scientist. The computer scientist was executing the SPARQL query and browsing through the links and URIs, while the clinical research scientist was only looking at the abstracts which the computer scientist selected. As a result, we obtained 18 keywords which appeared in both LLD1 and LLD2 datasets. We split this set into two halves as shown in Table 2, and then performed the following two steps in two iterations: first, *Group 1* is used for *training* the model, and *Group 2* for *testing* it. In the second iteration, the two sets are swapped.

Table 2. Topics of interest divided into two groups for training/testing the Random Indexing method

Group 1	Group 2
acetylcholinesterase	Posttraumatic Stress Disorder
synergistic effect	trial
cholinergic signaling	bladder cancer
PTSD	Adverse events
antagonist	trauma
efficacy	antioxidant
clinical trial	magnesium
cognitive	cystectomy
lung	5-HT receptors

2. **Training the model.** we generated RI models for several variations of the following RI parameters for both LLD1 and LLD2:
   - vector dimension: 500, 1000, 1500, 1800, 2500
   - seed length: 10, 50, 100, 300, 500, 1000
   - term frequency: 1, 2, 5, 8, 10

   This resulted in 290 runs (145 per dataset[8]). We then searched for the top ten similar words for each topic of interest from the *training* set, and presented them to clinicians who accessed the relevance. The combinations for parameters which lead to the best results (measured through MAP) were considered as the optimal setting for testing the method in the next step.
3. **Testing the model.** for the models generated using the best parameters retrieved in the previous step, we retrieved ten similar words for each topic of interest from the *testing* set and calculated MAP. The correctness of the retrieved terms was assessed by clinical research scientists to whom we gave the terms in the form of a survey (see below).

[8] 5 runs are missing from this count, corresponding to the situation where the seed size is 1000, and the vector dimensionality is 500, which is impossible.

Human Assessment. The retrieved keywords for each topic of interest in both *training* and *testing* sets were assessed by humans. We merged the results from all searches into one pool (sorted alphabetically to avoid bias), and gave this list to the scientists in the form of a survey. When the similar term was a URI, we have extracted the label from LLD and showed it in brackets. This is to ensure that the scientists can concentrate on meaning of these rather than looking and searching LLD in order to find the label. An example task looked similar to this:

```
-----------------------------------------------------------------
Is 'trauma' related to (delete URIs/words which are not related):
-----------------------------------------------------------------
arteriopathy
back-projection
barotraumas
gunshot
http://linkedlifedata.com/resource/umls/id/C0003048 (Animal
                                                     Experimentation)
http://linkedlifedata.com/resource/umls/id/C0004601 (Back Injuries)
http://linkedlifedata.com/resource/umls/id/C0005604 (Birth trauma)
............
```

The most difficult task when designing this experiment was to define the meaning of *relevant*. Relevant, in this context, is any word related to the given keyword. This is a quite broad definition, which has, as it has been reported by clinical research scientists who were involved in this experiment, posed a number of difficulties due to many different levels of relevance. One of them stated that it would not be easy to repeat the same tasks and mark the same words as relevant if they had to repeat the same task again. that are not deleted as relevant. Therefore, only those words which have been marked as relevant twice (by two different clinicians) were eventually used when evaluating our results.

4.4 Results

In this section we first look into the results of training the model and finding the *best parameters* with two separate groups independently. Then, we look at the results of testing the RI method using these *best parameters*.

Training the Model. We expect to see variations of MAP, for different values of dimensionality, seed length, and minimum term frequency parameters. Our goal is to find the combination of parameters for which MAP is highest, so as to use those in the testing phase.

Figure 1 shows the distribution of MAP across all cases, and for each group used for training. It seems that the keywords from *Group 1* were more challenging for the method, as MAP values are much lower on average. However, as we can see in Table 3 results for *Group 1* were better with LLD2 in comparison to LLD1, while for *Group 2* results were better with LLD1. The reason is a high difference in MAP for keywords: *5-HT receptors*, *trauma* and *trial*. All other keywords from Group 1 performed similarly for both datasets. However, looking closely into results of *Group 2* and the differences of MAP per keyword, there

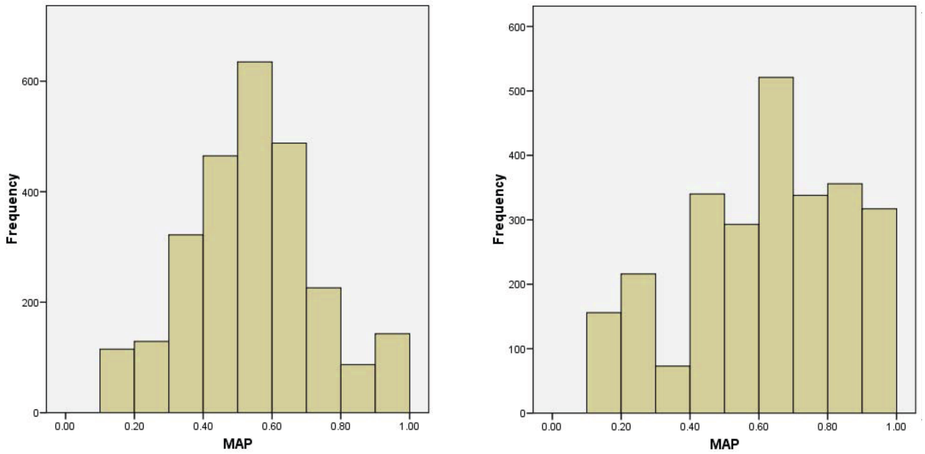


Fig. 1. The distribution of the Mean Average Precision for all combinations of parameters, for *Group 1* (left) and *Group 2* (right) used as training sets

is a fluctuation with one half performing better with LLD1, while the other half performing better with LLD2 (see scattergram in Figure 2).

Table 3. The dispersion values for the distribution of MAP across two datasets

	Dataset	Number of runs	Mean	Std. Deviation
Group 1	1	1305	0.50	0.16
	2	1305	0.59	0.19
	1&2	2610	0.54	0.18
Group 2	1	1305	0.65	0.22
	2	1305	0.59	0.24
	1&2	2610	0.62	0.23

Looking closely into the effect of parameter variations, considering LLD1 and LLD2 groups independently, the results reveal that variations of the parameters did not have major influence on MAP. Detailed diagrams outlining the influence of the variation of all parameters are shown in Figures 3, 4, and 5.

Although parameter variations seem not to have a significant influence on MAP, there are certain patterns which are visible. Namely, the best *minimum term frequency parameter* is in the bottom range (1 for *Group 2* and 2 for *Group 1*) for the smaller dataset, while for the larger, it seems to be in the top (10 for both groups). This might be an explanation for MAP being lower for the larger dataset: more data causes more noise which seems to be filtered nicely using the minimum frequency parameter.

With regards to *dimensionality*, its variation has more influence on MAP with the smaller dataset, than with the larger one. This indicates that the value span which we chose for dimensionality parameter for LLD2 needs to be expanded in

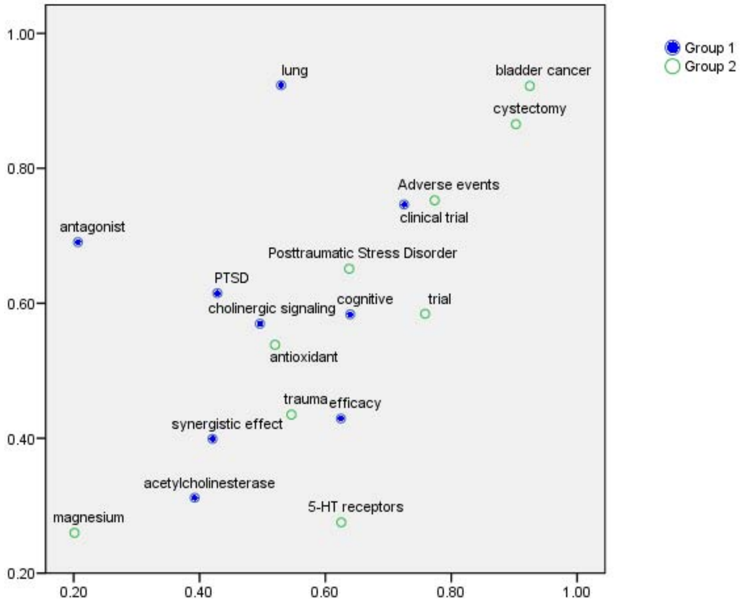


Fig. 2. Correlation of MAP for LLD1 (X axis) and LLD2 (Y axis) for all keywords

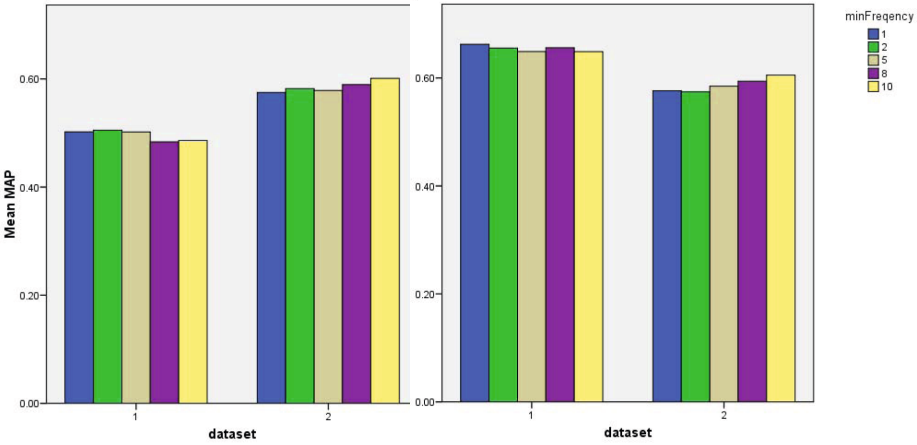


Fig. 3. The effect of the variation of minimal term frequency on *MAP*, across two datasets, for *Group 1* (left) and *Group 2* (right) used as training sets. The distribution of MAP across all categories of minimum term frequency is the same (independent samples Kruskal-Wallis test, p=0.44 and p=0.444 for LLD1 and LLD2 respectively, Group 1; p=0.808 and p=0.784 for LLD1 and LLD2, Group 2).

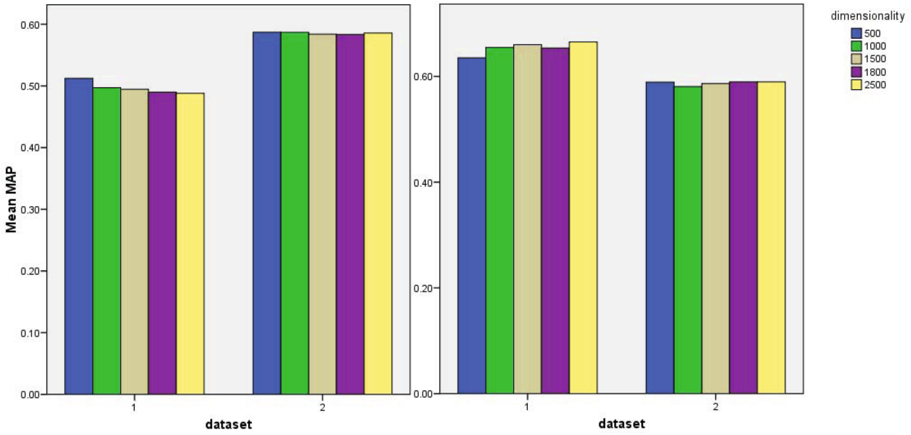


Fig. 4. The effect of the variation of dimensionality on *MAP*, across two datasets, for *Group 1* (left) and *Group 2* (right) used as training sets. The distribution of MAP across all categories of dimensionality is the same (independent samples Kruskal-Wallis test, p=0.676 and p=1.0 for LLD1 and LLD2 respectively, Group 1; p=0.587 and p=0.996 for LLD1 and LLD2, Group 2).

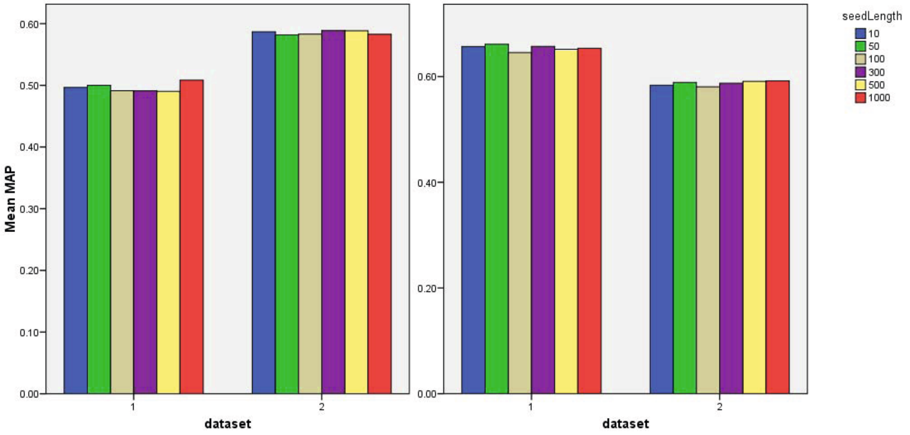


Fig. 5. The effect of the variation of seed length on *MAP*, across two datasets, for *Group 1* (left) and *Group 2* (right) used as training sets. The distribution of MAP across all categories of minimum term frequency is the same (independent samples Kruskal-Wallis test, p=0.931 and 0.997 for LLD1 and LLD2 respectively, Group 1; p=0.961 and 0.998 for LLD1 and LLD2, Group 2).

order to have any effect on results. However, MAP values are within reasonable range.

The variation of *seed length* parameter value seems not to cause any significant changes to MAP across both datasets, and hence, we consider the lowest value of this parameter as the optimal one, due to the fact that the computational

Table 4. Optimal parameters chosen for *Group 1* and *Group 2* used as training sets

	Group 1		**Group 2**	
Dataset	LLD1	LLD2	LLD1	LLD2
Min frequency	2	10	1	10
Seed length	10	10	10	10
Dimensionality	500	500	1500	500
MAP	0.55	0.61	0.65	0.61

resources required to build and search the semantic space are proportional to the value of seed length. Table 4 outlines optimal parameters: those that we chose to use in the testing phase.

Finally, the size of the dataset had a significant influence on MAP (Mann-Whitney U Test, $p < 0.0001$) for both Group 1 and 2 meaning that the larger set (LLD2) resulted in producing the higher value of MAP for Group 1, while for Group 2 the results were better with the smaller dataset (LLD1).

Testing the Model. In what follows we explore whether the model built using the optimal parameters just presented can be used to effectively test the model. In our context, testing the model means evaluating the set of related terms (literals and URIs) returned by our method for the set of testing keywords given as input.

We ran the search method using *Group 2* as a *testing* set against the RI model trained with *Group 1*, and then *Group 1* as a *testing* set against the RI model trained with *Group 2*. Results are shown in Table 5. The RI method results in as good or better MAP for *Group 2* in comparison to MAP for the best trained model (*Group 1* column in Table 4), while for *Group 1* the resulting MAP for LLD2 is as good as that of the best trained model (*Group 2* column in Table 4), while for LLD1 it is lower for 0.15. This is due to the distribution of keywords in *Group 1*, due to which MAP for the RI model with optimal parameters is only 0.05 higher (0.55).

In the testing phase, MAP across both groups reached 0.565 and 0.61 for LLD1, and LLD2 respectively.

Table 5. Testing the Random Indexing method using *Group 2* and *Group 1* as *testing* sets

	Group 2		**Group 1**	
Dataset	LLD1	LLD2	LLD1	LLD2
Min frequency	2	10	1	10
Seed length	10	10	10	10
Dimensionality	500	500	1500	500
MAP	0.63	0.61	0.5	0.61

Also important to observe is the fact that when the data corpus increases (e.g. LLD2 vs. LLD1) the method becomes very stable, and observed MAP values in the training process are reproduced in the subsequent test phase. Arguably this is due to the small difference in MAP across parameters, but it still shows that RI is a stable method even in this unusual use-case we are dealing with.

Human Assessment. In order to assess the overall difficulty of the task which we solve using the RI method, we calculated Inter-annotator agreement, and indeed *Observed agreement* and *Cohen's Kappa agreement* (see Section 4.2). The observed agreement across all keywords was 0.81, and the Cohen's Kappa was 0.61 which indicates that the given task of selecting relevant keywords for a topic of interest was indeed difficult for domain experts.

The code and datasets from the described experiment, including generated virtual documents and semantic spaces, can be downloaded from the LarKC Wiki[9].

Performance. The parameter values affect not only the quality of results but also the required resources and the indexing time. Increasing the value of *dimensionality* and *seed length* almost exponentially increases the time to generate the semantic space (from 0.67 minutes for 500 dimensions to 3 minutes for 2500, LLD1; from 3.78 minutes for 500 dimensions to 11.5 minutes for 2500, LLD2). The higher the value for *seed length* and *dimensionality*, the higher the requirements for the computational resources and RAM in particular[10]. Application of RI to the whole LLD dataset poses the scalability issues related to the size of our corpus. While indexing is a one-off operation (that takes $\tilde{1}$6 hours on MDC computer with 256G RAM), the search for 'lung' after the space is generated takes 14 minutes. Therefore, in our related work reported elsewhere [1] we looked at the parallelisation of the RI search algorithm in order to make exploring large RDF graphs using the contextual similarities of the comprising nodes applicable in real time applications.

5 Conclusion and Future Work

We described the application of the Random Indexing method for the task of searching large and unknown RDF graphs. We tested our method in the domain of life sciences, by training it using the variation of parameters, and then involving domain experts to judge on the relevance of retrieved terms. None of the parameters had a significant influence on MAP, apart from the size of the dataset. However, the value of MAP reaching 0.59 on average indicates that the generation of virtual documents as described in this paper and generating the semantic index using the RI method has promising results, especially considering that the human agreement on the same task revealed its difficulty for domain experts. The reason for the stability of the RI method might have been the span of the parameters which we used, and hence in our future work we will expand the variation span and also repeat the runs across the same parameter variations in order to increase the significance of results.

[9] http://wiki.larkc.eu/LarkcProject/statisticalSemantics

[10] The experiments are conducted on the MDC super-computer: 2 IBM x3950M2, 32 Cores, 256 Gbytes of main memory.

Acknowledgments. We would like to thank creators of the SemanticVectors[11] library which is used in the experiments reported in this paper. This research has been supported by the EU-funded LarKC[12] (FP7-215535) project.

References

1. Assel, M., Cheptsov, A., Czink, B., Damljanovic, D., Quesada, J.: MPI Realization of High Performance Search for Querying Large RDF Graphs using Statistical Semantics. In: García-Castro, R., et al. (eds.) ESWC 2011 Workshops. LNCS, vol. 7117, pp. 156–171. Springer, Heidelberg (2011)
2. Cheng, G., Ge, W., Qu, Y.: Falcons: Searching and Browsing Entities on the Semantic Web. In: Proceedings of WWW 2008, pp. 1101–1102 (2008)
3. Cohen, T., Schvaneveldt, R., Widdows, D.: Reflective random indexing and indirect inference: A scalable method for discovery of implicit connections. Journal of Biomedical Informatics (2009)
4. Cohen, T.: Exploring medline space with random indexing and pathfinder networks. In: AMIA.. Annual Symposium proceedings / AMIA Symposium, pp. 126–130 (2008)
5. Croft, B., Metzler, D., Strohman, T.: Search Engines: Information Retrieval in Practice, 1st edn. Addison Wesley (2009)
6. Damljanovic, D., Petrak, J., Cunningham, H.: Random Indexing for Searching Large RDF Graphs. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010. LNCS, vol. 6088, pp. 1–32. Springer, Heidelberg (2010)
7. Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science 41, 391–407 (1990)
8. Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V., Sachs, J.: Swoogle: a search and metadata engine for the semantic web. In: Proceedings of the 13th ACM International Conference on Information and Knowledge Management, pp. 652–659. ACM, New York (2004)
9. Eugenio, B.D., Glass, M.: The kappa statistic: a second look. Computational Linguistics 1(30) (2004) (squib)
10. Franz, T., Schultz, A., Sizov, S., Staab, S.: TripleRank: Ranking semantic web data by tensor decomposition. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 213–228. Springer, Heidelberg (2009), http://dx.doi.org/10.1007/978-3-642-04930-9_14
11. Hogan, A., Harth, A., Decker, S.: Reconrank: A scalable ranking method for semantic web data with context. In: Second International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2006), Athens, GA, USA (2006)
12. Hripcsak, G., Heitjan, D.: Measuring agreement in medical informatics reliability studies. Journal of Biomedical Informatics 35, 99–110 (2002)
13. Johnson, W.B., Lindenstrauss, J.: Extensions to lipschiz mapping into hilbert space. Contemporary Mathematics 26 (1984)

[11] www.code.google.com/p/semanticvectors/

[12] www.larkc.eu

14. Karlgren, J., Sahlgren, M.: From words to understanding. In: Uesaka, Y., Kanerva, P., Asoh, H. (eds.) Foundations of Real-World Intelligence, pp. 294–308. CSLI Publications, Stanford (2001)
15. Prud'hommeaux, E., Seaborne, A.: SPARQL Query Language for RDF. W3C Recommendation. W3C (January 15, 2008)
16. Qu, Y., Hu, W., Cheng, G.: Constructing virtual documents for ontology matching. In: Proceedings of WWW 2006, pp. 23–31 (2006)
17. Quesada, J., Brandao-Vidal, R., Schooler, L.: Random indexing spaces for bridging the human and data webs. In: d'Amato, C., Fanizzi, N., Grobelnik, M., Lawrynowicz, A., Svatek, V. (eds.) IRMLeS 2010: The 2nd ESWC Workshop on Inductive Reasoning and Machine Learning for the Semantic Web (2010)
18. Sahlgren, M.: An introduction to random indexing. In: Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005. Citeseer (2005)
19. Tummarello, G., Delbru, R., Oren, E.: Sindice.com: Weaving the Open Linked Data. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 552–565. Springer, Heidelberg (2007)

Reconciling Provenance Policy Conflicts by Inventing Anonymous Nodes

Saumen Dey[1], Daniel Zinn[2], and Bertram Ludäscher[1,2]

[1] Dept. of Computer Science, University of California, Davis
[2] Genome Center, University of California, Davis

Abstract. In scientific collaborations, provenance is increasingly used to understand, debug, and explain the processing history of data, and to determine the validity and quality of data products. While provenance is easily recorded by scientific workflow systems, it can be infeasible or undesirable to publish provenance details for all data products of a workflow run. We have developed PROPUB, a system that allows users to publish a *customized* version of their data provenance, based on a set of publication and customization requests, while observing certain provenance publication policies, expressed as logic integrity constraints. When user requests conflict with provenance policies, repair actions become necessary. In prior work, we removed additional parts of the provenance graph (i.e., not directly requested by the user) to repair constraint violations. In this paper, we present an alternative approach, which ensures that all relevant nodes are retained in the provenance graph. The key idea is to introduce new anonymous nodes to represent lineage dependencies, without revealing information that the user wants to protect. With this new approach, a user may now explore different provenance publication strategies, and choose the most appropriate one before publishing sensitive provenance data.

1 Introduction

In the emerging paradigm of collaborative, data-intensive science, sharing data products even prior to publication may be desirable [1,2]. Yet, without a proper scientific publication associated with shared data, its validity and accuracy is difficult to assess. This is problematic in collaborative environments, where data shared by one scientist is used by another scientist as input for further studies. In such settings, *data provenance* (the lineage and processing history of data) can help to ensure data quality [3,4,5,6,7]. It is thus desirable to publish data products together with their provenance. In many cases, however, provenance data can be sensitive and may contain private information or intellectual property that should not be revealed [7,8,5]. Consequently, one has to balance between (i) the desire to publish provenance data so that collaborators can understand and rely on the shared data products, and (ii) the need to protect sensitive information, e.g., due to privacy concerns or intellectual property issues (Figure 1).

We view provenance as a bipartite, directed, acyclic graph, capturing which *data nodes* were consumed and produced, respectively, by *invocation nodes* (computations). Our model thus corresponds to the Open Provenance Model (OPM) which captures the dependencies between data artifacts and process invocations [9].

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 172–185, 2012.

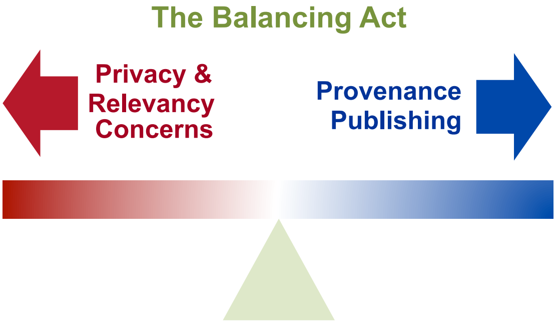


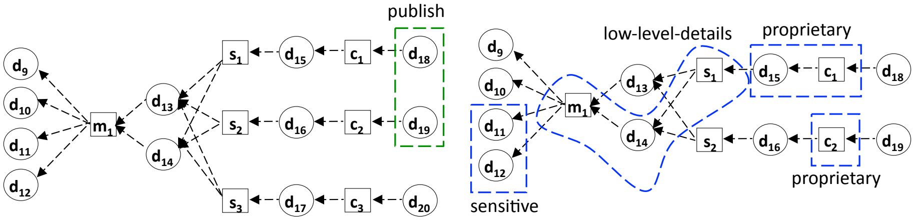
Fig. 1. In collaborative settings, scientists publish provenance for an improved understanding of the result data. With increasing privacy concerns, collaborators have to choose the right balance between providing sufficient provenance data and protecting sensitive information.

To sanitize provenance graphs, a scientist can remove sensitive data nodes or invocations nodes from the provenance graph. Alternatively, she can *abstract* a set of sensitive nodes by grouping them into a single, abstract node. These updates may violate some of the integrity constraints of the provenance graph [10]. For example, grouping multiple nodes into one abstraction node may introduce new dependencies, which were absent in the initial provenance graph. Hiding nodes may also make some nodes in the final graph appear independent of each other even though they are dependent in the original graph. Thus, one can no longer trust that the published provenance data is "correct" (e.g., there are no false dependencies) or "complete" (e.g., there are no false independencies). Therefore, we propose a system that allows a publisher to provide a high-level specification what parts of the provenance graph are to be published and what parts are to be sanitized, *while guaranteeing* that at the same time certain provenance publication constraints are observed.

2 Motivating Example

Figure 2(a) shows a simplified version of the provenance graph (PG) from the First Provenance Challenge [11]. Scientific workflow systems often automatically record such provenance [12,13], and the provenance graphs may resemble the workflow graph, i.e., the former can be seen as instances of the latter [2]. At the workflow specification level, *actors* are used to represent the computational steps, implemented by software components, while at the provenance (or instance) level, we have *invocations* of those actors. We depict *data nodes* as circles and *invocation nodes* as boxes. Dependencies among them are shown as directed edges. These edges capture the lineage of data nodes and thus are typically drawn from right (newer nodes) to left (older nodes), i.e., in the opposite direction of the dataflow edges in a workflow specification. For example, $\mathtt{d}_{16}$ *was generated by* an invocation $\mathtt{s}_2$, and was in turn *used* by invocation $\mathtt{c}_2$, denoted by, respectively $\mathtt{s}_2 \xleftarrow{gen\_by} \mathtt{d}_{16}$ and $\mathtt{d}_{16} \xleftarrow{used} \mathtt{c}_2$.

Assume the user wants to publish data products $\mathtt{d}_{18}$ and $\mathtt{d}_{19}$ along with their provenance information, i.e., the data lineage of these nodes. This publication request is



(a) Provenance graph (PG) and publication user requests

(b) User-selected PG′ ⊆ PG with various privacy issues

Fig. 2. (a) Publication requests to publish the lineage of $\{d_{18}, d_{19}\}$; and (b) privacy issues: (i) data nodes $\{d_{11}, d_{12}\}$ are sensitive, (ii) nodes $\{m_1, d_{14}, s_1\}$ are low level details (i.e. not very useful) for the intended user, and (iii) nodes $\{c_1, d_{15}, c_2\}$ are proprietary

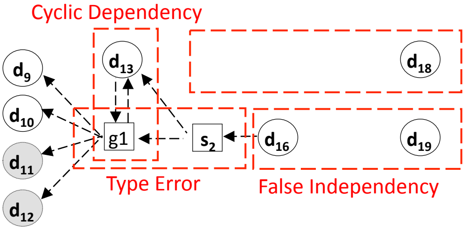


Fig. 3. Provenance graph after resolving all the privacy issues. The modified provenance graph introduces a cyclic dependency, a type error (the graph is non-bipartite), and a false independence.

shown in Figure 2(a). A recursive query is used to retrieve all the data and invocation nodes upstream from d_{18} and d_{19}, i.e., the nodes on which the latter depend. The resulting subgraph (PG′) is shown in Figure 2(b). Note that the lineage of d_{20} up to s_3 is not in the lineage of d_{18} and d_{19} and hence not included in PG′. Further assume that before publishing PG′, the user also wants to sanitize the provenance data as it may have various privacy issues as shown in Figure 2(b).

Figure 3 shows the provenance graph we get after sanitation by (i) removing the value refenences from data nodes d_{11} and d_{12}, (ii) abstracting nodes m_1, d_{14}, and s_1 into a group node g_1, and (iii) removing nodes c_1, d_{18}, and c_2. In this modified provenance graph, we see that there is a cycle between the data node d_{13} and invocation node g_1; a type error for the edge between invocation nodes s_2 and g_1 (the graph should be bipartite); and there are no dependencies from data nodes d_{18} and d_{19} to the rest of the graph. Thus, we need a systematic way to customize provenance while respecting general properties of the graph (e.g., acyclicity, bipartiteness) and while preserving correctness and completeness of the remaining provenance information.

In this work, we develop a strategy to (i) hide sensitive information as specified by the user, (ii) maintain all relevant nodes, which do not have low level details or proprietary information, in the customize provenance graph, (iii) maintain the original direct and transitive dependencies among the relevant nodes, and (iv) produce only graphs that comply with the structural properties of provenance graphs (acyclicity and bipartiteness).

Outline. In Section 3, we describe the provenance model, user requests and provenance policies, and the logical architecture of the framework. Section 4 presents our key ideas, and techniques to solve individual policy violations by introducing new, anonymous nodes. We discuss related work and conclude in Section 5.

3 Provenance Publisher

In our recent work, we developed the system ProPub [10], which uses a declarative approach to publish customized policy-aware provenance. ProPub accepts a provenance graph and three inputs: (1) *user requests* to publish and customize provenance, (2) *provenance policies*, modeled as integrity constraints aiming to ensure the validity of the customized provenance graph, and (3) a (total) *preference order* among provenance policies. ProPub checks whether all user requests and provenance policies can be satisfied together. If not, the approach selects a subset of requests and policies according to the user-specified ranking. The outputs of ProPub are the customized provenance graph, as well as a list of satisfied and ignored user requests and policies.

In this work, we present an extension to ProPub that invents new, anonymous nodes that are inserted in the customized graph. We show that with this technique, it is possible to always satisfy all user requests and policies simultaneously, without the need of a user-specified preference order. For example, by subsequently applying the user requests in a specific way, none of the provenance policies as described in Table 3 will be violated.

Provenance Model. Our provenance model is based on the Open Provenance Model OPM [14] and our earlier work [15]: A *provenance* (or *lineage*) *graph* is an acyclic graph $PG = (V, E)$, where the nodes $V = \mathrm{D} \cup \mathrm{I}$ represent either *data* items D or actor *invocations* I. The graph G is bipartite, i.e., the edges $E = E_{\mathtt{use}} \cup E_{\mathtt{gby}}$ are either *used* edges $E_{\mathtt{use}} \subseteq \mathrm{I} \times \mathrm{D}$ or *generated-by* edges $E_{\mathtt{gby}} \subseteq \mathrm{D} \times \mathrm{I}$. Here, a *used* edge $(\mathtt{i}, \mathtt{d}) \in E$ means that invocation `i` has read `d` as part of its *input*, while a *generated-by* edge $(\mathtt{d}, \mathtt{i}) \in E$ means that `d` was *output* data, written by invocation `i`. Data and invocation nodes have opaque identifiers. We use the relations `data` and `actor` to map each data and invocation node to a URL where the data value can be retrieved or the implenting actor identified. The ProPub Datalog implementation uses the schema shown in Table 1.

User Requests. Table 2 summarizes the user requests supported by our system. User requests are asserted as relational facts, which together with ProPub rules can be used by a Datalog rule engine to infer additional facts or to check integrity constraints. A user request can be a publication request or a customization request.

Figure 4 shows examples of publication and customization requests. The relation `lineage` defines the user's initial publication requests. The relations `abstract`, `hide`, and `anonymize` are used to abstract the nodes with low level details, to remove proprietary nodes, and to remove the value references from the sensitive nodes, respectively.

Table 1. ProPub Provenance Model

Relation	Description
`used(I, D)`	An edge specifying that the invocation `I` used the data artifact `D`.
`gen_by(D, I)`	An edge to indicate that the data artifact `D` was generated by invocation `I`.
`actor(I, A)`	An invocation node `I`, which was executed by actor `A`.
`data(D, R)`	A data artifact node `D`, whose value can be retrieved using the reference `R`.
`dep(X, Y)`	Combined dependency relation `dep` = `used` ∪ `gen_by`. Specifies that node `X` depends on node `Y`, irrespective of their types.

Table 2. User requests for lineage publication and customization

User Request	Description
`ur:lineage(D)`	Selects the complete lineage for the data artifact `D`
`ur:anonymize(N)`	Erases the actor/process identity or the data reference from the node `N`
`ur:hide(N)`	Removes the invocation or data node `N`
`ur:abstract(N, G)`	Collapses all nodes `N` to the abstract group `G`

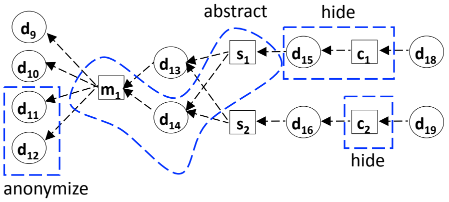
Provenance Policies. As mentioned above, a provenance graph is a bipartite DAG in our model. Moreover, an invocation can read (i.e., use) many data artifacts, but a data artifact is generated by exactly one invocation. We use three provenance policies, represented as logical integrity constraints, to verify if these structural properties are satisfied in the customized provenance graph CG that results from applying all customization requests to PG′. The framework supports two more provenance policies to ensure the correctness and completeness of information, see Table 3.

We use a set of integrity constraints (ICs) to check whether the provenance policies defined in Table 3 are satisfied. Table 4 lists the *witness relations* that are used to detect particular IC violations and report the "culprits". For example, we can detect a write conflict, where a data node `D` is created by different invocations `X` and `Y`, with the Datalog rule: `ic:wc(X,Y) :- gen_by(D,X), gen_by(D,Y), X` $\neq$ `Y.`

3.1 Logical Architecture

The logical architecture of the framework is shown in Figure 5. The user submits a set of publication and customization requests U_0. The module Direct-Conflict-Detection detects direct conflicts among the given user-requests. For example, a `ur:hide` and a `ur:lineage` request on the same node are directly in conflict. The user then needs to update her original requests until all direct conflicts are resolved, resulting in a consistent, conflict-free user request U. The Lineage-Selection module computes the subgraph PG′, containing all to-be-published data items together with their data lineage.

The User-Request-Application module *applies* all the `ur:hide`, `ur:abstract`, and `ur:anonymize` requests in U on PG′. It deletes from PG′ all data and invocation nodes selected by the `ur:hide` and `ur:abstract` requests, together with their associated `gen_by` and `used` edges. This module then applies the `ur:anonymize` requests to remove value references (in case of a data node) or references to the source code (in case of an invocation node). As this module deletes nodes and incident edges, two relevant nodes may now appear independent, even though they were dependent in PG′.



(a) Provenance subgraph PG' with outlined customization requests.

Publication User Requests
lineage(d_{18}). lineage(d_{19}).

Customization User Requests
abstract($d_{14,}$ g1). abstract($s_{1,}$ g1). abstract($m_{1,}$ g1).
hide(d_{15}). hide(c_1) . hide(c_2).
anonymize(d_{11}). anonymize(d_{12}).

(b) Publication and customization requests as Datalog facts.

Fig. 4. (a) User requests to abstract, anonymize, and hide parts of PG'; e.g., proprietary nodes as in Fig. 2(b), can be hidden using the ur:hide user request. (b) User requests represented as Datalog facts: e.g., to abstract nodes $\{\mathtt{m}_1, \mathtt{d}_{14}, \mathtt{s}_1\}$ into an abstract node $\mathtt{g}_1$, we use $\mathtt{abstract}(\mathtt{m}_1, \mathtt{g}_1)$, $\mathtt{abstract}(\mathtt{d}_{14}, \mathtt{g}_1)$, and $\mathtt{abstract}(\mathtt{s}_1, \mathtt{g}_1)$. The module prefix "ur:" is optional here.

Table 3. Provenance Policies

Provenance Policy	Description
No-Write Conflict (NWC)	A data artifact can be written by only one invocation.
No-Cyclic Dependency (NCD)	There is no cycle in the provenance graph.
No-Type Error (NTE)	Bipartite graph: edges only between data and invocations.
No-False Dependence (NFD)	Two nodes are dependent in CG only if they are dependent in PG.
No-False Independence (NFI)	Two nodes are independent in CG only if they are independent in PG.

The Dependency-Injection module connects all relevant nodes in the customized provenance graph CG by reproducing the same dependencies as found originally in PG'. While connecting nodes, this module introduces anonymous nodes to avoid cycle-dependency, type-error, write-conflict, and false-dependency constraint violations.

The final output, the customized provenance graph CG, satisfies all the provenance policies mentioned in Table 3, honors all the conflict-free user requests, and maintains all relevant nodes.

4 Approach

The basic idea of our approach is to first remove data or invocation nodes based on the user's hide and abstract requests, and then to connect the remaining nodes using three key ideas: (i) maintain all relevant nodes, (ii) maintain their dependencies, and (iii) invent new, anonymous nodes to avoid policy violations.

Maintain Relevant Nodes. In case nodes have sensitive or proprietary information, or simply too much, low level details (cf. [16]), the user can request those nodes to be removed, abstracted, or anonymized using the requests described above. All the data and invocation nodes, which are not selected using ur:abstract or ur:hide are considered *relevant* nodes to the user. Our approach does not remove any of these nodes from PG' and in turn maintains them in CG.

Table 4. Integrity constraint relations used to detect policy violations

Constraint	Description
ic:wc(X, Y)	Write conflict: two invocations X and Y are generating the same data node.
ic:cd(X)	Cyclic dependency through node X.
ic:te(X, Y)	Type error: nodes X and Y are connected via *used* or *gen_by* edges, but don't have the corresponding node types.
ic:fd(X, Y)	False dependency: node Y depends on X in CG, but not in PG.
ic:fi(X, Y)	False independence: node Y depends on X in PG, but not in CG.

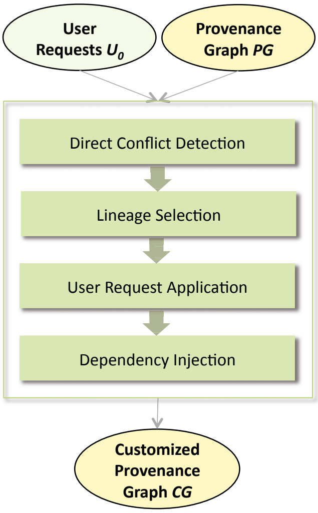


Fig. 5. Logical Architecture: The framework accepts a set of user requests and the provenance graph and runs through a series of four modules to produce the customized provenance graph.

Maintain Dependencies. While removing nodes from PG′, as a consequence of the user's customization requests, we also remove the associated gen_by and used edges. This may make CG incomplete (i.e., dependencies are omitted) as shown in Fig. 3. Our framework avoids these provenance policy violations by maintaining the dependencies among the relevant nodes as described in Section 4.2.

Inventing New Nodes. While connecting the remaining data and invocation nodes, the framework may invent new nodes to avoid policy violations (e.g., an invocation node is invented to connect two data nodes to avoid NTE violations, i.e., type errors). New nodes can be data or invocation nodes and have no relation to any of the nodes being replaced. In particular, new nodes are anonymous, i.e., do not have references associated, so that no sensitive information is revealed, as requested by the user.

4.1 Dealing with Structural Constraint Violations

Before describing our approach in detail (see Section 4.2), we first provide an overview of the possible remedies that we can use to deal with certain constraint violations.

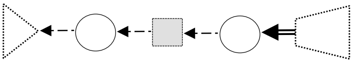


Fig. 6. An invocation node is invented to connect two data nodes, avoiding a NTE violation

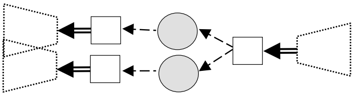


Fig. 7. Two data nodes are invented between invocation nodes, avoiding a NWC violation

No-Type Error. This policy is violated in case there is a direct dependency between two nodes of the same type (i.e., a direct dependency between data nodes or invocation nodes). While connecting two data nodes, our framework invents an invocation node to avoid this policy violation as shown in Fig. 6. Similarly, the framework invents a data node to connect two invocation nodes.

No-Write Conflict. If a data node depends on two different invocation nodes (i.e., the data node is generated by two different invocations), this policy is violated. This may occur when inventing a data node and connecting it with two or more gen_by edges to maintain dependencies. While adding edges to an invented data node or connecting a disconnected relevant data node, our framework ensures that only one gen_by edge is added. Thus, our framewrok avoids this policy violation as shown in Fig. 7.

No-False Independence. This policy is violated if two nodes are dependent in PG', but appear independent in CG. This may occur as a result of user requests, as shown in Fig. 8. Our framework connects the corresponding nodes in CG, to preserve the dependence present in PG'.

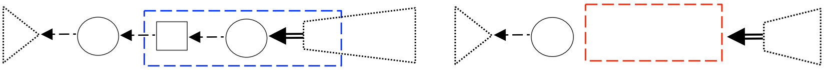
No-False Dependence. This policy is violated if two nodes are independent in PG', but appear dependent in CG (Fig. 9). Our framework avoids this conflict by connecting the relevant nodes using a number of anonymous nodes to preserve the dependencies in CG as they were in PG' (see Section 4.2 for details).

No-Cyclic Dependency. This policy is violated, if there is a cyclic dependency in the provenance graph, i.e., there are nodes in the graph that depend on themselves (either directly or indirectly via other nodes).[1] If the original PG' was acyclic, then the resulting graph CG will also be acylic, as we do not introduce cycles between nodes from PG', nor do we introduce cycles involving newly inserted nodes.

4.2 Module Implementation

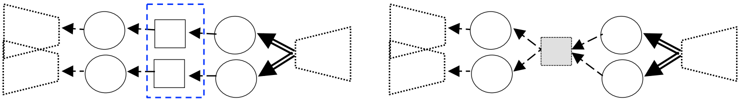
We now provide more details about each of the modules mentioned in Section 3.1. In our framework, the provenance graph PG and user requests U_0 are given as logic facts

[1] Recall that provenance (lineage) graphs are inherently acyclic, since they behave like causality graphs, where an effect (the data output of a computation) cannot precede its cause (the inputs to that computation).



(a) User request to `hide` nodes (b) Policy Violation

Fig. 8. Once the requests to hide nodes are executed (a), a false independence arises (b)



(a) User request to `abstract` nodes (b) Policy Violation

Fig. 9. Once the invocation nodes are abstracted (a), false data dependencies appear (b)

(EDB or base relations in Datalog parlance). All four modules in our framework are specified declaratively, as a set of Datalog rules that are evaluated over PG and U_0, to derive the customized provenance graph CG. The modules Direct-Conflict-Detection and Lineage-Selection are implemented using Datalog rules as shown in our earlier work [10]. The lineage subgraph PG′, expressed as the dependency relation `dep'`, is calculated (based on the `ur:lineage` publication user requests) as follows:

```
dep'(X,Y) :- ur:lineage(X), dep(X,Y).
dep'(X,Y) :- dep'(_,X), dep(X,Y).
```

Note that `dep'` is not the transitive closure of `dep` but rather the subgraph of edges in `dep` that is reachable from the nodes in `lineage` that the user requested to be published.

User-Request-Application. This module accepts the provenance graph PG′ and the conflict-free user requests as inputs. It removes from PG′ the nodes selected by the `ur:abstract` and `ur:hide` requests and their incident edges. For example, the following rules are used to apply the `ur : hide` user requests:[2]

```
del_node(N) :- ur:hide(N), node'(N).
del_dep(X,Y) :- ur:hide(X), dep'(X,Y).
del_dep(X,Y) :- ur:hide(Y), dep'(X,Y).
```

In a similay way, `ur:abstract` user requests are applied. This module then applies all the `ur:anonymize` user requests by removing the references to the value for the selected data nodes and removing the references to the source code for the selected actor nodes. At the end of this module, we get a graph with only relevant nodes, in which some of them are anonymized. However, many of the dependencies among relevant nodes in PG′ may now be missing in this graph.

Dependency-Injection. The objectives of this module are to (i) connect all the relevant nodes (bringing back lost dependencies) using a minimum number of invented nodes, and (ii) maintain the original direct and transitive dependencies among the remaining

[2] Relations whose name starts with "`del_`" denote auxiliary relations that mark items to be deleted, here, e.g., nodes and edges.

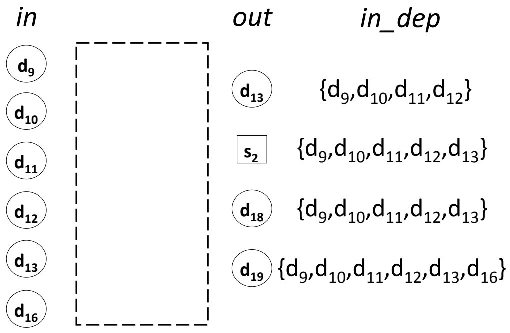


Fig. 10. Stage I: Nodes of the *in* and *out* sets and *in_dep* dependencies from *out* to *in* nodes

nodes. This module performs a 3-stage process to achieve these goals. We describe the stages below:

In the first stage, the framework develops *in* and *out* sets as shown in Fig. 10. Here *in* is a set of relevant nodes (those in dep′), on which one or more nodes from the del_node relation are dependent, while *out* is a set of relevant nodes, which are dependent on one or more nodes in the del_node set. In our running example from Fig. 4(a), we have *in* = $\{\mathtt{d}_9, \mathtt{d}_{10}, \mathtt{d}_{11}, \mathtt{d}_{12}, \mathtt{d}_{13}, \mathtt{d}_{16}\}$ and *out* = $\{\mathtt{d}_{13}, \mathtt{s}_2, \mathtt{d}_{18}, \mathtt{d}_{19}\}$.

The framework also calculates an *in_dep* relation for each node in *out*: in_dep′(o,i) holds for a node $o \in out$, if o depends (directly or transitively) on $i \in in$.

Given $o \in out$, the *in_dep* set (for o) is the set of inputs i on which o depends, i.e., $in\_dep_o = \{i \in in \mid \mathtt{in\_dep}(o, i) \text{ holds}\}$. For example, data node $\mathtt{d}_{19}$ is dependent on all the members of the *in* set (see Fig. 10).

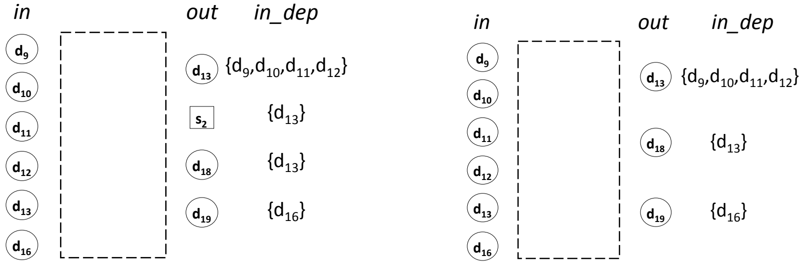
We use the following Datalog rules[3] to calculate *in*, *out*, and *in_dep*:

```
in(Y) :- del_node(X), dep′(X,Y), ¬del_node(Y).
out(X) :- del_node(Y), dep′(X,Y), ¬del_node(X).
in_dep(X,Y) :- dep′*(X,Y), in(Y), out(X).
```

In the second stage, the framework analyzes the dependencies among the nodes of a specific in_dep_o set of a node o. In case there is a node $i \in in\_dep_o$ which depends on another node $i' \in in\_dep_o$, the framework removes i' from in_dep_o. All these i' nodes in in_dep_o are called *redundant* for the node o. The reason is that when o depends on i, then o will also transitively depend on i' (in a sense, i' is "covered" by i, since the lineage of i includes the lineage of i'). For example, node $\mathtt{d}_{18}$ from the *out* set has an *in_dep* set with nodes $\mathtt{d}_9$, $\mathtt{d}_{10}$, $\mathtt{d}_{11}$, $\mathtt{d}_{12}$ and $\mathtt{d}_{13}$. Now, since $\mathtt{d}_{13}$ is dependent on all of $\mathtt{d}_9$, $\mathtt{d}_{10}$, $\mathtt{d}_{11}$, and $\mathtt{d}_{12}$ as shown in Fig. 4(a), the framework optimizes this *in_dep* set by removing all nodes except the node $\mathtt{d}_{13}$. This process is performed for all elements from the *out* set. The result is shown in Fig. 11(a).

Next, we check if there is any (transitive) dependency that uses only non-deleted nodes and edges between a node from the *out* set and the nodes of its *in_dep* set. In case there is, the respective node from the *in_dep* set is removed, since the required dependency is already present in the graph.

[3] Here dep′* denotes the transitive closure of dep′ and is defined as usual in Datalog.



(a) *in*, *out*, and reduced *in_dep* sets (b) reduced *out* set

Fig. 11. Stage II: (a) *in_dep* sets are "optimized" (reduced) by removing redundant nodes. (b) *out* is further reduced by removing nodes which are directly dependent on all *in_dep* nodes.

For example, there is a direct dependency from $\mathtt{s}_2$ to $\mathtt{d}_{13}$. Thus $\mathtt{d}_{13}$ is removed from the *in_dep* set of the invocation node $\mathtt{s}_2$. Finally, if an *in_dep* set for an *out* node becomes empty, the node is removed from the *out* set. This is the case for the invocation node $\mathtt{s}_2$ as shown in Fig. 11(b). Since there is already an edge between $\mathtt{s}_2$ and $\mathtt{d}_{13}$, the framework does not add an additional edge.

```
in_dep1(X,Y2):- in_dep(X,Y1), in_dep(X,Y2), dep'*(Y1,Y2).
in_dep2(X,Y) :- in_dep(X,Y), ¬in_dep1(X,Y).
dep''(X,Y) :- dep'(X,Y), ¬del_node(X), ¬del_node(Y).
dep''*(X,Y) :- dep''(X,Y). % transitive dependencies via remaining nodes
dep''*(X,Z) :- dep''(X,Y), dep''*(Y,Z).
in_dep3(X,Y):- in_dep2(X,Y), dep''*(X,Y).
in_dep4(X,Y) :- in_dep2(X,Y), ¬in_dep3(X,Y).
```

In the third and final stage, we invent one data node for each invocation node A in the *in* set and updates all the *in_dep* sets by replacing X with the respective newly invented data node D = f(A) while keeping the dependencies by connecting D with A. This is done to avoid type-errors. The Datalog rules are as follows:

```
in_actor(A) :- in(A), actor(A,_).
ins_data(f(A)) :- in_actor(A).
ins_dep(f(A),A) :- in_actor(A).
in_dep5(X,Y) :- in_dep4(X,Y), ¬in_actor(Y).
in_dep5(X,f(Y)) :- in_dep4(X,Y), in_actor(Y).
```

We can now calculate distinct *in_dep* sets using the following Datalog rules:

```
diff_in_dep(X1,X2):- in_dep5(X1,Y),in_dep5(X2,_),¬in_dep5(X2,Y).
diff_in_dep(X1,X2):- diff_in_dep(X2,X1).
same_in_dep(X1,X2):- in_dep5(X1,_), in_dep5(X2,_),
                      ¬diff_in_dep(X1,X2).
not_smaller(X):- same_in_dep(X,Y), X > Y.
unique(X):- out(X), ¬not_smaller(X).
```

Here, the unique relation provides the list of *out* nodes with unique *in_dep* sets. The relation same_in_dep pairs *out* nodes having the same *in_dep* sets.

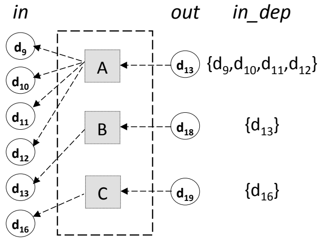


Fig. 12. Stage III: Dependencies are recreated among relevant nodes, based on the *in_dep* sets

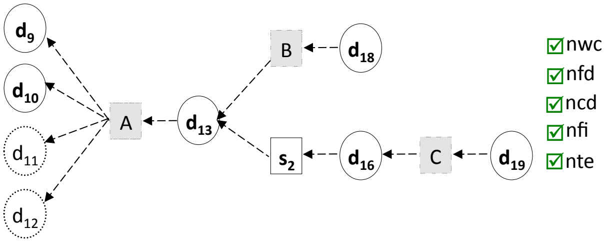


Fig. 13. Customized Provenance Graph CG after applying all user requests. CG satisfies all provenance policies and maintains all relevant PG nodes. `A`, `B`, and `C` are new anonymous nodes.

We then invents on data node `f(Y)` for each unique `same_in_dep` group `Y`, and inserts edges between `f(Y)` and all invocation nodes in this group:

```
ins_data(f(Y)) :- actor(X,_), same_in_dep(X,Y), unique(Y).
ins_dep(X,f(Y)) :- actor(X,_), same_in_dep(X,Y), unique(Y).
```

Finally, we invent one invocation node for each *same_in_dep* group, and insert dependency edges to connect the nodes in *in* and *out* based on the *in_dep*; see Fig. 12:

```
ins_actor(g(S)) :- unique(S).
ins_dep(D,g(S)) :- same_in_dep(D,S), unique(S), data(D,_).
ins_dep(f(S),g(S)) :- same_in_dep(X,S), unique(S), actor(X,_).
ins_dep(g(S),D) :- unique(S), in_dep5(S,D).
```

The result of this module is a graph with relevant and newly created nodes. For our example, it is shown in Fig. 13. PROPUB has removed all the nodes selected using the `ur:abstract` and `ur:hide` user requests and invented three anonymous node `A`, `B`, and `C` to maintain the dependencies among the relevant nodes. The framework also anonymized the data nodes d_{11} and d_{12} selected using `ur:anonymize` user requests.

5 Summary and Conclusions

Data provenance can be used in many ways, e.g., to interpret results, diagnose errors, fix bugs, improve reproducibility, and generally to build trust on the final data products and the underlying processes [3,4,5,6,7]. In addition, provenance information can be used to enhance exploratory processes [17,18,19], and techniques have been developed to efficiently store and query provenance from scientific workflow runs [20,21].

With the increasing use of provenance information, privacy issues become more important as well [7,8]. For example, provenance recorded by a scientific workflow system may carry sensitive information, such as data about human subjects in the case of biomedical studies, or proprietary information that a provenance provider might not want to reveal. By studying and analysing workflow provenance, one can, e.g., infer parts of the workflow specification or guess actor functionality from observing the relationships between inputs and outputs. The security view approach [5] limits the available provenance to a user by providing a partial view of the workflow through a role-based access control mechanism, and by defining a set of access permissions on actors, channels, and input/output ports as specified by the workflow owner at design time. The ZOOM*UserViews approach [16] allows to define a partial, zoomed-out view of a workflow, based on a user-defined distinction between relevant and irrelevant actors. Provenance information is restricted by the definition of that partial view of the workflow.

In our recent work [10], we developed PROPUB, which uses a declarative approach to publish customized policy-aware provenance. Conflicts between user requests to hide or anonymize provenance information and provenance policies are resolved in [10] by removing additional nodes, beyond those requested by the user. In contrast, in this paper, we developed a new way to reconcile conflicts by inventing anonymous nodes that preserve the original lineage dependencies, without revealing information that the user wants to protect. Using this approach, we can now (i) honor all conflict-free user requests, (ii) comply with all provenance policies, (iii) maintain all relevant nodes in the final provenance graph, and (iv) maintain the original direct and transitive dependencies among the remaining nodes. Our current PROPUB system is based on the open provenance model (OPM). We plan to extend our prototype to include provenance model extensions, e.g., to support structured data items, e.g., nested data collections [21].

References

1. Nature: 461, Special Issue on Data Sharing (September 2009)
2. Missier, P., Ludäscher, B., Bowers, S., Dey, S., Sarkar, A., Shrestha, B., Altintas, I., Anand, M., Goble, C.: Linking multiple workflow provenance traces for interoperable collaborative science. In: 2010 5th Workshop on Workflows in Support of Large-Scale Science (WORKS), pp. 1–8. IEEE (2010)
3. Bose, R., Frew, J.: Lineage retrieval for scientific data processing: a survey. ACM Computing Surveys (CSUR) 37(1), 1–28 (2005)ss
4. Simmhan, Y., Plale, B., Gannon, D.: A survey of data provenance in e-science. ACM SIGMOD Record 34(3), 31–36 (2005)
5. Chebotko, A., Chang, S., Lu, S., Fotouhi, F., Yang, P.: Scientific workflow provenance querying with security views. In: The Ninth International Conference on Web-Age Information Management, WAIM 2008, pp. 349–356. IEEE (2008)
6. Freire, J., Koop, D., Santos, E., Silva, C.T.: Provenance for Computational Tasks: A Survey. Computing in Science and Engineering 10(3), 11–21 (2008)
7. Davidson, S., Khanna, S., Roy, S., Boulakia, S.: Privacy issues in scientific workflow provenance. In: Proceedings of the 1st International Workshop on Workflow Approaches to New Data-centric Science, pp. 1–6. ACM (2010)

8. Davidson, S.B., Khanna, S., Tannen, V., Roy, S., Chen, Y., Milo, T., Stoyanovich, J.: Enabling Privacy in Provenance-Aware Workflow Systems. In: CIDR, pp. 215–218 (2011)
9. Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., et al.: The open provenance model core specification (v1. 1). Future Generation Computer Systems (2010)
10. Dey, S.C., Zinn, D., Ludäscher, B.: PROPUB: Towards a Declarative Approach for Publishing Customized, Policy-Aware Provenance. In: Bayard Cushing, J., French, J., Bowers, S. (eds.) SSDBM 2011. LNCS, vol. 6809, pp. 225–243. Springer, Heidelberg (2011)
11. Moreau, L., Ludäscher, B., Altintas, I., Barga, R., Bowers, S., Callahan, S., Chin, J., Clifford, B., Cohen, S., Cohen-Boulakia, S., et al.: Special issue: The first provenance challenge. Concurrency and Computation: Practice and Experience 20(5), 409–418 (2008)
12. Ludäscher, B., Bowers, S., McPhillips, T.M.: Scientific Workflows. In: Encyclopedia of Database Systems, pp. 2507–2511. Springer, Heidelberg (2009)
13. Davidson, S.B., Boulakia, S.C., Eyal, A., Ludäscher, B., McPhillips, T.M., Bowers, S., Anand, M.K., Freire, J.: Provenance in Scientific Workflow Systems. IEEE Data Engineering Bulletin 30(4), 44–50 (2007)
14. Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., Plale, B., Simmhan, Y., Stephan, E., den Bussche, J.V.: The Open Provenance Model core specification (v1.1). Future Generation Computer Systems 27(6), 743–756 (2011)
15. Anand, M., Bowers, S., McPhillips, T., Ludäscher, B.: Exploring Scientific Workflow Provenance using Hybrid Queries over Nested data and Lineage Graphs. In: Winslett, M. (ed.) SSDBM 2009. LNCS, vol. 5566, pp. 237–254. Springer, Heidelberg (2009)
16. Biton, O., Cohen-Boulakia, S., Davidson, S.: Zoom* userviews: Querying relevant provenance in workflow systems. In: Proceedings of the 33rd International Conference on Very Large Data Bases, VLDB Endowment, pp. 1366–1369 (2007)
17. Davidson, S., Freire, J.: Provenance and scientific workflows: challenges and opportunities. In: SIGMOD Conference, pp. 1345–1350. Citeseer (2008)
18. Freire, J., Silva, C., Callahan, S., Santos, E., Scheidegger, C., Vo, H.: Managing rapidly-evolving scientific workflows. Provenance and Annotation of Data, 10–18 (2006)
19. Silva, C., Freire, J., Callahan, S.: Provenance for visualizations: Reproducibility and beyond. Computing in Science & Engineering, 82–89 (2007)
20. Heinis, T., Alonso, G.: Efficient Lineage Tracking For Scientific Workflows. In: SIGMOD, pp. 1007–1018 (2008)
21. Anand, M., Bowers, S., Ludäscher, B.: Techniques for efficiently querying scientific workflow provenance graphs. In: 13th Intl. Conf. on Extending Database Technology (EDBT), pp. 287–298 (2010)

Information Resource Recommendation in Knowledge Processes

Tadej Štajner, Dunja Mladenić, and Marko Grobelnik

Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
{tadej.stajner,dunja.mladenic,marko.grobelnik}@ijs.si

Abstract. This paper proposes a framework and an implementation for proactive just-in-time information resource delivery, based on knowledge processes. We focus on providing a desktop application that presents a ranked list of information resources, such as documents, web sites or e-mail messages, that are considered to be most relevant in that point in time. The paper decomposes the recommendation problam into subproblems and provides evaluation on several event, action and process models. Results show that defining actions based on clustering of events yields the best recommendations.

Keywords: Process modeling, Information delivery, Clustering, Process mining, Just-in-time Information Retrieval.

1 Introduction

Given the scenario of knowledge workers in an enterprise whose workflows consist of accessing various resources, such as web pages, documents and e-mail messages, we propose a prototype for process-based prediction and its implementation. The premise of this work is that we can learn a model from the knowledge worker's use of information resources. We refer to that model as the knowledge process model. We use the learned model with the purpose of streamlining the workflow by suggesting information resources to the user before he needs to retrieve them [10]. For the purpose of this paper, we use the term *knowledge process model,* which corresponds to a set of patterns that describe how a knowledge worker is using various information resources [1]. We construct it by learning from existing event logs under the assumption that the event logs are generated by an underlying knowledge process model.

We capture these events in an on-line fashion through workspace instrumentation and logging infrastructure. For example, the events consist of visiting search engines, web sites, documents on the desktop and receiving and sending e-mail. We use the resulting event stream to learn the model, as well as react with suggestions for information resources given the live data. The data in our model is constructed from three basic components: text, social network, and time.

We decompose the problem into subproblems that form a *knowledge process framework* that enables us to consider process mining on top of primitive events. We outline various *event*, *action* and *process* models that fit various scenarios and

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 186–193, 2012.

patterns that we have encountered in our experiments. The *event models'* role is feature construction: fitting sequences of primitive events with various flavors of content, semantics and social network information and encoding them into feature vectors. The *action models* take these sequences of feature vectors and transform them into sequences of actions, which can them be processed by *process models* to act as predictors of most likely future actions, from which we can then apply a ranking on a candidate set of information resources. Information resources should have a representation compatible with the representation of the primitive events.

2 Related Work

Process mining is often used to uncover dependencies, control flows and patterns in a given business process execution log, encoding the model in eEPC diagrams or Petri nets [2]. Whereas these sort of approaches were built for the purpose of discovering process knowledge that enabled organization optimization, we are targeting the use case of pro-active information delivery. As presented in Holz et al. [3], such approaches mainly vary in two dimensions: process support and information delivery, meaning that some information delivery approaches work on top processes ranging from ad-hoc to strictly-structured on one dimension, and information delivery mechanisms ranging from light-weight to heavy-weight. In our domain, having light-weight modeling is important to simplify deployment and adaptation to various knowledge worker use cases. Our design goal is to maximize information delivery performance while learning from usage logs alone, without any user supervision. Besides enterprise knowledge processes, systems that learn while monitoring were also proven successful in the e-learning domain [4] with work-integrated learning [5].

3 Knowledge Process Model Framework

We have designed and implemented a framework for prediction on top of knowledge processes that solve issues that we have encountered with designing recommendation use cases on top of classic process mining models. Process mining expects pre-defined atomic actions as its input – each event unambiguously representing an action. This requirement may not be easily satisfied in some domains, such as knowledge work [6], where a lot of activity is ad-hoc and does not follow a prescribed process. The data that we are dealing with is a natural example of the *TNT* (text, network, time) framework [8]: the data points are events, carrying temporal information, content and a social network component. To fit into that model, we use a framework that is designed to handle this domain to support process mining on top of semi-structured data [7].

We decompose the information delivery problem in knowledge processes into a framework with three separate subproblems, each solveable with multiple approaches. To better illustrate the design decision behind the three distinct steps, let us start from the final step: the *process model.* In business process modeling, a process model describes patterns of atomic actions and is able to provide us with a probability estimate that one action will follow another. In other word, a process model maps from an action history to a probability distribution across possible following actions.

Given that our input is in form of primitive events, we need to transform the TNT events into actions – using an *action model*. However, directly constructing an action model to map from TNT events to process actions is not always possible nor practical, since the content within the events may not have the same properties across different domains. For this purpose we define the *event model*, which represents the feature construction phase from the raw TNT events.

The prediction scenario uses all of these three steps in the following setting: given a user's history of TNT events and possible resources that will be used, transform both the history and the future candidates into actions. Next, use the process model to evaluate each candidate's probability of appearing given the observed history. This information is then used in evaluation as a ranking score for information resources.

3.1 Feature Construction with Event Models

TNT events may contain the information on the actor, textual content, social network information, event metadata and the time of execution. We partition the log of events into sessions, which represent instances of knowledge process executions.

A **vector space event model** is represents the TNT event as a vector of features, derived from the event's text, social network and metadata. For text-derived features, we apply a TF-IDF weighing scheme.

This model can be extended by also encoding the information from neighbouring events. The sequential addition to the simple vector space event model enables event representations that captures some temporal dependencies, resulting in a **session vector space** model, which takes into account the fact that another has been executed in the same session within a given window.

3.2 Action Models

As basic process models still map from a space of action-to-action transitions to conditional probabilities, we still need to map sequences of documents into sequences of actions.

Independent Feature Actions. There are several ways to avoid high dimensionality of actions: one is to treat event features independent and consider them as individual actions themselves. However, this leads us to an issue where we have multiple possible actions per event, which we need to interpret. We solve this in the following fashion: given a pair of consecutive event vectors, consider at each possible feature pairing from these two vectors as a potential sequence. For instance, given a sequence of two consecutive events $d_1 = \{a = 0.7, b = 0.7\}$ and $d_2 = \{x = 0.7, y = 0.7\}$, we translate this to action sequences of two consecutive actions: (a,x), (b,x), (a,y) and (b,y). In other words, each of these action sequences is a possible interpretation of (d_1,d_2) in this action model. Since the operation considers all possiblities, we only consider the immediate neighboring event. The consequence of this approach is that for some concrete dataset, the feature space of events is the same as the feature space of actions. This is one way to model the fact that an event may have many features. This model provides a tradeoff that keeps the space of possible actions reasonably low-dimensional, but assumes conditional independence of features.. Where multiple actions may represent a single event. To resolve this, we choose the most likely interpretation.

Clustering-Based Actions. As has been suggested in [6], one way to define actions is by performing various types of clustering on the event feature vectors while still retaining high predictive power [7]. This model allows us to control the dimensionality of the action space by varying the *k* parameter in clustering. Upon mapping events into actions, we classify the new set of events into actions via a centroid classifier.

3.3 Process Models

The core of the framework are the process models. They estimate the probability that a certain action will take place conditioned by the last couple of observed actions.

Smoothing of the process model is necessary because sparseness of data. In order to successfully compute a probability of a sequence, none of the sub-sequence probabilities must turn out to be zero. However, if there is no evidence for a particular action following a particular sequence, it does not mean that that sequence will also not occur in test data. For that purpose we employ Laplace add-one smoothing.

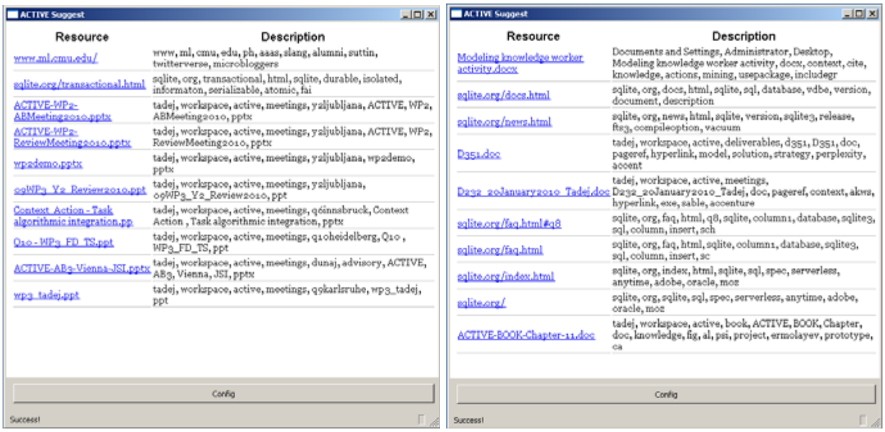
Even though the framework allows any possible combination of event, action and process models, some combinations yield better results than others. This framework enables two different ways to encode dependencies in the knowledge process: either via features in sequential- and session-based event models, or explicitly within the core process model that models conditional probabilities between successive actions. We allow and experiment on both scenarios, since interesting dependencies might exist either on the level of individual features, or between actions, having a higher level of abstraction.

4 Implementation

We have implemented prediction that fits into the desktop environment of knowledge workers. The scenario is the following: imagine a desktop widget that tracks the current state of the user's workspace with most recently accessed resources. The widget itself is rendered as a ranked list of information objects that are considered most relevant for the user. We use the following semantic properties as features for individual events: bag-of-words of document content, social roles of participants (inside vs. outside of organization, manager, developer, researcher, private vs. multiple people, single vs. multiple organizations) and event metadata (type of event, type of media).

For cases where the user has just started his session and does not have any recently accessed resources or there is are no possible predictions from the process model, we use multiple criteria. We ranking using the following criteria:

- The probability of the candidate information resource d_i given the history: $P(d_i|d_{hist\,0}..., d_{hist\,j})$;
- The similarity of the candidate information resource to the most recently observed information resource: $sim(d_i,d_{hist\,j})$;



Resource	Description
www.ml.cmu.edu/	www, ml, cmu, edu, ph, aaas, slang, alumni, suttin, twitterverse, microbloggers
sqlite.org/transactional.html	sqlite, org, transactional, html, sqlite, durable, isolated, informaton, serializable, atomic, fai
ACTIVE-WP2-ABMeeting2010.pptx	tadej, workspace, active, meetings, y2ljubljana, ACTIVE, WP2, ABMeeting2010, pptx
ACTIVE-WP2-ReviewMeeting2010.pptx	tadej, workspace, active, meetings, y2ljubljana, ACTIVE, WP2, ReviewMeeting2010, pptx
wp2demo.pptx	tadej, workspace, active, meetings, y2ljubljana, wp2demo, pptx
09WP3_Y2_Review2010.ppt	tadej, workspace, active, meetings, y2ljubljana, 09WP3_Y2_Review2010, ppt
Context_Action - Task algorithmic integration.pp	tadej, workspace, active, meetings, q6innsbruck, Context Action , Task algorithmic integration, pptx
Q10 - WP3_FD_TS.ppt	tadej, workspace, active, meetings, q10heidelberg, Q10 , WP3_FD_TS, ppt
ACTIVE-AB3-Vienna-JSI.pptx	tadej, workspace, active, meetings, dunaj, advisory, ACTIVE, AB3, Vienna, JSI, pptx
wp3_tadej.ppt	tadej, workspace, active, meetings, q9karlsruhe, wp3_tadej, ppt

Resource	Description
Modeling knowledge worker activity.docx	Documents and Settings, Administrator, Desktop, Modeling knowledge worker activity, docx, context, cite, knowledge, actions, mining, usepackage, includegr
sqlite.org/docs.html	sqlite, org, docs, html, sqlite, sql, database, vdbe, version, document, description
sqlite.org/news.html	sqlite, org, news, html, sqlite, version, sqlite3, release, fts3, compileoption, vacuum
D351.doc	tadej, workspace, active, deliverables, d351, D351, doc, pageref, hyperlink, model, solution, strategy, perplexity, accent
D232_20January2010_Tadej.doc	tadej, workspace, active, meetings, D232_20January2010_Tadej, doc, pageref, context, akws, hyperlink, exe, sable, accenture
sqlite.org/faq.html#q8	sqlite, org, faq, html, q8, sqlite, column1, database, sqlite3, sql, column, insert, sch
sqlite.org/faq.html	sqlite, org, faq, html, sqlite, column1, database, sqlite3, sql, column, insert, sc
sqlite.org/index.html	sqlite, org, index, html, sqlite, sql, spec, serverless, anytime, adobe, oracle, moz
sqlite.org/	sqlite, org, sqlite, sql, spec, serverless, anytime, adobe, oracle, moz
ACTIVE-BOOK-Chapter-11.doc	tadej, workspace, active, book, ACTIVE, BOOK, Chapter, doc, knowledge, fig, al, psi, project, ermolayev, prototype, ca

Fig. 1. Examples of recommendations for various situations in the knowledge process

The implemented front-end is a graphical user interface, displaying an ordered list of most probable information objects given the current state of the user, as shown in Figure 1. Given the live events coming from the monitoring infrastructure, the information delivery framework converts these events into feature vector documents and then action sequences which represent the user's session history. Then, it selects a set of possible interesting information resources with standard text retrieval techniques and evaluates the probability of each known information object given the observed history using the knowledge process model recommender.

Figure 1 shows some examples of recommendations, offered by the service. The first image shows the state of the recommender after reading a documentation of a back-end system for process mining, which is reflected by similar content types being offered. The second image demonstrates another point in the knowledge process, showing recommendations after opening a technical presentation on machine learning. It offers various related documents, as well as web sites that might contain useful information in that situation which have been accessed in similar situations.

5 Experiments

In our experiments, we have performed evaluation by constructing combinations of models with one set of data and evaluating them with another subset using ten-fold cross-validation to account for generalization error. We ran the experiment ten times, using nine partitions as training data and the remaining one as a test dataset. Given this setup, we can observe several evaluation metrics, relevant to evaluation of models. The dataset of 31882 events was gathered by monitoring three knowledge workers in a telecommunications company.

We evaluated various knowledge process model configurations by measuring how high do they rank the actual events in the test set. In order to conduct this experiment, we simulate the prediction behaviour by querying the system with partial sequences to which we already know the correct TNT event that will follow and then check how high does that particular event rank in the prediction output. In other words, we are measuring how well can this system predict the actual workflow. We then report the average reciprocal rank for a particular combination of models and average number of predictions that score in the top 20, which are the key indicators in our experiment.

We have evaluated on the knowledge worker logs. Since the TNT events have no pre-defined actions, we use the vector space event models – *IDF* and *SessionIDF*. They all use the TF-IDF weighing scheme for features, but differ in the way that the features are generated: whereas *IDF* contains only the features from a particular event, *SessionIDF* also encodes between the features of the events in the same session. For process models, we experimented on either using an identity transform which assumes uniform processes and acts as a baseline without a process model or a bi-gram Markov model with Laplace smoothing, modeling the dependencies between successive events.

Table 1. Experiment results with various combinations of event, action and process models for the desktop use case

Event Model	Action Model	Process Model	Reciprocal rank	Percentage in top 20
IDF	Independent	None	0.0612	0.2220
IDF	Independent	Laplace	0.0803	0.2377
IDF	Clustered:10	None	0.0794	0.2697
IDF	Clustered:10	Laplace	**0.1076**	**0.3485**
IDF	Clustered:30	None	0.0853	0.3081
IDF	Clustered:30	Laplace	0.0797	0.2490
SessionIDF	Independent	None	0.0774	0.2895
SessionIDF	Independent	Laplace	0.0750	0.2674
SessionIDF	Clustered:10	None	0.0756	0.2807
SessionIDF	Clustered:10	Laplace	0.0701	0.2384
SessionIDF	Clustered:30	None	0.0832	0.3013
SessionIDF	Clustered:30	Laplace	0.0874	0.3051

We have ran the experiments on the dataset, gathered in two separate two-month time periods, totaling 31882 events. Here, the events consist of web browsing events, editing of Microsoft Office documents and working with e-mail. Results, shown in Table 1, show the following: best performing configuration is the one with plain TF-IDF features of events, defining a clustered action model with few distinct clusters as actions and a process model with Laplace smoothing. All in all, we are able to place the correct information resource in the top-20 list roughly on over one third of occasions.

6 Conclusion

An important lesson is that many of the issues in process mining lie in transforming the input data into well-defined actions. By developing a framework that integrates the data transformation steps within the mining process itself, we can simultaneously solve the issue of transforming events into atomic actions, as well as learning process models, reducing the preprocessing requirements for implementing a predictive application. We have discovered that encoding the knowledge process patterns in the feature vectors using the session-based feature construction is a valid approach, showing significant improvement from the baseline while having a very simple implementation. However, our use case shows that using an explicit process model can ultimately outperform session-based features in events. Results show that assuming independence of features does not provide desirable performance.

Implementation-wise, we have produced a simple user interface that does not require that the user interacts with it in any special way to train it, lowering the barrier to practical usage. For our future work, the clustering-based action definition will include using a complex graph representation of data so that we can avoid flattening the semantic network structure into event features. To take advantage of the structural information and to correctly handle differences in distributions across people, events and resources, we will represent the data in a relational representation and employ multi-relational clustering algorithms which are able to handle such representations. We will also experiment in applying this model in other specific knowledge worker domains, focusing on supporting multiple knowledge workers with personalized models. Future work on refining recommendation will also focus on constructing more personalized models by constructing a general model combined with a personalization layer. In terms of incorporating external knowledge, we will consider semi-supervised learning from user feedback, as well as employing pre-existing process models in combination with learned process models. Further evaluation will also consider sensitivity of a recommendation. In real scenarios, the benefit of a recommendation may be negated by the interruption of the user's workflow.

Acknowledgments. This work was supported by the Slovenian Research Agency and the IST Programme of the EC under ACTIVE (IST-2008-215040), ALERT (ICT-249119-STREP) and PASCAL2 (IST-NoE-216886).

References

1. Warren, P., Kings, N., Thurlow, I., Davies, J., Bürger, T., Simperl, E., Ruiz, C., Gomez-Perez, J., Ermolayev, V., Ghani, R., Tilly, M., Bösser, T., Imtiaz, A.: Improving knowledge worker productivity the ACTIVE approach. BT Technology Journal 26(2) (2009)
2. van Dongen, B.F., de Medeiros, A.K.A., Verbeek, H.M.W(E.), Weijters, A.J.M.M.T., van der Aalst, W.M.P.: The ProM Framework: A New Era in Process Mining Tool Support. In: Ciardo, G., Darondeau, P. (eds.) ICATPN 2005. LNCS, vol. 3536, pp. 444–454. Springer, Heidelberg (2005)
3. Holz, H., Maus, H., Bernardi, A., Rostanin, O.: From lightweight, proactive information delivery to business process-oriented knowledge management. Journal of Universal Knowledege Management 2, 101–127 (2005)
4. Lokaiczyk, R., Faatz, A., Beckhaus, A., Goertz, M.: Enhancing just-in-time e- Learning through Machine Learning on Desktop Context Sensors. In: Kokinov, B., Richardson, D.C., Roth-Berghofer, T.R., Vieu, L. (eds.) CONTEXT 2007. LNCS (LNAI), vol. 4635, pp. 330–341. Springer, Heidelberg (2007)

5. Rath, A.S., Devaurs, D., Lindstaedt, S.N.: Studying the Factors Influencing Automatic user Task Detection on the Computer Desktop. In: Wolpers, M., Kirschner, P.A., Scheffel, M., Lindstaedt, S., Dimitrova, V. (eds.) EC-TEL 2010. LNCS, vol. 6383, pp. 292–307. Springer, Heidelberg (2010)
6. Štajner, T., Mladenić, D.: Modeling Knowledge Worker Activity: Workshop on Applications of Pattern Analysis, Cumberland Lodge (2010)
7. Štajner, T., Mladenić, D., Grobelnik, M.: Exploring Contexts and Actions in Knowledge Processes. In: Proceedings of the 2nd Workshop on Context, Information and Ontologies (2010)
8. Grobelnik, M., Mladenić, D., Ferlež, J.: Probabilistic Temporal Process Model for Knowledge Processes: Handling a Stream of Linked Text. In: Proceedings of SiKDD 2009 Conference on Data Mining and Data Warehouses (2009)
9. Gomez-Perez, J., Grobelnik, M., Ruiz, C., Tilly, M., Warren, P.: Using task context to achieve effective information delivery. In: Proceedings of the 1st Workshop on Context, Information and Ontologies, pp. 1–6. ACM (2009)

Service Adaptation Recommender in the Event Marketplace: Conceptual View

Yiannis Verginadis[1], Ioannis Patiniotakis[1],
Nikos Papageorgiou[1], and Roland Stuehmer[2]

[1] Institute of Communications and Computer Systems,
National Technical University of Athens,
{jverg,ipatini,npapag}@mail.ntua.gr
[2] FZI Forschungszentrum Informatik, Karlsruhe, Germany
stuehmer@fzi.de

Abstract. In this paper, we present the conceptual architecture of a highly scalable federated platform for processing vast number of events coming from distributed sources, in order to detect interesting situations that lead to service adaptation recommendations. The approach presents an Event Marketplace, a platform for mediating between event providers and (complex) event consumers in very large and heterogeneous environments which is enriched with timely reaction capabilities, through situational-driven service adaptations. We focus on an approach for detecting real-time interesting situations that lead to service adaptations.

Keywords: Event Marketplace, Distributed Systems, Service Adaptation.

1 Introduction

Adaptation as one of the basic phenomena of biology is the evolutionary process whereby a population becomes better suited to its habitat [1], [2]. The notion of adaptation has been extensively used, especially nowadays, in the computer science domain. It is considered as one of the most desired functionalities of today's highly dynamic, distributed and ubiquitous environments in the service-oriented setting. The need for highly flexible services that can be orchestrated in order to provide certain behaviors and at the same are able to react and adapt inside their "habitat" (i.e. change based on the context and events that formulate the imprinting of a highly dynamic service environment), is considered to be a desired but difficult to achieve fact.

The recent expansion of the "everywhere" deployment of the wireless sensors networks (part of the Internet of Things) introduces the information from stationary or moving objects in the global service adaptation task. Let us consider the following scenario as an example of ubiquitous interaction between services resulting in a more personalized and adapted service execution:

"Paul is a businessman who has been flying from Paris to New York. He used the entertainment service on board, but hasn't finished watching the movie before the landing. Two hours later he is entering his room in the downtown hotel he booked earlier and wow: the room entertainment service is ready to play the movie Paul was

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 194–201, 2012.

watching in the plane – of course only the unfinished part.” Such a scenario involves the ubiquitous interaction between services, the processing of dispersed events and the actual adaptation of service execution in a way that satisfies the customer. All these constitute multidimensional problems. Let us name just a few challenges. First of all, such an interaction cannot be modeled in the design time, since it is not possible to predict all interesting interactions in advance. Secondly, a common understanding between different actors in a heterogeneous environment is requested. Next, it is challenging to ensure a proper (on time, complete and relevant) delivery of interesting information in a large scale distributed environment.

In this paper we present the concept and the architecture of a platform that can satisfy these requirements (section 2). The platform uses its cloud-computing nature to be “elastic” and operate in the “pay as you go” mode. We can consider the platform as a kind of the Event marketplace (similar to the notion of the service marketplace) where events coming from different producers can be arbitrary combined by different event consumers. In addition, we focus and give details about a core part of this platform that recommends changes (adaptations) of services’ configurations, composition or workflows, in order to overcome problems and achieve higher performance by reacting to real time events (i.e. formulating an interesting situation) at the right time and in the right way (section 3). In section 4, we present the related work of our approach, whereas section 5 contains concluding remarks.

2 Proposed Conceptual Architecture

The initial conceptual architecture for our platform is depicted in Figure 1. In this section we introduce the components and present their main functionalities.

The Distributed Service Bus (DSB) provides the service oriented architecture (SOA) and event driven architecture (EDA) infrastructure for components and end user services. It acts as the basis for service deployments, and processes (BPEL, BPMN), routing synchronous and asynchronous messages from services consumers to service providers. Based on the principles of the system integration paradigm of Enterprise Service Bus the DSB is distributed by nature.

The Governance component allows users to get information about services and events, as well as specifying QoS requirements as service level agreement (SLA) contracts using the WS-Agreement standard [3]. The Governance component extends a standard Service-based governance tool (OW2 Petals Master [4]) by adding governance mechanisms for event-based systems. Its role is to provide ways to govern services and events. It provides standards-based APIs and a graphical user interface.

The Event Cloud provides storage and forwarding of events. The role of the Event Cloud is a unified API for events, real-time or historic. To that end, it contains a peer-to-peer network to store histories of events durably in a distributed fashion. In the same way the list of subscribers is distributed across the peers to notify subscribers if a given new event is stored at any node and a corresponding matching subscription exists in the system. Subscriptions may use a simple set of operators such as conjunctive queries which can be evaluated efficiently on a single peer. More complex queries are executed in the DCEP component.

The DCEP component (Distributed Complex Event Processing) has the role of detecting complex events and reasoning over events by means of event patterns defined in logic rules. To detect complex events, DCEP subscribes to the Event Cloud for any simple event defined in the event patterns at a given point in time. DCEP supports traditional event operators such as sequence, concurrent conjunction, disjunction, negation etc., all operators from Allen's interval algebra [5] (e.g., during, meets, starts, finishes etc.), window operators, filtering, enrichment, projection, translation, and multiplication. Out-of-order event processing is supported (e.g. events that are delayed due to different circumstances such as network anomalies).

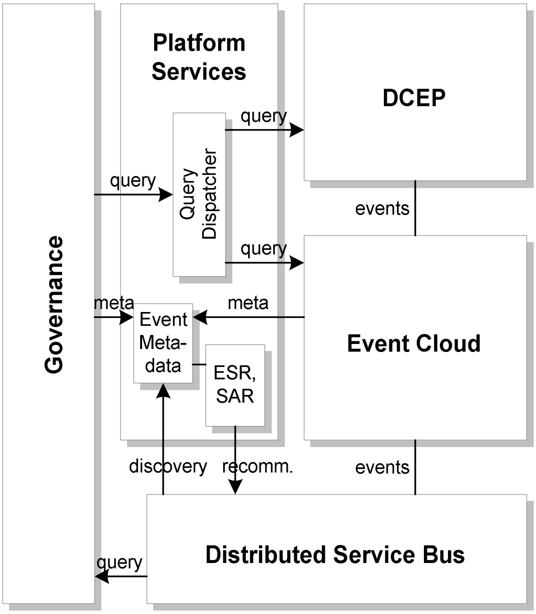


Fig. 1. Conceptual Architecture

The Platform Services component incorporates several functional additions to the platform as a whole. The Query Dispatcher has the role of decomposing and deploying user subscriptions in pieces supported by the Event Cloud and DCEP respectively, taking into account the expressivity supported by the two target components. The Event Metadata component stores information about events, such as source descriptions, event type schemas, etc, to enable the discovery of relevant events for an event consumer and to provide data to the subscription recommender. The ESR and SAR components form the Event Subscription Recommender (ESR) and Service Adaptation Recommender (SAR). ESR will recommend or perform dynamic subscriptions for complex events based on situations detected from semantic-enabled events and the context of subscribed services. Thus, ESR will provide assistance to services that will have the option to be subscribed to specific events at the "right time" without the services having complete knowledge about the supply in the marketplace at a given time. SAR is thoroughly discussed in the next section of this paper.

3 Service Adaptation Recommender (SAR)

The objective of SAR is to suggest service administrators, changes (adaptations) of their services' configurations, composition or workflows, in order to overcome problems or achieve higher performance. Based on recognized situations, SAR will be able to define adaptation pointcuts (points in a service flow that need to be adapted as a reaction to a certain situation) and advices (what to adapt and how based on a number of service adaptation strategies). In this section, we present technical requirements that apply for Service Adaptation Recommender (SAR) software component along with its initial conceptual view (figures 2 and 3).

One of the basic capabilities SAR feature is the situation awareness and detection functionality. This corresponds to the ability to sense situations relevant to service objectives and operation and the ability to track transitions between situations, by processing events and contextual service information. We consider complex events, detected in real time, as a way to signify situations that may require adaptation and we plan to enrich them with contextual information for defining dynamically what modifications are needed. Based on situational awareness module, SAR will be able to make intelligent recommendations for service adaptations (figure 2). It will improve service performance by detecting problems that need to be resolved (e.g. underperforming services, or suboptimal service workflows for the given situation) and by providing adaptation advices to be implemented in the appropriate workflow places (e.g. service tasks that need reordering, alteration or substitution etc.).

We have defined some general requirements that apply here. Firstly, it is important that SAR will be able to register for simple events as well as for complex events that carry combined information (e.g. the last 20 minutes, there is a 5% radiation increase). Both simple and complex events need to carry semantics that will allow further processing and reasoning. The federated middleware of our platform must also guarantee that SAR will receive all events that they have been subscribed to. Failure to deliver an event to SAR may lead to the non-detection of a new situation. For reasoning purposes domain knowledge is needed, which may be comprised of an event ontology, a context and a situation model. Scalability is another general requirement that applies to SAR, as the ability to cope with and extract valuable information from a "burst" of events, is imperative for detecting interesting situations. Finally, since our platform will be federated, it is important to detect global situations across the several service busses (i.e. Distribute Service Bus).

Since, SAR will be the dedicated software component to suggest changes (adaptations) of services' configurations, it needs to leverage domain knowledge for processing contextual information (e.g. preferences) of services and for analyzing service composition information (Fig. 2 - "Service Analyzer"). SAR also needs the ability to comprehend semantics carried by events, reasoning event semantics with service information, in order to detect relevant situations or trigger situational transitions (Fig. 2 - "Situation Awareness Module"). By acquiring all of the above it will detect and define appropriate service adaptation solutions as timely answers to situations that dictate reactions (Fig. 2 – "Adaptation Recommender"). Regarding DSB, SAR needs subscription/unsubscription capabilities from event sources, in response to situation transitions, as well as query capability to event storage for past events and historical service data in order to identify similar past cases ("Collaborative Filter").

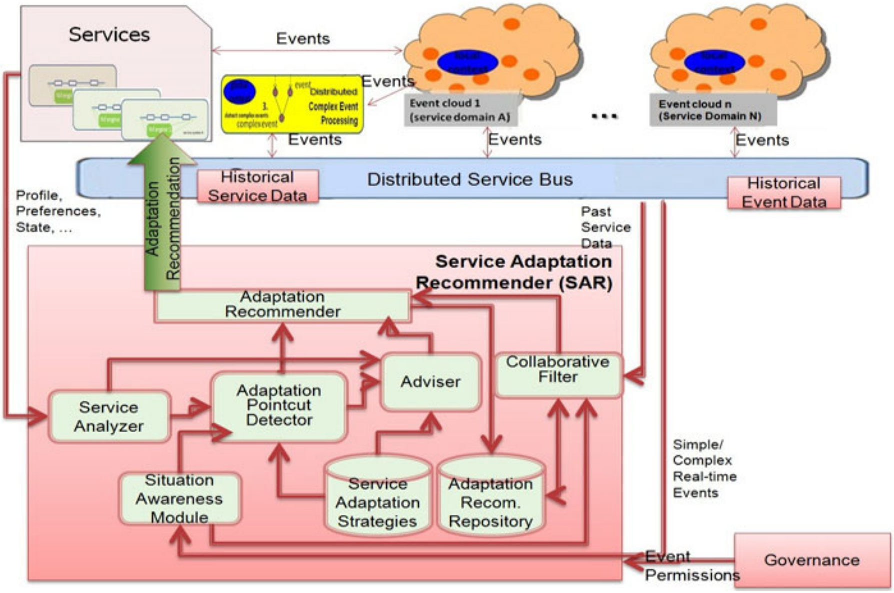


Fig. 2. SAR: Conceptual View

Service Adaptation Recommender (SAR) is considered as a core part of our platform and aims to provide recommender services for reacting at the right time and in the right way. Service Adaptation Recommender (SAR) will suggest changes of services configurations in order to overcome problems or improve service performance by:

- retrieving and analyzing service preferences/context (“Service Analyzer”)
- analyzing service composition information (“Service Analyzer”)
- deducing the situational model for the service using its semantics, composition, and preferences (“Situation Awareness Module”)
- defining where in process flow we might need an adaptation (“Adaptation Pointcut Detector”)
- evaluating and defining several alternative solutions of adaptation based on service adaptation strategies (“Adviser”)
- identifying and recommending and storing service adaptations as reaction to a current situation (“Adaptation Recommender”)
- exploiting collective intelligence by deriving adaptation recommendations based on previous ones in similar situations (“Collaborative Filter”)

Service Adaptation Recommender (SAR) will be Java based software component that will depend on the events acquired from our federated platform along with their semantics and the contextual information gathered from the related services. SAR’s recommendations may include: removal of a problematic service, replacement of an underperforming service, addition of a new service, alteration of the process flow. Specifically, based on recognized situations, SAR will be able to define adaptation pointcuts (points in a service flow that need to be adapted) and advices (what to adapt and how).

In figure 3 a number of provided and required interfaces are given for SAR. The provided interfaces include setting and/or retrieving permissions to resources (events) for each service, receiving events from Event Cloud (this is the callback interface for Event Cloud) and sending adaptation recommendations to service administrator. The required interfaces include subscribing to / unsubscribing from events from the Event Cloud, querying Event Cloud for past / stored events and retrieving service preferences.

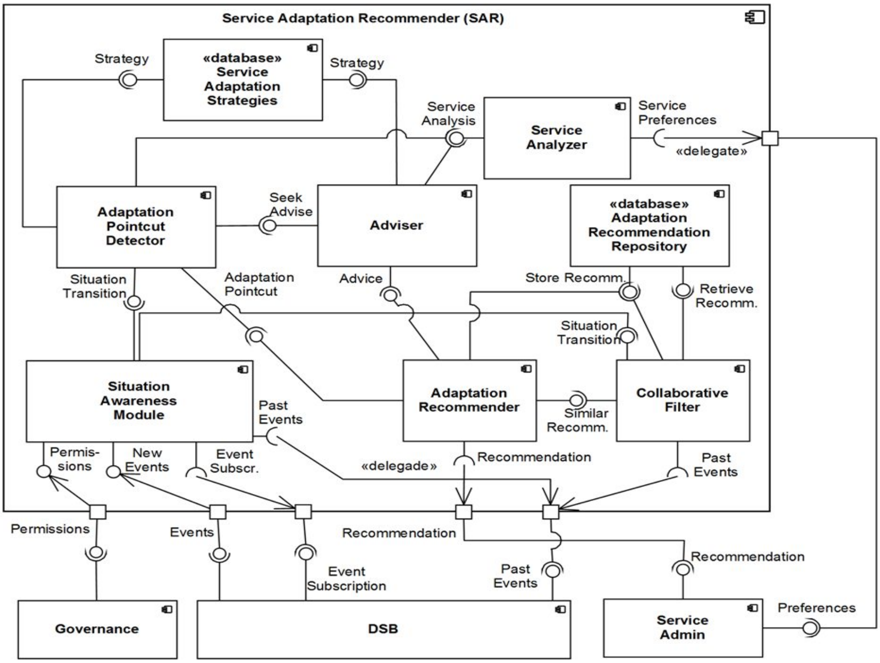


Fig. 3. SAR's Component Diagram

4 Related Work

Recently, there has been a significant paradigm shift towards real-time computing. Previously, queries against databases and data warehouses were concerned with looking at what happened in the past. On the other hand, complex event processing (CEP) is concerned with processing real-time events, i.e., CEP is concerned with what has just happened or what is about to happen in the future. Complex event processing is a very active field of research and is being approached from many angles [6]. Distributed complex event processing approaches circumvent resource limitations by taking advantage of pre-existing (shared) or dedicated network infrastructure. The Padres system [7] is such a distributed approach relying on a pre-existing network of brokers. A broker may be an end-point for publishers (event sources) and subscribers (event sinks) to access the network. More importantly it is the task of the brokers to match events to the available subscriptions. The set of all interconnected brokers forms an overlay network across the underlying infrastructure. A drawback of Padres is the event pattern language used in its subscriptions. It is limited to handle only key-value maps for events and the set of event operators is very limited, for example, temporal relationships between events cannot be expressed declaratively but must be expressed on a timestamp-arithmetical level.

S-Cube [8] as the most prominent Network of Excellence in service adaptation, points out the evolution and adaptation methods and tools as keys to enable service-based applications (SBAs). Following the S-Cube's terminology, the term adaptation refers to the modification of a specific instance of a system during run-time (e.g. re-execution of a unavailable service or a substitution of a unsuitable service). Nowadays, several efforts that try to cope exactly with this issue point to Aspect-Oriented Programming (AOP), as a novel way to weave alternative actions in business processes at run-time. AOP has been proposed as a technique for improving the separation of concerns in software systems and for adding crosscutting functionalities without changing the business logic of the software. One of the most recognizable approaches for service adaptations using AOP is the work in [9] where the AO4BPEL is introduced. AO4BPEL is an XML-based language that creates a wrapper around the BPEL and has the ability to weave aspects at runtime to business processes. Aspects consist of one or several pointcuts and advices. AO4BPEL is based on XPath [10], which is used to select activity join points (i.e., points corresponding to the execution of activities) and internal join points (i.e., points inside the execution of activities such as the point where the outgoing message of an invoke activity is generated). An advice is the new behaviour to be included at a join point and contains the new code to be executed. SAR will use and extend such AOP based approaches for semantically and dynamically weaving changes in service based systems.

5 Conclusions and Future Research

We presented a novel approach for large scale, context-driven and quality-aware distributed event processing and we focused in the conceptual view of a service adaptation recommender for coping with the challenges of rapidly changing distributed service based systems. We are currently working on implementing the main functionalities of SAR, presented in this paper. The backbone mechanism that will support the core of SAR component will be based on a new modelling and execution framework for situation-aware applications. This framework will be built around the notion of goal-driven and hierarchical Situation-Action-Networks (SANs) which will provide modelling primitives for goals, situations, context, actions and loose mappings of abstract situations to generic action pools. This loose mapping at design time will allow for real situation-driven service adaptation recommendations for SBAs, at run time. The work in SANs actually extends well known research efforts in the planning domain (e.g. Hierarchical Task Networks [11], Behavioural trees [ref], etc.)

This research has been performed in the scope of a research project the vision of which is to develop and validate an elastic and reliable architecture for dynamic and complex, event-driven interaction in large highly distributed and heterogeneous service systems. Such architecture will enable ubiquitous exchange of information between heterogeneous services, providing the possibilities to adapt and personalize their execution, resulting in the so-called situational-driven adaptivity. We plan to test and validate our platform and specially its situational driven reaction capabilities, in a nuclear crisis management scenario and in a smart taxi service system. Both these use cases are characterized by dispersed event sources and are considered to create dynamically changing environments that dictate for distributed event processing and service adaptations.

Acknowledgements. This work has been partially funded by the European Commission under FP7 project PLAY. The authors would like to thank the project team for comments and suggestions.

References

1. Martin, E.A.: The Oxford Dictionary of Science, 6th edn. Oxford University Press (2010); ISBN-13: 9780199561469
2. Williams, G.C.: Adaptation and natural selection: a critique of some current evolutionary thought. Princeton Univ. Press, Princeton (1966)
3. Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S., Xu, M.: Web Services Agreement Specification (WS-Agreement) (2004), `http://www.gridforum.org/Meetings/GGF11/Documents/draft-ggf-graap-agreement.pdf`
4. Petals Master SOA Governance Solution, `http://petalsmaster.ow2.org/`
5. Allen, J.F.: Maintaining knowledge about temporal intervals. Commun. ACM 26(11), 832–843 (1983)
6. Luckham, D.C.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley Longman Publishing Co., Inc., Boston (2001)
7. Fidler, E., Jacobsen, H.-A., Li, G., Mankovski, S.: The padres distributed publish/subscribe system. In: 8th International Conference on Feature Interactions in Telecommunications and Software Systems, pp. 12–30 (2005)
8. S-Cube, `http://www.s-cube-network.eu/`
9. Charfi, A., Mezini, M.: Ao4bpel: An aspect-oriented extension to bpel. World Wide Web 10(3), 309–344 (2007)
10. XML Path Language (XPath), `http://www.w3.org/TR/xpath/`
11. Nau, D., Au, T.-C., Ilghami, O., Kuter, U., Muñoz-Avila, H., Murdock, J.W., Wu, D., Yaman, F.: Applications of SHOP and SHOP2. IEEE Intelligent Systems 20(2) (2005)
12. Lim, C.U., Baumgarten, R., Colton, S.: Evolving Behaviour Trees for the Commercial Game DEFCON. Applications of Evolutionary Computation, 100–110 (2010)

Authoring and Publishing Units and Quantities in Semantic Documents

Mihai Cîrlănaru[1], Deyan Ginev[1], and Christoph Lange[1,2]

[1] Computer Science, Jacobs University Bremen, Germany
{m.cirlanaru,d.ginev,ch.lange}@jacobs-university.de
[2] Universität Bremen, Germany

Abstract. This paper shows how an explicit representation of units and quantities can improve the experience of semantically published documents, and provides a first authoring method in this respect. To exemplify the potential and practical advantages of encoding explicit semantics regarding units w.r.t. user experience, we demonstrate a *unit system preference* service, which enables the user to choose the system of units for the displayed paper. By semantically publishing units, we obtain a basis for a wide range of applications and services such as *unknown unit lookup*, *unit and quantity semantic search* and *unit and quantity manipulation*. Enabling semantic publishing for units is also presented in the context of a large collection of legacy scientific documents (the arXMLiv corpus), where our approach allows to non-invasively enrich legacy publications.

1 Motivation

Units and quantities[1], although widely spread, lack a formal standard representation for semantic publishing. A multitude of problems [45] arise from the different flavors (country specific unit standards) and formats (abbreviations, special cases of occurrence) of units, making it hard for the untrained reader to fully understand the information provided. Semantic publishing solves most such problems by disambiguating the unit and quantity occurrences, thus enabling a wide range of applications and services to interact with them.

A **unit** is *any determinate quantity, dimension, or magnitude adopted as a basis or standard of measurement for other quantities of the same kind and in terms of which their magnitude is calculated or expressed* [32], but from the top-most level of perception, it simply provides information on a wide range of quantifiable aspects. Concrete examples for the great extent of units and quantities include cooking recipes, medical prescriptions, scientific papers and many other. Semantic publishing can provide the middle layer that would ensure an (automated) way of identifying and understanding these occurrences, which can enable the evolution of useful technologies and services.

[1] We chose the *units and quantities* wording in order to emphasize the semantic dependence between the *unit* and its *quantity* (i.e. amount, magnitude). An in-depth analysis of the two concepts is provided in section 2 of [9].

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 202–216, 2012.

At the perception level, aside from quantifying properties and relations between objects, units bring the meaning of scale. Moreover, units have allowed scientists to better transmit and exchange knowledge among themselves.

In real life, the misinterpretation of units and their quantities has often caused accidents with harsh/expensive consequences. Consider losing a $125 million satellite [30] because of the differences between metric and imperial unit systems, or running out of fuel in mid-flight with an aircraft whose fuel sensors were faultily configured in displaying the units [2]. Fields like medicine, commerce, civil engineering have also been marked by such types of errors and pitfalls [45].

Providing semantics to units and their quantities for the publishing industry, either by supplying semantic authoring tools or by semantically enriching their occurrences in legacy documents, has high impact benefits. It will enable transparent exchange of scientific knowledge between different academic communities, typical of technical papers with a high occurrence of units and quantities, and also enhance the reader's experience, via novel interactive services with day-to-day published material, e.g. cooking recipes or technical manuals.

In the following sections, we introduce preliminaries (section 2), outline our approach to semantic units and quantities, and review relevant state of the art (section 4). That provides a basis for presenting *unit and quantity interaction services* (section 6). We outline immediate strategies (section 5) for extending the benefits of semantic units to legacy documents (section 7) and conclude with a summary of our mid-term outlook of future work (section 8).

2 Preliminaries

The core of semantic publishing resides in open and standardized markup languages used to encapsulate semantics. OpenMath and *Content* MathML are the most widely used semantic markup (also called "content markup") languages for mathematical expressions, which are ubiquitous in science and engineering.

2.1 OpenMath and Content MathML

OpenMath [7] and the semantically equivalent Content MathML [4] are standards for the representing the semantics of mathematical expressions [28] – as annotations to visual renderings, or for the purpose of communication between computational services. Our investigations focus on these two languages.[2]

Structurally, both OpenMath and MathML provide a valuable basis for machine processing of mathematical expressions; that makes them suitable for

[2] The prevalence of XML-based semantic markup languages for representing mathematical expressions – as opposed to RDF – has historical reasons but is also due to the complex n-ary and ordered structures of mathematical expressions, which are hard to break down into RDF triples, be it standoff RDF markup or embedded RDFa annotations. Both representations have in common that the vocabulary terms (here: functions, operators, sets, constants) are identified by URIs. We refer to [29] for an in-depth treatment.

semantic publishing of units and quantities. The expressivity of MathML, provided by its vocabulary having close to 100 XML elements for mathematical functions and operators [28] and multiple *unit and quantity* representation possibilities [13], and the modularity and extensibility of OpenMath's vocabulary by way of modular ontologies ("Content Dictionaries", abbreviated as CDs), enable the development of applications and services (some of which are discussed in section 6.2) that build upon the semantic publishing of units and quantities.

2.2 The Semantic Publishing Pipeline

Semantic Publishing, conceived as a process, consists of at least three components, namely *authoring*, *publishing* and *interaction*. Usually these processes imply three different groups of contributors – authors, publishers and readers. Incorporating the full publishing lifecycle into a single system, striving for integration and collaboration between the different participants, brings great benefits. In this paper, we take the benefits of the social web for well-established and accepted[3] and focus on the more novel semantic aspects of the publishing realm. To this extent, we develop our work in the context of the Planetary eMath3.0 system [34,27], which provides, on top of a stable, well-established Web 2.0 framework, an architecture for semantic services that interact with semantically annotated mathematical and technical documents.

In our work on units and quantities, we have concentrated on setting the necessary technological foundation, hence building on the languages introduced in section 2.1 to select and enhance the authoring and interaction aspects.

3 Semantic Units – Idea Outline

In order to understand how a semantic representation of units and quantities will integrate with the publishing flow of our framework of choice, one first needs to pinpoint what they comprise and how they are *represented*.

A computational *semantic entity* is an object with explicit *structure*, representable in a machine-understandable form, and denoting a corresponding real-world entity. The denotation is usually encoded via a machine-readable ontology. This definition is directly applicable to semantic units and quantities, which are exactly the machine-readable representations of their physical counterparts.

For the *representation* we choose OpenMath, since it encompasses units through modular ontologies, called Content Dictionaries (CDs) [9]. OpenMath CDs enable extensibility through the creation of new such ontologies that can add new symbols, or simply through the extension of the existing unit ontologies/CDs.

[3] For mathematics, including the mathematical foundations of science and engineering, see, e.g., the PlanetMath free encyclopedia [35] and the Polymath wiki/blog-based collaboration effort [5].

As a running example, we consider a semantic representation of the physical ***quantity*** $\boxed{100\,\text{km/h}}$; one possible OpenMath representation is[4]:

```
<OMA>
    <OMS cd="arith1" name="times"/>
    <OMI>100</OMI>
    <OMA>
       <OMS cd="arith1" name="divide"/>
       <OMA>
         <OMS cd="units_ops1" name="prefix"/>
         <OMS cd="units_siprefix1" name="kilo"/>
         <OMS cd="units_metric1" name="metre"/>
       </OMA>
       <OMS cd="units_time1" name="hour"/>
    </OMA>
</OMA>
```

Listing 1. OpenMath representation of $\boxed{100\,\text{km/h}}$

4 State of the Art

We review the relevant prior work involving units and quantities in the context of semantic publishing. Note that we do not cover the publishing dimension itself; we consider it a stand-alone level within a semantic publishing framework, independent of the processed content.

4.1 Representation

The semantic publishing aspect of units in scientific documents has not yet accumulated a sizable body of prior work. Previous research has mainly been concerned with the standardization of unit and quantity representation, which is far from complete (not covering every unit occurrence possibility) or sufficiently machine comprehensible. There is a number of units-related semantic web ontologies: The authors of the Measurement Units Ontology [6] review a number of ways of representing units in RDF. The SWEET (Semantic Web Earth and Environmental Terminology [43,37]) and QUDT (Quantities, Units, Dimensions and Data Types [21]) ontologies are particularly remarkable for linking units to related machine-comprehensible information. SWEET 2.0 describes just 91 units but comprises 150 modules that cover many different sciences as well as common foundations of science; it links units to the SWEET descriptions of the fields of science where they occur. QUDT 1.1 covers 807 units and links all quantities[5],

[4] This is one out of several ways of representing units (cf. [13]). For a detailed description of the XML schema see section 3.1.2 of [7].

[5] Differing from this paper, QUDT uses the term “quantity” for “an observable property of an object [. . .] that can be measured and quantified numerically”, and uses the term “quantity value” for the numerical value of a quantity [21].

units, and dimensions to their counterparts in the DBpedia dataset [12], where users or automated agents can then explore further relations.

For OpenMath, a representation of units and quantities has been proposed (cf. [13]), and several CDs covering common units have been provided. The in-depth analysis of the prospective representations of units and their dimensions that [13] proposes (taking into account the pros and cons of each approach) allows for a broader view on the multitude of semantic publishing possibilities. The two most significant sets of OpenMath unit CDs have been developed by James Davenport and Jonathan Stratford [38] and Joseph Collins [9], respectively. The former are remarkable for their explicit representation of conversion rules (see also Section 4.4). The latter ones provide a standards-compliant implementation of SI[6] quantities and units, providing strong insight on the concepts of *quantity* and *unit* and on the prospects of capturing more of their semantics in the representation.

4.2 Authoring

In "pre-semantic" environments, such as LaTeX, there are first approximations of content-oriented macros that represent units. A prominent example is the LaTeX package *SIunits* [20] which covers the full range of base and derived units in the SI system, as well as SI prefixes, a range of widely accepted units external to SI and generic mechanisms for creating custom author-specified unit constructs. The package enables a large set of abbreviative commands, which are internally built up from the compositional application of atomic building blocks. In this sense, the authoring process via *SIunits* is *nearly semantic* on the interface level, but *entirely presentational* on the output side.

Still, all major semantic authoring systems (e.g. the semantic LaTeX extensions sTeX [25], SALT [19], the Ontology Add-in for Microsoft Office Word [14], or the semantic content management system PAUX [33]) have so far neglected the specific use case of units. This can be partially explained by the lack of a widely agreed standard representation, as well as different primary development foci – mathematics for sTeX, rhetorical structures for SALT, life sciences terminology for the Word ontology add-in, and educational texts from areas unrelated to physics, such as law, for PAUX. Notably, sTeX could, in principle, support units already, as its wide coverage of the conceptual model of OpenMath and its generic mechanism for defining new symbols and concepts could easily be utilized for specifying the relevant unit and quantity symbols. Section 5 presents how we have done that in a way that does not disrupt existing LaTeX authoring practices. While LaTeX is commonly used in mathematics, science, and engineering, our solution is unlikely to appeal to life scientists, where Microsoft Office is more widely used; however, we leave unit support for word processors to future work.

[6] The International System of Units [39].

4.3 Computation

This section briefly covers computation as a common prerequisite of interacting with units and quantities, which is covered in the following section. To realize why unit conversion requires more powerful computation facilities than just multiplication, consider conversions of dates between different calendars, such as the Gregorian and the Julian calender with their different notions of months and leap years [47].

OpenMath has been designed for exchanging mathematical expressions between computer algebra systems and automated theorem provers; any OpenMath-aware computer algebra system can therefore, in principle, perform unit conversions on OpenMath expressions (cf. [46] for details). In contrast, RDF and OWL do not allow for defining mathematical operators and functions in a way as straightforward as in OpenMath. SWEET and QUDT introduce custom OWL properties for describing conversion factors (e.g. *qudt:conversionMultiplier*), for which applications would have to provide hard-coded support – until recently. With SPIN (SPARQL Inferencing Notation [22]), there is an emerging standard for representing rules and constraints on RDF graphs, which has been utilized for converting quantities described using QUDT [24]. For computation, SPIN draws on the basic arithmetic operations supported by the SPARQL RDF query language [36]. More complex functions can, in principle, be provided as SPIN rules; so far, there is, however, just one library that implements SPIN [23].

4.4 Interaction

Applications taking advantage of the semantic publishing of units and their quantities have been experimented with by various authors, albeit the lack of authoring support. The unit conversion service [42,38] by Jonathan Stratford, which users can easily extend by uploading new Content Dictionaries (CDs) with new units and conversion rules, provides a good example of the power of semantically annotated units. Besides having implemented a service, Stratford has also identified the difficulties of unit conversion and the limitations of OpenMath's current state with regard to unit representation.

Stratford's conversion service is interactive in that users can enter quantities into a web form and upload definitions of new units. We have additionally made it interactively accessible from web documents that contain MathML formulas with OpenMath annotations, as created by the publishing pipeline explained in section 2.2 (cf. [16]). This interaction with units in publications has, however, remained a proof of concept so far, as *producing* suitably annotated documents required manual authoring of quantity expressions in OpenMath XML markup – a barrier that we are trying to overcome with the work presented in this paper.

Wolfram|Alpha [48], another interactive (web) service, provides unit conversion capabilities through its API [44] and widgets[7] [49]. However, as its preferred input representation is natural language, and its output representation does not make the semantic structures explicit, we did not consider it for our research.

[7] Mini-apps built on top of Wolfram|Alpha queries [50].

5 Semantic Authoring of Units and Quantities

We have revised the available methods and technologies and established that semantic authoring support for units does not formally exist at present. Consequently, we set out to make the first steps towards extending one of the more prepared software solutions, namely sTEX, with a special authoring module for units, by building on the existing pre-semantic toolbox of the *SIunits* LATEX package. sTEX [25] is essentially a collection of LATEX packages that offer semantic macros. sTEX can be translated into XML markup using LATEXML [31] bindings, thus enabling easier subsequent processing – including semantic web publishing (cf. [11]). Our units extension follows a similar approach[8].

As described in section 4.2, *SIunits* provides an sTEX-like content authoring interface. For our running example, we are interested in authoring $\boxed{100\,\mathrm{km/h}}$ in order to create the content representation shown in Listing 1. There are many ways to author the representation in LATEX, e.g. via `$\textrm{100\,km/h}$`. The *SIunits* package makes the process less ad-hoc by focusing on the content and factoring out the presentational quirks, in the form of package options. Hence, one would instead write the more semantic `\unit{100}{\kilo\metre\per\hour}`. Moreover, the use of sTEX (unit) modules enables a more appropriate semantic (markup) representation of a quantity-unit pair by eliminating the inadequate `times` operator (cf. Listing 1) with a generic quantity constructor of the form: $quantityFN : real \times unit \rightarrow quantity$. Also, individual unit constructors can be defined (e.g. $unitFN : real \rightarrow quantity$) to further simplify the authoring process, e.g. `\unit{100}{\gramme}` would be authored as `\gramme{100}`.

It is interesting to observe that a completely different motivation than ours, namely to provide a convenient and centralized interface to control the *presentation* of the unit entities on a document level, essentially leads to the same result that we desire – a *semantics-oriented* authoring interface.

In our effort to leverage this functionality, we first created a LATEXML binding for the *SIunits* package. It helped us to pinpoint the semantic map between the interface and the OPENMATH representation and provided a non-invasive semantic enrichment for LATEX documents based on the package. Next, we use the gained understanding in building a native sTEX module for units, roughly based on the *SIunits* interface. Table 1 shows a small snippet comparing the different stages. One easily notices the abbreviative power of the sTEX approach, which hides the verbose and overly complex binding declaration under its hood, exposing the author to a controlled LATEX vocabulary and facilitating reuse.

The Planetary eMath3.0 system, into which we have integrated the components of our semantic publishing pipeline, provides in-browser editing of sTEX documents with semantic syntax highlighting as well as context-aware autocompletion of semantic macros and links [27]. In the same environment, the user can

[8] The SIunits bindings and sTeX extension will be released in the respective bundles (the arXMLiv binding library and the sTeX package on CTAN) with the authors' strong commitment to free software licenses compatible with the originals.

Table 1. Definitions for \kilo\metre, typeset as ‘km’

Language	Definition	Semantics

LaTeX	`\newcommand{\kilo}{\ensuremath{\mathrm{k}}}`
	`\newcommand{\metre}{\ensuremath{\mathrm{m}}}`
		✗

LaTeXML	`DefConstructor('\kilo{}','`
	`<ltx:XMApp>`
	`    <ltx:XMTok meaning="prefix" cd="units_ops1"/>`
	`    <ltx:XMTok meaning="kilo" cd="units_siprefix1">`
	`       k`
	`    </ltx:XMTok>`
	`    #1`
	`</ltx:XMApp>');`
	`DefConstructor('\metre','`
	`<ltx:XMTok meaning="metre" cd="units_metric1">`
	`    m`
	`</ltx:XMTok>');`
		✓

sTeX	`\symdef[name=kilo,cd=units_siprefix1]{kiloPX}{\mathrm{k}}`
	`\symdef[name=metre,cd=units_metric1]{metre}{\mathrm{m}}`
	`\symdef[name=prefix,cd=units_siprefix1]{prefixFN}{}`
	`\symdef{kilo}[1]{\mixfixii{}{\kiloPX}{\prefixFN}{#1}{}}`
		✓

interact with the published versions of these documents, as we will explain in the following section.

6 Interaction with Units and Quantities

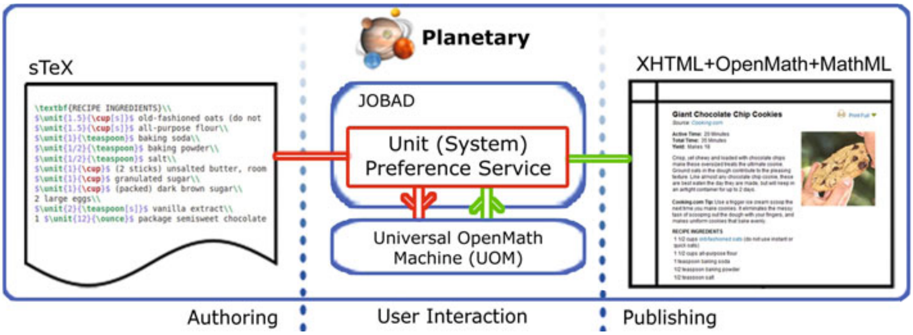
Given the provisions for authoring support, we move to the added-value benefits one could reap from interacting with a published document. This section details relevant use cases and explains the prerequisites that are already available.

6.1 Unit (System) Preference Service

A concrete scenario for a prospective service that would take advantage of semantically published papers, based on the ideas from section 3, can be evolved on top of common published material like *cooking recipes*. These provide a good use case thanks to the high density of units and quantities they contain. Moreover, the physical quantities are restricted to a small subset (quantity/mass related units) including special types of *units* [1] which are not formally defined and might prove to be misleading:

$$\begin{aligned} 1\ \textbf{teaspoon (tsp)} &\approx 5\ \textit{millilitres (mL)} \\ 1\ \textbf{tablespoon (tbsp)} &\approx 15\ \textit{millilitres (mL)} \\ 1\ \textbf{cup} &\approx 250\ \textit{millilitres (mL)} \end{aligned}$$

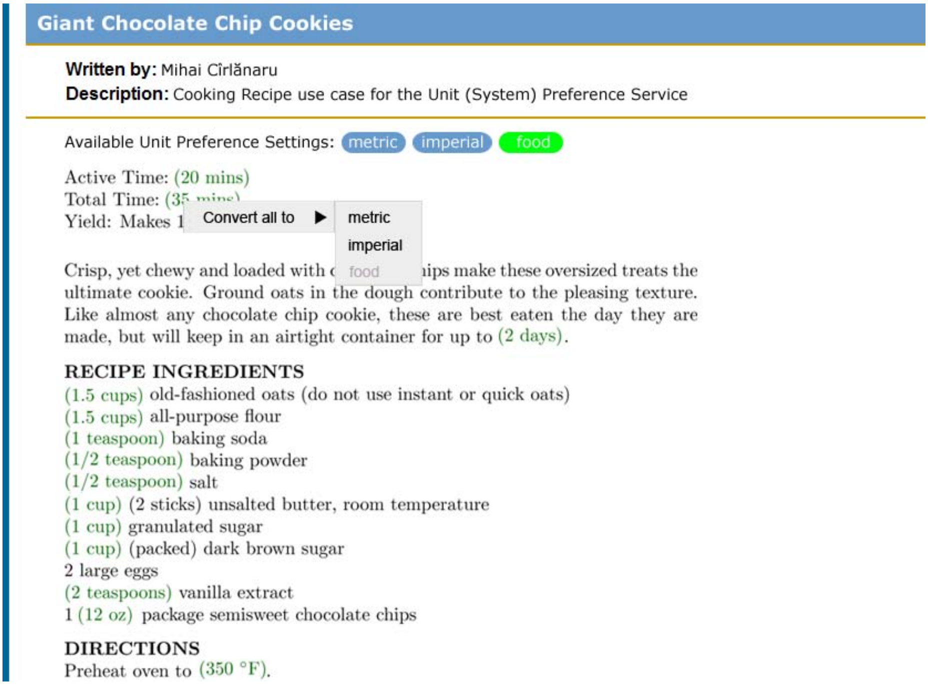
The idea of the *unit (system) preference* service is to allow the user/reader to choose a preferred system of units (e.g. imperial, metric) or simply preferred types of units (e.g. "minutes" instead of "hours", "kilogrammes" instead of "grammes") for the representation of physical quantities and then seamlessly adapt the document to these preferences. This can only be achieved at the end of the semantic publishing pipeline: The publishing process implemented by the Planetary eMath3.0 system requires a machine-comprehensible representation of knowledge (here: units and quantities) as described in section 3 and generated, e.g., via the authoring support introduced in section 5, and then applies a *semantics-preserving transformation*, resulting in a human-comprehensible published document with user-invisible but machine-readable annotations (here: XHTML with OpenMath-annotated MathML formulae) [27]. Into these annotations, the Planetary frontend hooks interactive services, utilizing the JOBAD library (Javascript API for OMDoc-based Active Documents [16]), which provides for communication with web services, manipulation of the user-visible as well as the machine-readable parts of the document, and providing user interface primitives such as a context menu. In our *unit (system) preference* service for Planetary, the computational facilities required for converting quantities are provided by the Universal OpenMath Machine web service [51], which reasons and computes with OpenMath objects. Figure 1 visualizes the architecture and data flow.


Planetary
sTeX
XHTML+OpenMath+MathML
JOBAD
Unit (System) Preference Service
Universal OpenMath Machine (UOM)
Giant Chocolate Chip Cookies
Authoring
User Interaction
Publishing

```
\textbf{RECIPE INGREDIENTS}\\
$\unit{1.5}{\cup[s]}$ old-fashioned oats (do not
$\unit{1.5}{\cup[s]}$ all-purpose flour\\
$\unit{1}{\teaspoon}$ baking soda\\
$\unit{1/2}{\teaspoon}$ baking powder\\
$\unit{1/2}{\teaspoon}$ salt\\
$\unit{1}{\cup}$ (2 sticks) unsalted butter, room
$\unit{1}{\cup}$ granulated sugar\\
$\unit{1}{\cup}$ (packed) dark brown sugar\\
2 large eggs\\
$\unit{2}{\teaspoon[s]}$ vanilla extract\\
1 $\unit{12}{\ounce}$ package semisweet chocolate
```

Fig. 1. Architecture of the Unit (System) Preference service, and data flow in the Chocolate Chip Cookies recipe [10] use-case

We chose a *cooking recipe* use-case as a proof of concept for such a generic (web) service taking advantage of semantically published units; this should not be treated as a fully fledged, independent software product. The design choice was to present the user only with the generic, commonly-used, unit systems (e.g. metric, imperial) and not with specific units as options for preference, especially due to the numerous existing types of units (some of which might not even apply to the subset of unit occurrences in the document). Once the user selects a unit preference, either from the *Available Units Preference Settings* bar on top, or from the context menu, each semantically annotated unit in the document passes through the data flow shown in figure 1. It is replaced with the converted quantity and the new unit (again the choice was made for the most commonly used units, e.g. *tablespoons* would be converted to *milli litres* for the metric preference), rendered in a human-readable way, but with the semantic structure preserved in a machine-readable annotation. Figure 2 demonstrates the interactive user interface of the unit (system) preference service as well as Planetary's rendering of the sample cooking recipe.[9] For further technical details about the architecture of the service, we refer to [8].



Giant Chocolate Chip Cookies

Written by: Mihai Cîrlănaru
Description: Cooking Recipe use case for the Unit (System) Preference Service

Available Unit Preference Settings: metric imperial food

Active Time: (20 mins)
Total Time: (35 mins)
Yield: Makes 1

Crisp, yet chewy and loaded with c ips make these oversized treats the ultimate cookie. Ground oats in the dough contribute to the pleasing texture. Like almost any chocolate chip cookie, these are best eaten the day they are made, but will keep in an airtight container for up to (2 days).

RECIPE INGREDIENTS
(1.5 cups) old-fashioned oats (do not use instant or quick oats)
(1.5 cups) all-purpose flour
(1 teaspoon) baking soda
(1/2 teaspoon) baking powder
(1/2 teaspoon) salt
(1 cup) (2 sticks) unsalted butter, room temperature
(1 cup) granulated sugar
(1 cup) (packed) dark brown sugar
2 large eggs
(2 teaspoons) vanilla extract
1 (12 oz) package semisweet chocolate chips

DIRECTIONS
Preheat oven to (350 °F).

Fig. 2. Screenshot of the Unit (System) Preference service for the cooking recipe

Note that Google's cooking recipe search [18] offers a related web service. However, their semantic markup for cooking recipes [17] does not extend down to an explicit representation of units and quantities, and thus the interactive conversion capabilities are limited or non-existent.

6.2 Prospective Services Based on Semantically Published Units

Having described in detail one service that enhances the user experience by publishing units semantically, we now list further potential services and applications that the same technology could enable:

[9] `http://trac.mathweb.org/planetary/wiki/Demo_PlanetBox`

- **Mapping Natural Sciences Concepts to their Respective Units:** defining Content Dictionaries that would describe the connection of units to general natural sciences concepts like *force* (measured in Newtons: $N = \frac{kgm}{s^2}$ or any variant of the ratio) or *energy* (measured in Joules: $J = Nm = \frac{kgm^2}{s^2} = \ldots$) and plenty of other examples. The interconnection of concepts in sciences: $Energy = Force \times displacement$ can further enable scientific formula "spell checking" which might prove to be of great value to physicists, astronomers and many others.
- **Unknown Unit Lookup:** In theoretical scientific papers authors usually use abbreviations for concepts (e.g. N for *Newtons* – the unit for *force*) without mentioning anything about units/dimensions, which might turn out to be difficult for the readers who would be interested to know, for example, the order of measurement (magnitude) for the unknown physical quantities and also a (small) description of the respective concept (e.g. Pa is the unit for *pressure*). Defining a generic way in which semantics can be added to such unknown symbols will enable showing/hiding units for expressions/formulas.
- **Unit and Quantity Semantic Search:** a library-level service that would allow searching for units by their type, name and magnitude and return the relevant results independently of the measuring standard of the occurrences in the paper (e.g. imperial or metric) and also independent of their form (N or $\frac{kgm}{s^2}$).[10]
- **Quantity and Unit's Magnitude Manipulation:** a document interaction service that is able to transform for example $100N \rightarrow 0.1kN$ or $0.1 \times 10^3 N$ or $0.1 \times 10^3 \frac{kgm^2}{s^2}$. This can be useful when it comes to simplifying representations and adapting them consistently to a certain type of magnitude (for example *all occurrences of force expressions should have their unit represented in kN*).

As detailed at the beginning of this paper, having a standard, uniform understanding of units and quantities can prevent hazards and even eliminate entire compatibility check processes in industry. The presented list of prospective enabling technologies shows only a few of the numerous opportunities of interacting with units and quantities in semantically published documents and serves as a strong motivation for future research in this direction.

7 Enabling Semantic Units in Legacy Corpora

The ARXMLIV corpus is the ideal environment for the identification of units and quantities since it contains a collection of more than 600,000 scientific publications. It is based on Cornell University's ARXIV e-Print archive [3] originally typeset in LaTeX, converted to XML in order to achieve easy machine-readability, partial semantics recovery and clear separation of document modalities such as natural language and mathematical expressions [41]. Currently, the project

[10] In contrast, state-of-the-art scientific publication search services, such as Springer's LaTeX search [40], do not support the semantics of units.

has achieved a successful conversion rate of nearly 70% to a semantically enriched XHTML+MathML representation, natively understandable by modern web browsers [26].

A proof-of-concept check, performed via the arXMLiv build system (see [41]) revealed roughly 150 arXiv articles using the *SIunits* package, with an outlook for close to tripling the number when considering sibling packages such as *units* and *SIunitx*. This gives our work on creating a semantic binding for *SIunits* an even stronger benefit, as we can directly and non-invasively enrich legacy publications, putting them one step further on the path to semantic publishing. An additional, mid-term benefit is the opportunity to build a linguistic *Gold Standard* for units; we created both legacy (to presentational MathML) and semantic (to OpenMath) bindings in order to provide a raw, presentational output and its annotated, semantic counterpart. Having both as a basis, unit spotters can then be developed using methods of Computational Linguistics and Machine Learning, further enriching the arXMLiv corpus.

Such enhancements not only enable the interactive services of semantic publishing on legacy corpora, but also provide a tempting outlook to the development of an ecosystem of linguistic analysis modules, which can draw on the captured semantics of units and quantities, as originally envisioned by the LaMaPUn project [15].

8 Conclusions and Future Work

Units and quantities are sufficiently wide-spread and important to not be disregarded from the context of semantic documents. Unfortunately, by now, there have been only isolated approaches (see section 4) to exploit the semantic power of units. Moreover, the wide range of existing unit types and representations makes it almost impossible to identify and semantically enrich all of them, especially when we are talking about occurrence contexts as unrelated as cooking recipes, medical prescriptions, technical documents or scientific papers.

We have emphasized the importance of three major components of the semantic publishing process for units – *representation*, *authoring* and *interaction* –, and detailed technologies for improving each of them. Moreover, by providing a cooking recipe interaction use-case as well as a series of further potential services and applications on top of semantically published units, we contribute means of better manipulation and interpretation of *units and quantities* to the Semantic Publishing Industry and to legacy corpora.

Acknowledgments. The authors would like to thank Michael Kohlhase for his extensive support and advice regarding the writing of this paper, Anton Antonov for writing the LaTeXML bindings for the *SIunits* LaTeX package, and the other Planetary developers[11] for providing the context for developing our authoring and interaction services. This extended and revised version of a submission to the ESWC 2011 workshop on Semantic Publishing (SePublica)

[11] http://trac.mathweb.org/planetary/wiki/people

has benefited from the extensive helpful suggestions provided by the anonymous peer reviewers, and from constructive feedback and further suggestions given by the participants of the workshop.

References

1. Code of Federal Regulations – Food and Drugs, http://edocket.access.gpo.gov/cfr_2004/aprqtr/21cfr101.9.htm
2. Aviation Safety – Air Canada Accident Report, http://aviation-safety.net/database/record.php?id=19830723-0 (visited on October 25, 2010)
3. arxiv.org e-Print archive, http://www.arxiv.org
4. MathML 3.0. Recommendation. W3C (2010), http://www.w3.org/TR/MathML3
5. Barany, M.J.: [B]ut this is blog maths and we're free to make up conventions as we go along': Polymath1 and the modalities of 'massively collaborative mathematics. In: WikiSym (2010)
6. Measurement Units Ontology, http://forge.morfeo-project.org/wiki_en/index.php/Measurement_Units_Ontology (visited on April 16, 2011)
7. Buswell, S., et al.: OpenMath 2.0. Tech. rep. The OpenMath Society (2004), http://www.openmath.org/standard/om20
8. Cîrlănaru, M.: Authoring, Publishing and Interacting with Units and Quantities in Technical Documents. BSc. Thesis. Jacobs University Bremen (2011)
9. Collins, J.B.: OpenMath Content Dictionaries for SI Quantities and Units. In: Carette, J., Dixon, L., Coen, C.S., Watt, S.M. (eds.) MKM 2009, Held as Part of CICM 2009. LNCS, vol. 5625, pp. 247–262. Springer, Heidelberg (2009)
10. Cooking.com – Giant Chocolate Chip Cookies, http://www.cooking.com/recipes-and-more/recipes/Giant-Chocolate-Chip-Cookies-recipe-5112.aspx (visited on March 5, 2011)
11. David, C., et al.: Publishing Math Lecture Notes as Linked Data. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010. LNCS, vol. 6089, pp. 370–375. Springer, Heidelberg (2010)
12. DBpedia, http://dbpedia.org (visited on January 23, 2010)
13. Davenport, J.H., Naylor, W.A.: Units and Dimensions in OpenMath (2003), http://www.openmath.org/documents/Units.pdf
14. Fink, J.L., et al.: Word add-in for ontology recognition: semantic enrichment of scientific literature. BMC Bioinformatics 11, 103 (2010)
15. Ginev, D., et al.: An Architecture for Linguistic and Semantic Analysis on the arXMLiv Corpus. In: Applications of Semantic Technologies Workshop at Informatik (2009), http://www.kwarc.info/projects/lamapun/pubs/AST09_LaMaPUn+appendix.pdf
16. Giceva, J., Lange, C., Rabe, F.: Integrating Web Services into Active Mathematical Documents. In: Carette, J., Dixon, L., Coen, C.S., Watt, S.M. (eds.) MKM 2009, Held as Part of CICM 2009. LNCS, vol. 5625, pp. 279–293. Springer, Heidelberg (2009)
17. Google Cooking Recipe Publishing Schema, http://www.google.com/support/webmasters/bin/answer.py?answer=173379 (visited on June 5, 2011)
18. Google Cooking Recipe Search, http://www.google.com/landing/recipes/ (visited on May 6, 2011)

19. Groza, T., et al.: SALT – Semantically Annotated LATEX for Scientific Publications. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 518–532. Springer, Heidelberg (2007)
20. Heldoorn, M.: The SIunits package: Consistent application of SI units, http://mirror.ctan.org/macros/latex/contrib/SIunits/SIunits.pdf (visited on March 13, 2011)
21. QUDT – Quantities, Units, Dimensions and Data Types in OWL and XML, http://www.qudt.org (visited on July 15, 2011)
22. SPIN – Overview and Motivation. Member Submission. W3C, http://www.w3.org/Submission/2011/SUBM-spin-overview-20110222/
23. The TopBraid SPIN API, http://topbraid.org/spin/api/ (visited on July 15, 2011)
24. Units ontology with SPIN support published, http://composing-the-semantic-web.blogspot.com/2009/08/units-ontology-with-spin-support.html (visited on July 15, 2011)
25. Kohlhase, M.: Using LATEX as a Semantic Markup Format. Mathematics in Computer Science, 2.2 (2008)
26. Kohlhase, M., et al.: MathWebSearch 0.4, A Semantic Search Engine for Mathematics (2008), http://mathweb.org/projects/mws/pubs/mkm08.pdf
27. Kohlhase, M., et al.: The Planetary System: Web 3.0 & Active Documents for STEM. In: Procedia Computer Science 4 (2011): International Conference on Computational Science (ICCS). Finalist Executable Papers Challenge (2011)
28. Kohlhase, M., Rabe, F.: Semantics of OpenMath and MathML3. In: 22nd OpenMath Workshop (2009)
29. Lange, C.: Ontologies and Languages for Representing Mathematical Knowledge on the Semantic Web. Semantic Web Journal (accepted, 2011) http://www.semantic-web-journal.net/content/new-submission-ontologies-and-languages-representing-mathematical-knowledge-semantic-web
30. CNN – NASAs metric confusion caused Mars orbiter loss, http://articles.cnn.com/1999-09-30/tech/9909_30_mars.metric_1_mars-orbiter-climate-orbiter-spacecraft-team?_s=PM:TECH (visited on October 29, 2010)
31. LaTeXML: A LATEX to XML Converter, http://dlmf.nist.gov/LaTeXML/ (visited on March 3, 2011)
32. Oxford English Dictionary. “unit” definition, http://dictionary.oed.com/entrance.dtl (visited on October 29, 2010)
33. PAUX Technologies, http://paux.de (visited on October 10, 2010)
34. Planetary Developer Forum, http://trac.mathweb.org/planetary/ (visited on January 20, 2011)
35. PlanetMath.org – Math for the people, by the people, http://planetmath.org (visited on January 6, 2011)
36. SPARQL Query Language for RDF. Recommendation. W3C, (2008), http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/
37. Raskin, R.G., Pan, M.J.: Knowledge representation in the semantic web for Earth environmental terminology (SWEET). Computers & Geosciences 31 (2005)
38. Stratford, J., Davenport, J.H.: Unit Knowledge Management. In: Autexier, S., Campbell, J., Rubio, J., Sorge, V., Suzuki, M., Wiedijk, F. (eds.) AISC 2008, Calculemus 2008, and MKM 2008. LNCS (LNAI), vol. 5144, pp. 382–397. Springer, Heidelberg (2008)
39. The International System of Units (SI) Bureau International des Poids et Mesures 8 edn. (2006), http://www.bipm.org/utils/common/pdf/si_brochure_8_en.pdf

40. Springer, (ed.) LATEX Search, `http://www.latexsearch.com` (visited on April 16, 2011)
41. Stamerjohanns, H., et al.: Transforming large collections of scientific publications to XML. Mathematics in Computer Science, 3.3 (2010)
42. Stratford, J.: Creating an extensible Unit Converter using OpenMath as the Representation of the Semantics of the Units. Tech. rep. 2008-02. University of Bath, `http://www.cs.bath.ac.uk/pubdb/download.php?resID=290`
43. Semantic Web for Earth and Environmenta l Terminology (SWEET). NASA, `http://sweet.jpl.nasa.gov/` (visited on August 22, 2010)
44. Wolfram|Alpha API, `http://www.wolframalpha.com/developers.html` (visited on May 5, 2011)
45. US Metric Association "Unit Mixups" article, `http://lamarcolostate.edu/~hillger/unit-mixups.html` (visited on October 25, 2010)
46. Vrandečić, D., et al.: Semantics of Governmental Statistics Data. In: Web Science (2010), `http://journal.webscience.org/400/`
47. Wikipedia: Hebrew calendar, `http://en.wikipedia.org/wiki/Hebrew_calendar`
48. Wolfram|Alpha, `http://www.wolframalpha.com` (visited on May 5, 2011)
49. Wolfram|Alpha Units and Measures Widgets, `http://developerwolframalpha.com/widgets/gallery/category/?cat=units` (visited on May 5, 2011)
50. Wolfram|Alpha Widgets, `http://developer.wolframalpha.com/widgets/` (visited on May 5, 2011)
51. Zamdzhiev, V.: Universal OpenMath Machine. BSc. Thesis. Jacobs University Bremen (2011)

Policy Intelligence in the Era of Social Computing: Towards a Cross-Policy Decision Support System

Riccardo Boero[1], Enrico Ferro[2], Michele Osella[2],
Yannis Charalabidis[3], and Euripidis Loukis[3]

[1] Regione Piemonte, Corso Regina Margherita, 174 – 10152 Torino, Italy
ricboero@gmail.com
[2] Istituto Superiore Mario Boella, Via Boggio, 61 – 10138 Torino, Italy
{enrico.ferro,michele.osella}@ismb.it
[3] University of the Aegean, Gorgyras Str. – 83200 Karlovassi, Greece
{yannisx,eloukis}@aegean.gr

Abstract. The paper presents a policy analysis framework developed through a process of interdisciplinary integration as well as through a process of end users needs elicitation. The proposed framework constitutes the theoretical foundation for the Decision Support Component of a technological platform bringing together Social Media and System Dynamics simulation developed within the PADGETS project. The main novelties introduced have to do with the possibility to provide decision makers with a set of concise, fresh and relevant data in a cost effective and easily understandable way.

Keywords: ICT Governance, Policy Intelligence, Policy Modeling, eParticipation, Social Media, Decision Support Systems, Public Sector Innovation, eGovernment, PADGETS.

1 Introduction

In the second decade of the millennium European Governments are confronted with three important long-term trends.

1. The combined effect of an increase in the rate of change and in the level of interdependence and interconnectedness among regions, activities and groups is leading to a fast-evolving and unpredictable world characterized by significant levels of complexity and uncertainty. The concept of "liquid modernity" [1] proposed by Zygmunt Bauman represents a useful attempt to frame part of such phenomenon. According to the Polish sociologist, in fact, social forms and institutions no longer have enough time to solidify and cannot serve as frames of reference for human actions and long-term life plans to the extent they served in the past, so individuals have to find other complementary ways to organize their lives.
2. A push towards a more participatory and inclusive style of policy making poses significant challenges in terms of striking the right balance between openness and

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 217–228, 2012.

control, defining new and appropriate styles of management and, finally, integrating participatory activities into existing decision making processes.

3. A forecast, for years to come, of low economic growth and financial instability is leading to tighter budget constraints and less room for mistakes in the allocation of tax payers' money for Government's action.

The concurrence of such socioeconomic, institutional and financial trends calls for a reconceptualization of Government's roles and *modi operandi.*

The creation of the PADGETS (its full title being "Policy Gadgets Mashing Underlying Group Knowledge in Web 2.0 Media") project [2] may be placed in such landscape. The project has been financed in the context of the "ICT for Governance and Policy Modelling" call of the seventh European Framework Program of research (FP7). The main underlying idea of such research endeavor is to bring together social computing with System Dynamics simulation in order to help Governments to render policy making processes more participative and, at the same time, to provide advanced and more effective types of support to public sector decision making processes. In particular, the platform developed within the project blueprint will allow Public Administrations to set up a cost effective participatory process by moving the political discussion from official websites to Social Networks where citizens are already debating. The platform will enable a centralized posting of contents and micro-applications (termed "Policy Gadgets" or, coining a *portmanteau*, "Padgets") to many different Web 2.0 Social Media at the same time, followed by the collection of various types of users' interactions (e.g., views, likes/dislikes, ratings, comments) and by advanced processing and analysis of resulting data in order to provide effective policy intelligence services based on fresh and relevant information.

The aim of this paper is to present a policy analysis framework for this purpose, developed drawing from theories and concepts belonging to different scientific disciplines. The proposed framework constitutes the theoretical foundation on which the Decision Support Component (DSC) of the PADGETS platform has been designed. The development of such a Decision Support System represents a first attempt to provide policy makers with a set of tools that may be precious in tackling grand challenges discussed at the beginning of this section.

Concluding these introductory comments, the paper is structured in five sections. Section two provides a multidisciplinary theoretical background to the presented work. Section three illustrates pivotal principles underlying the proposed policy analysis framework. Section four explains how such framework has been transformed into a working software tool to be integrated into the PADGETS platform. Finally, section five provides some conclusive remarks on the value proposition of the Decision Support System proposed as well as some directions for stimulating future research.

2 Theoretical Background about Participative Decision Support Systems in the Public Sector

All different kinds of organizations, business, public, and non-governmental alike, are becoming aware of soaring complexity in decision making situations [3] [4]. Such complexity, inextricably related to the intricacy of systems [5], can be addressed through decision support tools which can enhance the quality of the decision process[1].

However, in a bevy of situations, multidisciplinary teams, top-notch skilled resources and world class computer suites do not suffice to cope with actual problems: a further need concerns the sharing and "externalization" [6] of tacit knowledge already existing in the society. In fact, collective intelligence emerges as a key ingredient of a "distributed problem-solving" system [7] whose output can significantly enrich the decision process traditionally carried out by experts: in accordance with this trend, politics is progressively moving towards higher public engagement and cooperation in decision making processes.

Even though, as underlined by [8], the number of solutions in the area regarding e-Democracy, e-Participation and related fields is increasing, support systems for decision making are, however, still used mainly in narrow professional circles and have not found their way to political decision makers or to the public. The challenge of successful implementation of DSSs in the public sector, with engagement over the whole spectrum of decision making, is still unmet. In particular, in order to enhance the quality and effectiveness of the decision through knowledge harvesting, simulation of future scenarios and structured comparison of alternatives, DSSs depend on the availability and accessibility of timely, relevant and accurate information [9], which frequently represents the scarce resource.

Since such information may derive from "social sources", new mechanisms are required to enable a public decision process open, transparent and participative in which citizens' contribution is a paramount ingredient characterized by a significant impact. Along these lines, since Web 2.0 applications are already being used in Government not only for soft issues (e.g., public relations, public service announcements) but also for core internal tasks (e.g., intelligence services, reviewing patents, support decision making) [10], it is highly desirable to proceed towards a systematic exploitation of the emerging Social Media by Government organizations in the processes of public policies formulation, aiming to enhance a frictionless e-Participation: by doing this, Governments make a step towards citizens rather than expecting the citizenry to move their content production activity onto the "official" spaces created for e-Participation [11].

To sum up, the implementation of successful e-Governance programs, which heavily leverage on the participative dimension, cannot ignore the presence of DSSs, as computer-based systems that help decision makers confront ill-structured problems through direct interaction with data and analytical models, notwithstanding the access to privileged channels aimed to enable a fully-fledged engagement: the evidence that "it has become impossible to restrict knowledge and its movement to castes of specialists" [12] makes researchers aware that "crowd wisdom" is not merely a Web 2.0 catchy buzzword, but is instead a strategic model to attract an interested and motivated platoon of stakeholders.

[1] The organizational decision making has its roots in the seminal contributions of renowned mavens such as Simon, Cyert and March; for a comprehensive discussion of these issues, see [13].

3 Building the Foundation of a Cross-Policy Decision Support Framework

In order to conceptualize a Decision Support Component (DSC) for the PADGETS platform we started from a set of key underlying assumptions regarding design principles as well as constraints we had to endure.

1. The design should be centered on the policy maker's perspective, focusing on the manifold needs of daily policy making.
2. The DSC as a whole has to be aligned to project mission and orientation: in particular the *leitmotif* to adhere to is the exploitation of many Social Media at the same time in a systematic and centrally managed manner.
3. Considering the economics of the project, reaching internal economies of scope represents for sure a desirable outcome. Thus, the effort has to be geared towards preventing the creation of non-communicating silos and towards avoiding the development "from scratch" of ad-hoc models for each specific pilot or *locus* of implementation.
4. In conceiving the application logic underpinning data elaboration, the novelty brought by PADGETS approach no longer considers individuals as isolated units of analysis but leverages their social connections and the context in which they are immersed as a potentially useful policy tool. By isolating particular behavior of specific groups, the policy maker may take advantage of an additional "weapon": by targeting more connected or more charismatic individuals s/he is likely to obtain better and faster results than by implementing a generic policy not taking into account the role individuals play in their social network.
5. Some potential threats pertain to the vast fields with which policy makers have to deal, such as the cognitive problem of synthesizing the distributed knowledge collected from stakeholders in many different environments and the intrinsic dynamics of public opinion. In light of such inescapable difficulties, it becomes paramount to keep moderate the cognitive effort required to policy makers while let the "machines" do most of the cumbersome work.

Keeping in mind afore-said cornerstones, we developed the architecture of a service that aims at informing the policy maker's decision process (i.e., a decision support tool) by effectively using the knowledge collected through the engagement with a plethora of stakeholders[2] in Web 2.0 Social Media, i.e., virtual spaces which nowadays may ideally symbolize modern *agorae*.

Taking into consideration the rich variety of policy fields, we decided to develop a decision support tool capable to be as much as possible "generic" and "horizontal", meaning that it should be easily and effectively employed for any kind of public policy. This was done, among other reasons, to enhance the appeal of the Decision

[2] We prefer the generic term "stakeholders" to "citizens" because we think that citizens are only the largest kind of stakeholders interested in interacting with policy makers, and that institutions, which cannot be reduced to their single individuals, can be interested too in the innovative ways of participatory policy making introduced by the project. Hence, actors such as, for instance, producers' and consumers' associations, political parties, trade unions, corporations and charities, could be encompassed under the label "stakeholders".

Support System in terms of commercialization, i.e., in order to be turned into a marketable product. As a matter of fact, the possibility to reach a wider pool of potential institutional adopters allows to benefit from economies of scope and scale, that contribute to lower the unit cost of service provision.

Moreover, considering the issue of synthesizing the widespread information collected through many different Web 2.0 participatory tools provided by the project, we started by interacting with local policy makers in order to identify the support they expect from such kind of a tool. Prominent *desiderata* coming from the "requirement phase" regard the potentiality of collecting through a unique tool various information stemming from dissimilar interaction patterns which are peculiar of different stages along the public policy lifecycle. In particular, policy makers would like to receive answers to the following five "archetypal" questions which are relevant during each phase of a public policy lifecycle[3] (agenda setting, policy analysis, formulation, implementation, monitoring):

1. Are stakeholders aware of the public policy?
2. Are stakeholders interested in the public policy?
3. What stakeholders think about the specific public policy solution that the policy maker has proposed? To what extent they accept it?
4. Which are the barriers to policy awareness and interest, and which are the barriers to changes in public opinion about the policy?
5. Which suggestions are coming from stakeholders?

To say it with other words, the first question investigates if stakeholders know that the policy under examination exists; the second question regards to what extent they are inclined to reason and debate about the policy theme. The third point, for its part, is centered on stakeholders' judgment about the policy (e.g., acceptance, rejection, neutrality, indifference). The forth question strives for hints about barriers hampering the policy deployment as well as obstacles hindering the diffusion of the policy message. Finally, the fifth question has its root in the concept of "crowd wisdom", since it tries to collect insightful contributions coming from the collective opinion in an attempt to reap the benefits stemming from bottom-up knowledge percolation.

The identified relevant questions allowed us to design a support tool capable of taking advantage of the fruitful synergy among different methodologies and techniques. In order to devise responses to the first three questions, the most relevant information concerning the policy proposal collected from stakeholders are framed along three basic dimensions (awareness, interest, acceptance); on the other hand, in order to answer questions 4 and 5, stakeholders' suggestions are analyzed through text mining (which is performed on unstructured texts) and opinion analysis (which relies on structured interactions occurring through polls, questionnaires *et similia*).

We conceived the awareness-interest-acceptance framework by taking into account concrete needs of public policy makers and, at the same time, drawing from preeminent theoretical frameworks developed in the disciplines of innovation studies and political science. According to innovation research of Rogers [14], the diffusion of an innovation occurs, in fact, through a five–step process, which is a type of decision making: awareness, interest, evaluation, trial, and adoption. Furthermore,

[3] Afore-mentioned phases of the policy making cycle are defined by OECD in [15].

OECD [16] identifies three stages of on-line engagement: information (for increasing stakeholders' awareness), consultation (providing opinions about the policy) and active engagement. In addition, the concept of policy acceptance is well-recognized in political science as it allows to understand the coherence between the proposed public action and the systems of values present in the society, a necessary precondition for a successful implementation of the policy; considering the literary landscape as well as down-to-earth policy initiatives, the concept of acceptance may be seen from a normative point of view or from innovation point of view[4].

4 The Architecture of the PADGETS Decision Support Component

The PADGETS Decision Support Component is the analytic engine processing and analyzing the results of the PADGETS Campaign[5] in order to extract useful information for the policy maker. To say it in a nutshell, it is the software component which prepares the information for supporting policy makers.

The DSC relies on information coming from the policy maker, from Social Media Platforms[6], and from the Padget[7], and consists of two main modules: the PADGETS Analytics and the PADGETS Simulation Model. Whilst the Analytics module aims at grouping and synthesizing raw information and at solving possible problems of statistical nature in collected data, the Simulation Model aspires to forecast future scenarios of opinion change.

In the following paragraphs we sketch the basic working mechanisms of the DSC, concluding with a discussion on the effectiveness of the proposed design both through the technical lens and the value-related lens.

4.1 Inputs

The inputs of the DSC model come from three sources, i.e., Social Media, Padgets and policy makers.

Data coming from Social Media (retrieved by public APIs) and from the Padgets may be unstructured (i.e., open text content) or, otherwise, structured (i.e., users' actions and selections). Unstructured data flow into text mining activities while structured data, for their part, constitute the inputs of quantitative analysis taking place in both PADGETS Analytics and PADGETS Simulation Model.

[4] For an example of EU funded research project on policy acceptance see [17].

[5] In the project jargon, a PADGETS Campaign entails a set of activities covering creation, distribution, interaction, monitoring and termination of one or more policy messages oriented towards a specific goal and related to the same theme.

[6] Major Social Media Platforms covered by the scope of work are: Facebook, LinkedIn, Twitter, Blogger, Digg, Scribd, YouTube, Picasa, Flickr. Currently the PADGETS consortium is examining the inclusion of the upcoming Google Plus in the Social Media panoply under investigation.

[7] In line with the project dictionary, a Padget is a resource (application or content), typically instantiating within a variety of Social Media Platforms, which provides interactivity with stakeholders through an *ensemble* of native and "augmented" social functionalities.

The inputs database contains two broad categories of information in terms of data organization:

1. policy maker's data referring both to the target stakeholders' group (socio-demographic data) and to Campaign attributes (e.g., date of launch, timeframe, question structure);
2. data stemming from social engagement, collected at the finest granularity (i.e., individual users' data with recourse to appropriate techniques of anonymization, in compliance with data protection legal frameworks) and structured according to the two dimensions of user and time (i.e., the user who acted and the time of action).

4.2 Output Indicators

DSC outputs are developed along three different concepts in accordance with section 3: awareness, interest and acceptance.

From a Padget end user's perspective, each concept is a set containing the following ones, but not vice versa. Thus, a user interested in the policy must be also aware of the policy, but the opposite might not hold (i.e., an aware user can be not interested in the policy).

The distinction between the concepts is that acceptance concerns polarized judgments (i.e., positive and negative) collected by means of the Padget, interest regards all data generated by a proactive behavior by users in Social Media, and finally awareness is a matter of an only passive reception of the policy message in Social Media (i.e., without further spreading or commenting the Padget announcement owing to a lack of interest).

The typologies of outputs that it is possible to compute are three. Firstly, it is possible to draw the distribution of data over the main categories of stakeholders identifiable according to socio-demographic variables. Secondly, it is possible to project the data into the actual world. Lastly, estimates on how policy awareness-interest-acceptance will change in the near future can be computed through algorithms included in the Simulation Model.

4.3 Modeling and Simulation

Keeping a helicopter view, the "big picture" of the PADGETS Decision Support Component structure (Fig. 1) displays how the two DSC modules transform the structured inputs coming from different sources in outputs useful for satiating policy makers' appetite, while an external module carries out text mining functionalities in order to determine stakeholders' suggestions. The figure, moreover, underlines that only the relevant information is presented to policy makers among the many we identified above.

Passing to the description of how the components work, actual distributions of awareness-interest-acceptance are obtained by a sort of mere data aggregation that simply groups raw Social Media and Padget data according to socio-demographic variables.

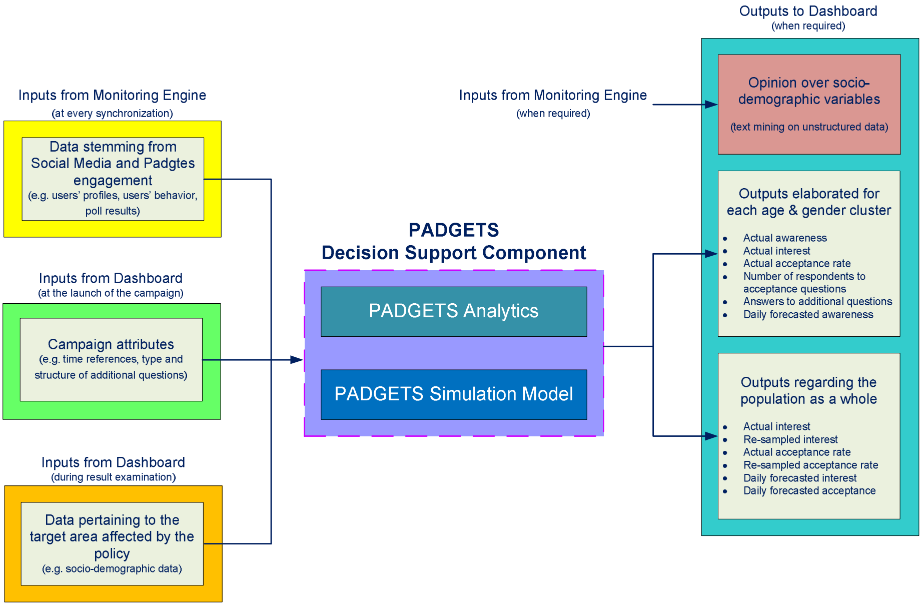

Outputs to Dashboard
(when required)
Inputs from Monitoring Engine
(at every synchronization)
Data stemming from Social Media and Padgtes engagement
(e.g. users' profiles, users' behavior, poll results)
Inputs from Monitoring Engine
(when required)
Opinion over socio-demographic variables
(text mining on unstructured data)
Outputs elaborated for each age & gender cluster
Actual awareness
Actual interest
Actual acceptance rate
Number of respondents to acceptance questions
Answers to additional questions
Daily forecasted awareness
PADGETS
Decision Support Component
PADGETS Analytics
PADGETS Simulation Model
Inputs from Dashboard
(at the launch of the campaign)
Campaign attributes
(e.g. time references, type and structure of additional questions)
Outputs regarding the population as a whole
Actual interest
Re-sampled interest
Actual acceptance rate
Re-sampled acceptance rate
Daily forecasted interest
Daily forecasted acceptance
Inputs from Dashboard
(during result examination)
Data pertaining to the target area affected by the policy
(e.g. socio-demographic data)

Fig. 1. The structure of the PADGETS Decision Support Component

In addition, in order to increase the real world significance of obtained results the re-sampling of raw data is computed: moving in this direction represents an attempt to remedy to possible underrepresentation of specific groups of stakeholders in the Social Media realm. To exemplify, elderly generations are likely to show lower penetration rates in Social Media: the resampling activity is thus aimed at reducing this bias (and several others) in the estimation of current and future awareness, interest and acceptance rates.

Finally, awareness-interest-acceptance future scenarios are the most complex results, representing the flagship of the DSC. They start from the re-sampling of raw data, on which a procedure computes trends evolution along the dimensions of policy indicators and in light of socio-demographic variables: for this purpose, a System Dynamics simulation is run in order to estimate the possible future outcomes and their probability of occurrence. Being the Simulation Model a System Dynamics one, there is the need for two main kinds of elements[8], stocks and flows. Thus the Simulation Model is based on the identification of relevant stocks according to socio-demographic variables and flows depending on the above-mentioned trends of evolution. In particular, the simulation is focused on how distinct socio-demographic clusters of stakeholders will change their level of awareness-interest-acceptance in the near future in light of intertwined social connections and resulting "viral" contagious phenomena; the rationale underlying the model entails that clusters are not

[8] For a complete introduction to System Dynamics, to its concepts and terminology, please refer to [18].

independent, therefore several feedback loops and cascade effects can be at work testifying a blurred overlap of endogenous evolution and external influences.

Under a procedural point of view, the Simulation Model operates as follows.

1. The System Dynamics model is built-up automatically in the background in accordance with policy maker's input; by doing this, significant results are achieved in hiding the inherent complexity characterizing the Simulation Model from policy maker's eyes.
2. A calibration procedure is performed through a regression which computes coefficients and parameters of the model, estimating both mutual interrelations and endogenous growth in view of the evolution registered hitherto.
3. Given the stochastic nature of the simulation, the heterogeneity of collected data and the uncertainty affecting some parameters, the simulation will be run to explore all the possible outcomes of variations in parameters (including the random seed for stochastic processes). As a result, confidence intervals are computed with a certain level of confidence for each forecasted policy indicator: by moving in this direction, it is possible to shift the perspective from deterministic to probabilistic.
4. Once the simulation routines have run "behind the scene", fresh and customized results are passed to the front-end and presented to the policy maker in a compelling way through a fully-fledged web-based visualization engine.

4.4 Effectiveness of the Solution

An overall glance on the DSC shows that elaborated outputs directly answer the first three policy makers' questions out of the five we want to address (section 3): presenting to policy makers detailed data about the three concepts of awareness, interest and acceptance, they will obtain extensive answers to their recurrent questions and they will also get a "dynamic" idea about future trends in society towards the policy on focus.

The remaining two policy makers' questions we plan to answer pertain to perceived barriers and suggestions coming from stakeholders. In this case the answers to policy makers are provided by the DSC as a whole, that is to say by the results coming from the Analytics and the Simulation Model**,** on which we focused in this paper, and by analytic activities on unstructured data such as text mining.

Delving into this topic, the first three policy questions we considered as aims of the DSC are directly answered by singular groups of indicators provided by the DSC. For instance, if the policy maker is interested in evaluating the policy awareness in the population, the DSC provides a set of indicators related to awareness which describe the contemporary level of that phenomenon accompanied by its near future trend. On the other side, the two remaining questions concerning emerging barriers and public suggestions are *per se* unforeseeable in their structure and content, and it thus becomes impossible to identify *ex ante* a "rigid" structure to organize such information. For this purpose opinion mining [19] methods will be exploited. In this field, the effort is geared towards extracting opinions from unstructured human-authored texts (posts, comments *et similia*) having recourse to techniques such as feature-based sentiment analysis, topic identification and sentiment classification. Semantic analyses in this vein provide an insightful glimpse on "what people think" capable to conspicuously reinforce the governmental policy intelligence.

For the actual implementation “on the field” of the DSC architecture, we chose to rely on autonomous and platform independent software classes with data interfaces for communicating inputs and outputs with other building blocks belonging to the platform. Our choice has been to code this software entirely in Java (avoiding recourse to external libraries or tools for System Dynamics modeling) in order to guarantee platform independence, eventual Web distribution and for relying on well-established libraries devoted to required activities of data management and regression.

In conclusion, a pivotal aspect not to be overlooked is the compliance of the DSC with policy regulation and data protection legislation. In particular, all front-end operations required for data collection are performed after properly informing the user on subsequent processing activities in a precise and descriptive manner. PADGETS employs W3C’s P3P [20], consequently for each PADGETS Campaign a P3P-compliant privacy policy is provided to inform the user about data elaboration within PADGETS in order to let him/her make a well-informed decision. Once obtained the positive user’s consent, PADGETS consortium operates without storing users’ IDs in a re-usable way. Finally, during the entire project lifetime there is no transfer of personal data to third parties: in fact, gathered data are stored on servers of one of the consortium partners inside the EU region and are owned by the consortium.

4.5 From Code Implementation to Value Creation

Since the “North Star” that guided our action is being markedly “value-driven” rather than being “tech-driven”, in our opinion it may sound wise to conclude the paper by coming back to the policy maker’s angle in order to pinpoint how the tool previously described is able to “make sense of data”, smoothing the way for a better informed policy decision.

Summarizing, from a policy maker’s perspective the value proposition of the decision support tool we designed may be recapitulated as follows:

1. A methodological contribution related to information classification, since the tool provides a well-grounded conceptual framework aimed to classify and aggregate data stemming from social engagement in light of an increasing level of stakeholders’ involvement (awareness, interest, acceptance).
2. A reduction of information complexity, given by a set of peculiar traits (e.g., data aggregation along multiple dimensions, cross-platform data analysis, data projection into the real world, simulation of phenomena evolution in the near future) leading to a well-framed synthesis of unstructured (and sometimes inadvertent) society’s input which could be used in order to forecast possible impacts of policies in light of surfacing *vox populi*.
3. A support to emerging Governance models, since it enables new ways for collecting, organizing and delivering information at different authority levels, opening-up on-going Governance models by letting a wider audience to contribute to the political debate.

5 Conclusions and Limitations

In this article we presented the preliminary results produced during the first year of activity of the PADGETS research project. In particular, the discussion focused on the policy analysis framework underpinning the DSC of the PADGETS platform. Such analytical framework was generated through a process of interdisciplinary integration (mainly drawing form the diffusion of innovation and political science literature) as well as through a process of end users needs elicitation.

The intent behind the development of the analytical framework was to provide a first contribution towards the creation of a Decision Support System that could help policy makers in facing a number of relevant questions often arising through the policy cycle. This was done by introducing an innovation bringing together Social Networks and System Dynamics simulation. To date, in fact, the use of ICT tools for decision support has traditionally been a "closed door" activity usually carried out with static external inputs in the form of codified or unstructured data coming from different sources (e.g., statistical offices). Such approach presents a number of important limitations: evident examples are the lack of a direct connection with the recent external reality on which the policy decision has to impact and the inherent delay present in the policy response due to the lead time necessary to collect and process the relevant data required for the analysis. To illustrate with a metaphor, such process could be compared to driving a car by only looking at the rear view mirror (a partial, indirect and delayed input) rather than through the windscreen. The innovation brought by PADGETS consists in opening up the decision support process by integrating it with the activities carried out over Social Media Platforms. This allows to establish a direct link between the decision process and the external world as well as to reason on fresh and relevant information. This, once the necessary organizational processes are in place, should contribute to produce a much more responsive and effective style of decision making in Government. Going back to our metaphor, the innovation introduced by the Decisions Support Component of PADGETS aims at allowing decision makers to drive looking through the windscreen supported by an intelligent navigation system able to anticipate some of the obstacles lying ahead (i.e., the predictive functionalities of the simulation module).

Finally, it is important to discuss also some of the limitations that characterize the solution presented, as they may represent an interesting starting point for future research. The resampling activity used for the generalization of the results in terms of interest and acceptance, for example, contributes to decrease some of the biases inherent in Social Media usage (e.g., age distribution) but it is far from producing a statistically significant representation of society. In addition, the implementation of a meaningful cross-platform tracking systems still presents a number of challenges having to do with identity management. Along these lines, potential criticalities could derive also in case of scarcity of personal information regarding end users due to heterogeneous policies adopted by Social Media Platforms as well as end users' privacy settings: even though the robustness of the Simulation Model has been repeatedly tested in "borderline" use-cases, the paucity of a Minimum Set of Data (basically users' age and gender, that represent key variables on which the clustering procedure is based) may reduce the representativeness of final results and, consequently, could lower the quality of elaborated reports.

Concluding, although far from being error free, it is our firm belief that the framework presented constitutes a significant step ahead in helping policy makers in dealing with the challenges arising from the complexity that more and more may be found in modern societies.

References

1. Bauman, Z.: Liquid Modernity. Polity Press, Cambridge (2000)
2. PADGETS Project, http://www.padgets.eu
3. Sterman, J.D.: Learning in and about complex systems. System Dynamics Review 10(2-3), 291–330 (1994)
4. Courtney, J.F.: Decision making and knowledge management in inquiring organizations: toward a new decision-making paradigm for DSS. Decision Support Systems 31(1), 17–38 (2001)
5. Beers, P.J., Boshuizen, H.P.A.E., Kirschner, P.A., Van den Bossche, P.: Decision-support and Complexity in Decision Making. In: 5th Junior Researchers of EARLI (JURE) Conference, Amsterdam, The Netherlands (2002)
6. Nonaka, I.: A dynamic theory of organizational knowledge creation. Organization Science 5(1), 14–37 (1994)
7. Brabham, D.C.: Crowdsourcing as a model for problem solving: An introduction and cases. Convergence: The International Journal of Research into New Media Technologies 14(1), 75–90 (2008)
8. Benčina, J.: Web-based Decision Support System for the Public Sector Comprising Linguistic Variables. Informatica 31, 311–323 (2007)
9. Kamel, S.: Decision Support Systems and Strategic Public Sector Decision Making in Egypt. Paper No. 3, Working Paper Series in Information Systems for Public Sector Management, Institute for Development Policy and Management, Manchester, UK (1998)
10. Osimo, D.: Web 2.0 in Government: Why and How. In: JRC Scientific and Technical Reports, European Commission, Joint Research Centre, Institute for Prospective Technological Studies (2008)
11. Charalabidis, Y., Gionis, G., Ferro, E., Loukis, E.: Towards a Systematic Exploitation of Web 2.0 and Simulation Modeling Tools in Public Policy Process. In: Tambouris, E., Macintosh, A., Glassey, O. (eds.) ePart 2010. LNCS, vol. 6229, pp. 1–12. Springer, Heidelberg (2010)
12. Lévy, P.: Collective intelligence: Mankind's emerging world in cyberspace. Perseus Books, Cambridge (1997)
13. Shim, J.P., Warkentin, M., Courtney, J.F., Power, D.J., Sharda, R., Carlsson, C.: Past, present, and future of decision support technology. Decision Support Systems 33(2), 111–126 (2002)
14. Rogers, E.M.: Diffusion of Innovations. Free Press, New York (1983)
15. OECD: Policy Brief: Engaging Citizens Online for Better Policy-Making. OECD Observer (2003)
16. OECD: Citizens as Partners: Information, Consultation and Public Participation in Policy-Making. OECD Publishing (2001)
17. European Commission: Population Policy Acceptance Study – The Viewpoint of Citizens and Policy Actors Regarding the Management of Population Related Change. Final report, EU Research on Social Sciences and Humanities, Brussels (2006)
18. Forrester, J.W.: Industrial Dynamics. The MIT Press, Cambridge (1961)
19. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (2008)
20. W3C's Platform for Privacy Preferences (P3P), http://www.w3.org/P3P

XHTML with RDFa as a Semantic Document Format for CCTS Modelled Documents and Its Application for Social Services

Konstantin Hyppönen, Miika Alonen, Sami Korhonen, and Virpi Hotti

University of Eastern Finland, School of Computing
P.O.B. 1627, FI-70211, Kuopio, Finland
{Konstantin.Hypponen,Miika.Alonen,Sami.S.Korhonen,Virpi.Hotti}@uef.fi

Abstract. For achieving semantic interoperability, messages or documents exchanged electronically between systems are commonly modelled using standard specifications, such as the UN/CEFACT CCTS (core components technical specification). However, additional requirements, such as the need for layout markup or common metadata for certain archiving scenarios might be applied to the documents. Furthermore, the management of resulting artefacts, i.e., core components, XML schemas and related infrastructure, could be cumbersome in some cases. This paper investigates the use of the W3C XHTML+RDFa (extensible hypertext markup language with resource description framework attributes) for representing both the layout and semantics of documents modelled according to CCTS. The paper focuses on the validation of XHTML+RDFa documents against a core components library represented as an ontology. In addition, the paper illustrates and validates this demand-driven solution in the scope of the Finnish National Project for IT in Social Services.

1 Introduction

For a long time, documents have been the main information exchange mechanism in public administration. As the service processes get automated, documents become replaced or aided by electronic message exchange. The standard requirement for the electronic messages is that they must convey necessary semantic meaning of their contents in a form understandable by the information systems involved.

In many cases, however, additional requirements are applied to the message exchange system, such as the generation of human-readable versions of the messages, or archiving them according to a certain archiving plan. This might require the addition of layout markup and certain document management metadata to the messages.

Nešić [12] defines a semantic document as a "uniquely identified and semantically annotated composite resource". The document is a content unit (CU) built of smaller CUs, which in turn can be either composite or atomic. Every CU

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 229–240, 2012.

should be understandable by both humans and machines. Furthermore, the content of semantic documents should be completely queryable, with addressable elements (i.e., CUs) of different granularity.

Message exchange can be modelled using standardised specifications [13,14,18] which describe processes and models for designing common building blocks used in different messages. The specifications also include standard ways of designing XML schemas for business documents and information entities by reusing these building blocks [13,17,19]. The resulting schemas are purely semantic-oriented and are targeted specifically at message exchange between information systems. Therefore, they do not include provisions for any layout markup. Human-readable versions of the messages must be generated using separate style sheets or other mechanisms.

On the contrary, document formats used for preparing documents in common text processors (such as OOXML or ODF) or for publishing the documents online (such as HTML or PDF) concentrate mostly on layout features. They are not suitable as such for information exchange, as they lack precise semantic markup. However, semantic markup and metadata can be added to them using annotation tools, thus bringing semantics and layout markup together. Storing semantic annotations inside a document usually requires the extension of the document format schema, which is not always possible. Uren et al. [20] define requirements for annotation tools and provide a summarizing overview of them with regard to these requirements. Eriksson [4] describes a technology for ontology-based annotation of PDF documents. Individuals of ontology classes are created from highlighted areas of PDF documents. A bidirectional link is then established between the individual in an ontology and the PDF annotation.

Decker et al. [3] argue that flexible semantic interoperability cannot be achieved with XML, and suggest using RDF (resource description framework) based technologies for defining the semantics of information exchange. Combinations of XML and RDF technologies are also available. A knowledge model of the business domain could be represented in a form of ontology and linked to the schemas, for adding more precise semantics to schema elements. For example, the W3C recommendation SAWSDL (semantic annotations for the web services description language) defines a way of providing semantic annotations to WSDL and XML schema constructs [5]. Among other mechanisms, annotations can be implemented as links to ontology classes.

W3C provides a recommendation for a way to embed semantic markup in XML (e.g. XHTML) documents [1]. RDFa (RDF attributes) is a specification of attributes for adding semantic metadata to any markup language. RDFa can be used in conjunction with XHTML for extending the basic layout-oriented HTML markup with semantic elements. However, the applicability of XHTML and RDFa for message exchange designed with standard modelling methods such as the UN/CEFACT Core Components Technical Specification (CCTS) [18] has not been examined closely.

Our results. We propose the use of XHTML+RDFa as a document format (carrier) for messages modelled according to the CCTS specification. In addition, we outline the structure of the validation service which ensures that the documents follow their defined structures, and performs other document type specific checks. A proof-of-concept implementation of the validation service is described. The applicability of this approach is examined in connection with requirements for the document exchange needed in Finnish social services.

The rest of the paper is structured as follows. Section 2 introduces the requirements for documents used in social services and provides an overview of the core components types defined in CCTS. Section 3 explains how core components can be represented in RDF, and Sect. 4 describes a validation service for XHTML+RDFa documents with CCTS-based artefacts. Section 5 discusses the pros and cons of our approach, and Sect. 6 provides concluding remarks.

2 Documents in Social Services

One of the goals of the IT project for social services in Finland (Tikesos[1]) is to model all the document types that are used in Finnish social services. Document structures have been defined by the social service experts on the content level. The social services system in Finland uses about 200 different types of documents [11]. The documents are created in standard social work, where they are used as part of social service processes. As the processes get automated, the document contents are analysed and modelled for document processing systems. The documents can be used for information exchange between systems.

In addition to information exchange, one of the main requirements is to archive all documents for future use. This is demanded by the Finnish Archive Law [6], which is applied to all public authorities. A centralised archive is currently being developed for storing all social services documents in Finland [11]. When implemented, the archive will to be accessed by social services information systems, for archiving the documents (as shown in Fig. 1) and fetching them later if needed. We note that the archiving requirement is of utmost importance, as the future IT infrastructure for social services in Finland is built around the centralized archive.

There are no international de jure or de facto standards for a document format for social services. Therefore, we set out to define a document format suitable for our needs. The following basic requirements are placed on the document format for Finnish social services:

1. A nationally defined set of metadata must be implemented.
2. Content units of different granularity should be marked in the document, so that they are addressable.
3. It should be easy for an information system to parse the document contents.
4. A human-readable version of the document should be easily produced.
5. The validation of the document structure and content should be possible, according to validation service requirements.

[1] www.tikesos.fi

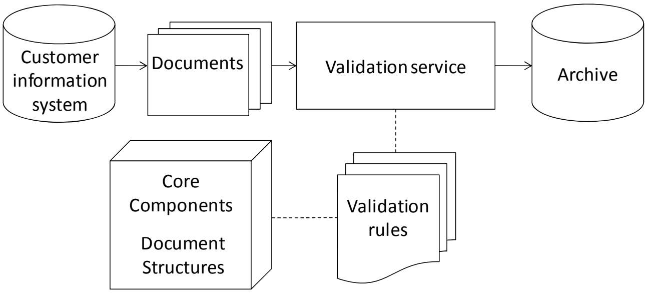


Fig. 1. The centralised archive validates and stores documents produced by customer information systems

The common building blocks of the documents used in social services were identified, analysed and modelled according to the UN/CEFACT CCTS [18]. The CCTS model was selected because it is widely used in Europe for similar projects [8], and has the status of an ISO standard. We provide here a short introduction to the concepts defined in CCTS. Only the concepts relevant to this paper are described.

CCTS defines a method for designing of common semantic building blocks called core components. The components form a language used in information exchange by business partners. A number of different core component types (CCTS artefacts) is defined (see Fig. 2).

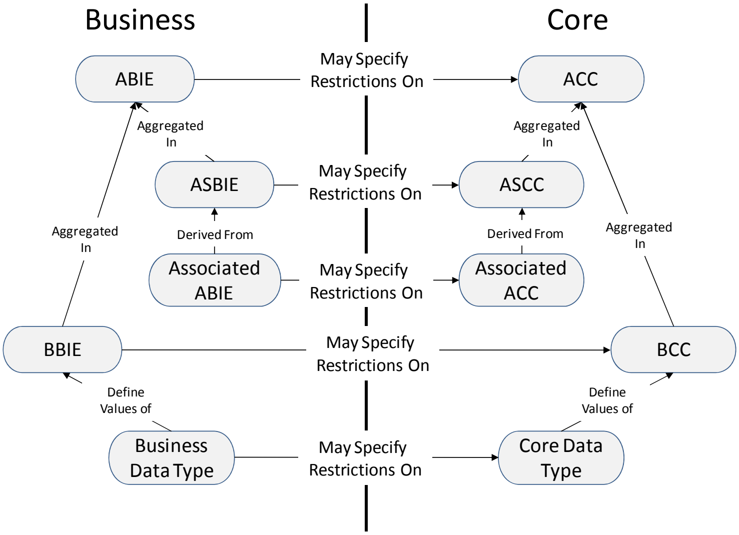


Fig. 2. Relationships of different types of core components (figure from [18])

An Aggregate Core Component (ACC) is an object class describing a concept with clearly defined semantics. The object class contains a number of properties related to this concept. An example of an ACC is Address.

A Basic Core Component (BCC) is a property appearing in a certain object class (ACC). BCCs are based on simple data types (called Core Data Types) such as Text, Number, Amount. For example, the street name in an address can be represented as a BCC.

An Associated Core Component (ASCC) is a property appearing in a certain object class (ACC). The property is based on another object class (another ACC). For example, the validity period of an address could be represented as an ASCC, based on the ACC Period.

An Aggregate Business Information Entity (ABIE) is an object class which is a qualified version of an ACC. It may have a narrower meaning than the parent ACC. Moreover, any property of the ACC may be qualified in the ABIE. An example of an ABIE is Trade_ Address (an address representation used in trade-related business documents).

Basic Business Information Entities (BBIE) and Associated Business Information Entities (ASBIE) are properties of ABIE, constructed by qualifying the corresponding BCC and ASCC properties of the parent ACC.

The resulting library of core components defined for social services in Finland has around 200 different object classes (ACCs and ABIEs), with about 1000 properties in them altogether. The core components are used in some 200 different documents, split in 15 groups (applications, decisions, notifications, agreements etc.).

Although XML schemas for all business documents can be produced relatively easily, their applicability for our requirements is questionable. For instance, it is rather difficult to handle documents based on 200 different schemas in a single archive. The systems which access the archive must support these schemas in order to parse the document contents. Furthermore, document structures tend to change from time to time, and with the updated versions the number of different schemas can easily reach thousands in a span of a few decades. The documents based on old schemas remain in the archive and should be accessible by users, at least as human-readable versions. Producing human-readable document outputs can be also troublesome, as the layout information is not included in documents based on standard CCTS-compliant schemas, and separate schema-specific style sheets are needed. A separate human-readable output generator could be implemented within the archive. However, the generator must handle all the different schemas on which the documents in the archive are based.

3 From Core Components towards RDFa

CCTS does not define a way in which core components should be stored. UN/CEFACT distributes their international core components library in a table format, and provides specifications for the representation of business information entities as XML schemas. We present a transformation to an RDFS/OWL representation from any core component library. The representation is aimed at simplifying the generation and validation of semantic XHTML+RDFa documents using existing core components. Business parties can model their documents according to the standard CCTS method, and use XHTML with RDF attributes as the data exchange medium.

The RDFS/OWL representation of a core component library is generated in a two-step process. First, a general XML serialisation of the library is produced, based on a custom schema. Second, the serialisation is transformed to the RDFS/OWL representation through an XSLT script. The script transforms core components into RDF descriptions that are subclasses of the following OWL types: owl:class, owl:objectProperty and owl:datatypeProperty (see Fig. 3). Unique URIs for RDF descriptions are generated from CCTS names of core components (qualifiers, object class terms, property and representation terms).

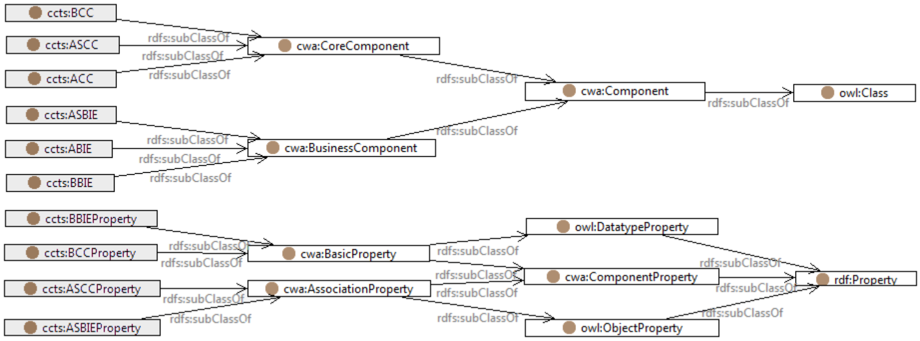


Fig. 3. CCTS artefacts represented as OWL classes. The classes showed on grey background are not in the model designed for validation, but may be used for more precise representation of the CCTS model.

Core component cardinalities are represented in each corresponding OWL class as property restrictions. CCTS cardinalities maxOccurs and minOccurs are represented as OWL properties: owl:maxCardinality, owl:minCardinality and owl:cardinality in conjunction with owl:Restriction as shown in Table 1.

In our model, most core data types are transformed into corresponding XML schema data types. However, if a data type contains additional elements (data

Table 1. Core components cardinalities represented as RDF/OWL restrictions

maxOccurs	minOccurs	Restriction	owl:Restriction
1	1	exactly 1	owl:cardinality
unbounded	k	Min k	owl:minCardinality
1	0	Max 1	owl:maxCardinality
n	k	Max n, Min k	owl:maxCardinality, owl:minCardinality

type attributes in CCTS such as code list identifier in the Code data type), it is represented as a class. Possible values for basic and associated core components are defined in properties as rdfs:range.

Definitions and examples for the core components are represented through the introduction of properties ccts:definition and ccts:example. Additional properties for useful information could be added accordingly. The resulting model is stored as a business ontology for its future use in the validation service.

It is also possible to create a CCTS specific OWL model which contains exactly the same information as the original CCTS model [2]. This type of model can be used for the construction of core components and business documents, including the automatic generation of XML schemas. However, in the case of XHTML+RDFa validation such model is not necessarily needed.

4 Validation of XHTML+RDFa Business Documents

Social services processes include a substantial amount of processing rules, part of which is visible on the document level. For example, there can be conditionally mandatory parts of the document, with the condition defined by code values. Documents could also include some unconditionally mandatory fields, such as the social security number of the customer. Most of such rules and checks can be implemented on the form level. However, as there could be several different implementations of the same forms, it must be ensured that invalid documents cannot be submitted to the archive. A validation service is a part of the archive that checks incoming documents against a number of rules. In addition to the standard validation of the document structure against its schema, the validation service might check the following:

1. Correctness of code values and code lists used in the document (similar to UBL code list value validation [14])
2. Conditionally mandatory fields and other rule-based checks
3. Adherence to the basic formatting requirements

The validation of business documents is based on the closed world assumption [15] because a document is in fact a snapshot of the information collected on a certain clearly defined issue at a single moment. Although more information may be available elsewhere, it is out of the scope of the document and should not be considered in validation. There have been several proposals for adding integrity constraints into OWL by defining semantics for the constraints [10,16]. Standard OWL reasoning cannot be used for strict validation because restrictions are used for inferring new information rather than for checking integrity constraint violations.

Our solution for the validation of XHTML+RDFa business documents works as follows:

1. validate the XHTML structure
2. extract the RDF graph from the XHTML markup enriched with RDFa attributes

3. compare the resulting graph with the corresponding RDFS/OWL schema
4. perform additional rule-based checks.

For our purposes, a top-level metamodel is created to separate the classes and properties used for validation (classes in namespace cwa in Fig. 3) from standard OWL structures. The metamodel defines new types for classes and properties that are used in validation to check whether the extracted RDF graph is consistent with the corresponding RDFS/OWL schema. The closed world constraint validation service depicted in Fig. 4 can check XHTML+RDFa documents for unknown predicates and classes, unexpected namespaces, ill-formed literals and cardinality constraints.

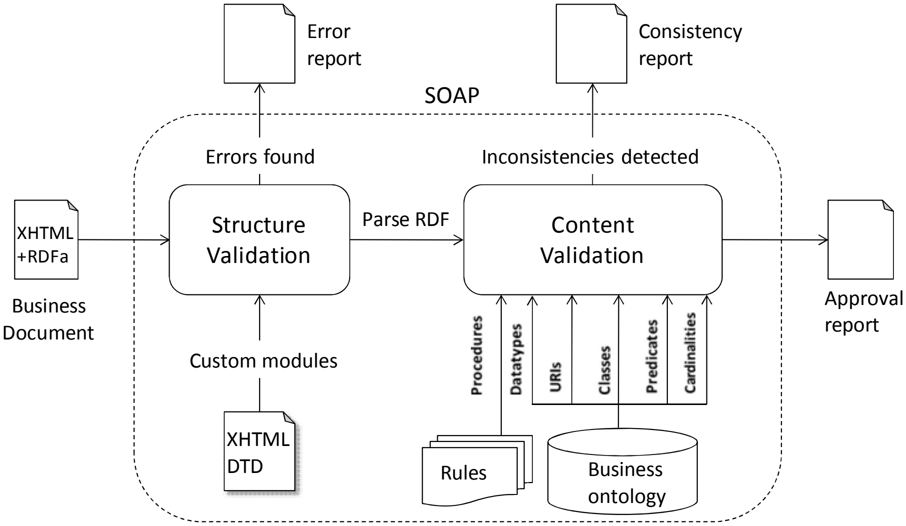


Fig. 4. Closed world constraint validation service

XHTML is based on a modular framework [9] that can be used to restrict or extend the language. The document can be validated against a custom schema or document type definition (DTD) for XHTML. For instance, the scripting module can be left out, as it is rarely relevant for business documents. In addition, the requirements for archiving might prohibit any references to external resources, as these can get outdated after the document has been archived.

An XHTML+RDFa document is first sent to the validation service wrapped in a SOAP request. The document is then validated against the (potentially customised) XHTML+RDFa DTD. For documents with impaired basic XHTML structure, a SOAP response with relevant error details is generated.

After the validation of the basic document structure, the document is accepted for the content validation, where the embedded RDF graph is extracted from it. The extracted RDF triples are then inspected against the business ontology which contains the core components, business information entities and business document structures.

The content validation is based on a set of inferencing rules and custom modules implemented using the Jena framework [7]. The triples extracted from the document are merged with the business ontology, and the validation rules are applied to the resulting model through inferencing. The inferencing rules for validation are implemented in the Jena rules language. Jena was selected mostly because of its licensing terms and speed; other rule engines could have been used instead. Most rules are designed to detect incorrect references between components (such as using a wrong BBIE in an ABIE), check the cardinalities and data types of literals, but there is also a number of more general rules, such as checks for unknown namespaces. As an example, the following rule is used for data type checks:

```
[DatatypeRangeError:
   (?documentFragment cwa:isFragmentOf ?s)
   (?s ?p ?o)
   (?p rdf:type cwa:BasicProperty)
   (?p rdfs:range ?range)
   notCastableAs(?o, ?range)
   makeSkolem(?errorNode,?p,?o,?range)
->
   (?errorNode rdf:type cwa:Failure)
   (?errorNode cwa:FailureMessage "Datatype error")
   (?errorNode cwa:onProperty ?p)
   (?errorNode cwa:found ?o)
   (?errorNode cwa:expecting ?range)
]
```

This rule infers several new triples describing a node of type `cwa:Failure` if the content is based on a wrong data type. For example, an ill-formed date in a business document would create the following new triples:

```
_:X1 cwa:expecting xsd:date
_:X1 cwa:found '2011-AUGUST-02'
_:X1 cwa:onProperty sos:endDate
_:X1 cwa:FailureMessage 'Datatype error'
_:X1 rdf:type cwa:Failure
```

At the end of the validation process, the inferred model is queried for failures. A report is constructed and returned as a SOAP response. The report contains either a list of failures or the approval of the document. A business document is considered valid if its basic XHTML structure is correct and the document is semantically consistent. A document with about 200 triples is validated in some 300 milliseconds; performance could be additionally improved by storing supported business document schemas in RAM.

5 Discussion

Compared with document exchange based on document type specific XML schemas, our approach has a number of advantages. We list the advantages and illustrate their impact on document exchange and document management.

First, layout information is integrated in the document. There is no need to develop and maintain separate scripts (e.g., XSLT) for producing human-readable versions of the documents. Instead, documents can be viewed in any web browser. At the same time, separate CSS style sheets can be used for fine-tuning the layout, or additional scripts written for converting the documents to other formats, such as PDF/A, which is better suitable for long-time archiving.

Second, all documents are based on a single schema (XHTML+RDFa DTD). This simplifies the management of documents in a single archive used by a number of different information systems. In the Finnish social services system the use of separate schemas for separate document types would mean that the documents in the archive would be based on some 200 schemas, not counting their versions. It is a strong requirement for a single information system to implement support even for a part of them. In the case of a single schema, the system will be able to show the document to the user without resorting to display format generation services, even if it cannot process the semantic markup.

Third, the version management of document structures is easier, because semantics are clearly separated from the document container. A new version of a document does not require a new version of a schema. In a purely schema-based system a new version of a commonly used core component may influence a number of documents and, in turn, force the generation of new versions for their schemas. We note that the management of the core component library itself does not become easier. However, there is no need to keep track of schema namespace changes spawned by the versioning of business information entities.

Finally, semantic markup can be applied only to the information that has clear impact on the automatic processing of information. We noticed that in some social services processes the amount of such information is rather low. This information can be stored as pure XHTML, and semantic markup may be added gradually in the future if the need arises.

A drawback of this approach is that the validation of the document structures becomes more difficult. In the previous section, however, we showed how this drawback can be addressed with closed world validation based on semantic technologies.

There is also a number of new requirements placed on the information systems that need to exchange information. They must produce documents with XTHML+RDFa markup instead of somewhat more straightforward generation of simple XML document instances. Importing the information from a document is also different, as it is not based on parsing the XML structure. Instead, the RDF graph stored in the document must be processed and imported. We note, however, that tool support for such functionality is currently sufficient. Furthermore, the extracted RDF graph can be even transformed to a more traditional XML representation, usable in legacy applications.

6 Conclusion and Future Work

When there is need for the interoperability of several systems, such as customer information systems, document type specific XML schemas are often used. However, if the number of schemas is big, or schemas are updated often, their management might become troublesome. In addition, scripts (e.g., XSLT) for producing human-readable versions of the documents and other schema-specific infrastructure have to be maintained. In the paper, we present a solution for exchanging CCTS-modelled documents using the XHTML+RDFa document format. In social services applications, where the exchange of human-readable content is sufficient in many cases, our solution can facilitate the incremental addition of semantic markup to the documents. Still, the validation of semantic markup which represents CCTS artefacts is possible to aid the interoperability of different systems.

The feedback received from customer information system developers was more in favour of the traditional schema-based approach. However, the system developers found some aspects of XHTML+RDFa, such as the change management model and the possibility for gradual addition of semantic markup, beneficial. The impact of the proposed solution on other aspects of the social services IT infrastructure has still to be analysed in more detail. We assume that the switch to XHTML+RDFa influences the design of at least the following services: end-user interfaces, code list server, digital signatures, statistics and research services, and message communication protocols. Currently, the use of XHTML+RDFa in Finnish social services is still under consideration.

We implemented a prototype of a validation service for XHTML+RDFa documents with CCTS-based content. The service reports problems in both the basic XHTML structure and CCTS content. We plan to develop the service further to improve its report generator, as currently the service does not indicate the line numbers in which ill-formed RDFa attributes and values are located. In addition, we plan to investigate the possibilities of using XHTML+RDFa documents with CCTS-based content in the implementation of business rules.

Acknowledgement. This work was supported by the Tikesos project (National Project for IT in Social Services).

References

1. Adida, B., Birbeck, M., McCarron, S., Pemberton, S.: RDFa in XHTML: Syntax and processing. a collection of attributes and processing rules for extending XHTML to support RDF (2008), http://www.w3.org/TR/2008/REC-rdfa-syntax-20081014
2. Biersteker, H., Hodgson, R.: The Netherlands Ministry of Justice Metadata Workbench: Composing XML message schemas from OWL models. Enterprise Data Journal (May 2010), http://www.enterprisedatajournal.com/article/netherlands-ministry-justice-metadata-workbench-composing-xml-message-schemas-owl-models.htm

3. Decker, S., Melnik, S., van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., Erdmann, M., Horrocks, I.: The semantic web: the roles of XML and RDF. IEEE Internet Computing 4(5), 63–73 (2000)
4. Eriksson, H.: The semantic-document approach to combining documents and ontologies. International Journal of Human-Computer Studies 65(7), 624–639 (2007)
5. Farrell, J., Lausen, H.: Semantic annotations for WSDL and XML schema (2007), `http://www.w3.org/TR/2007/REC-sawsdl-20070828/`
6. Finnish Government: Archives Act (831/1994 Arkistolaki) (1994)
7. Jena: A semantic web framework for Java, `http://jena.sourceforge.net/`
8. Laudi, A.: The semantic interoperability centre europe: Reuse and the negotiation of meaning. In: Charalabidis, Y. (ed.) Interoperability in Digital Public Services and Administration: Bridging E-Government and E-Business, pp. 144–161. IGI Global (2010)
9. McCarron, S., Ishikawa, M.: XHTML 1.1 - module-based XHTML. W3C Recommendation, 2 edn. (November 2010), `http://www.w3.org/TR/xhtml11/`
10. Motik, B., Horrocks, I., Sattler, U.: Bridging the gap between OWL and relational databases. Web Semantics: Science, Services and Agents on the World Wide Web 7(2), 74–89 (2009)
11. Mykkänen, J., Hyppönen, K., Kortelainen, P., Lehmuskoski, A., Hotti, V.: National interoperability approach for social services information management in Finland. In: Charalabidis, Y. (ed.) Interoperability in Digital Public Services and Administration: Bridging E-Government and E-Business, pp. 254–278. IGI Global (2010)
12. Nešić, S.: Semantic document model to enhance data and knowledge interoperability. In: Devedžić, V., Gaševič, D. (eds.) Web 2.0 & Semantic Web, vol. 6, pp. 135–160. Springer, US (2009)
13. NIEM Technical Architecture Committee (NTAC): National information exchange model naming and design rules. Version 1.3 (October 2008), `http://www.niem.gov/pdf/NIEM-NDR-1-3.pdf`
14. OASIS Universal Business Language (UBL) TC: Universal Business Language v2.0 (December 2006), `http://docs.oasis-open.org/ubl/os-UBL-2.0/`
15. Reiter, R.: On closed world data bases. In: Gaillaire, H., Minker, J. (eds.) Logic and Data Bases, pp. 55–76. Plenum Press, New York (1978)
16. Sirin, E., Tao, J.: Towards integrity constraints in OWL. In: Proceedings of the Workshop on OWL: Experiences and Directions, OWLED 2009 (2009)
17. Thompson, H.S., Beech, D., Maloney, M., Mendelsohn, N.: XML schema part 1: Structures second edition. W3C Recommendation (October 2004), `http://www.w3.org/TR/xmlschema-1/`
18. United Nations. Centre for Trade Facilitation and Electronic Business: Core components technical specification. Version 3.0 (September 2009), `http://unece.org/cefact/codesfortrade/CCTS/CCTS-Version3.pdf`
19. United Nations. Centre for Trade Facilitation and Electronic Business: XML naming and design rules technical specification. Version 3.0 (December 2009), `http://unece.org/cefact/xml/UNCEFACT+XML+NDR+V3p0.pdf`
20. Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F.: Semantic annotation for knowledge management: Requirements and a survey of the state of the art. Web Semantics: Science, Services and Agents on the World Wide Web 4, 14–28 (2006)

Balancing System Expressivity and User Cognitive Load in Semantically Enhanced Policy Modelling

Christos Tsarouchis, Declan O'Sullivan, and David Lewis

Knowledge & Data Engineering Group (KDEG),
School of Computer Science & Statistics,
Trinity College, Dublin, Ireland
{tsaroucc,Declan.OSullivan,Dave.Lewis}@cs.tcd.ie

Abstract. Policy Engineering is the process of authoring management policies, detecting and resolving policy conflicts and revising existing policies to accommodate changing resources, business goals and business processes. Policy authoring involves developing a policy rule base populated with policies that specify where actions specified on behalf of subjects may or should be performed on targets (resources). Policy engineering can be a daunting process in terms of complexity of the subjects, targets and actions used in rules, and the potential for conflicting policy rules to be committed to the rule base. Semantic modelling of rule elements can help automate the detection of such conflicts, but the additional layers of abstractions may themselves add to the complexity faced by the policy author. In this paper, we aim to assist the policy author by increasing the system expressivity with semantics, while at the same time minimizing the perceived cognitive load due to additional model complexity. We build on work with abstractions aimed at achieving this goal in the modelling of organisational grouping for subjects of policy-rules, which supports specific authority-based group abstractions to ease the maintenance of the subject model in the face of frequent organisational change. In this paper, we study this approach as used in combination with description logic based modelling of target semantics plus logic programming assertions across subject, targets and actions. This is performed through detailed analysis for policy authoring deliberations observed in a user evaluation of these modelling techniques.

Keywords: Policy engineering, semantic modelling, description logic semantics, policy rule semantics.

1 Introduction

Policy Engineering is the process of authoring policies, detecting and resolving policy conflicts and revising existing policies to accommodate changing resources, business goals and business processes [1]. Policy engineering can be a daunting process in terms of complexity of the subjects, targets and actions used in rules, and the potential for conflicting policy rules to be committed to the rule base. The policy author's cognitive load [2] can be stressed in an environment of fluid organizational structure and changing resources, which are the targets for policy enforcement.

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 241–252, 2012.

In this paper, we aim to assist the policy author by increasing the system expressivity with semantics, while at the same time minimizing the perceived cognitive load due to additional model complexity. Semantic modelling of rule elements can help automate the detection of system conflicts, but the additional layers of abstractions may themselves add to the complexity faced by the policy author.

Other research initiatives have attempted to solve such problems by combining the efficiency and the expressive power of rule-based systems with the semantic richness found in Description Logics [3]. For instance, a rule-based system can express variables and causality relationships whereas ontologies on the other hand, can represent complex relationships between entities, such as the relationships of equivalence and disjoint, in a taxonomic manner that could simplify maintainability.

In this paper, for the modelling of policy subjects, the rule-based Community-based Policy management (CBPM) [4] will be used. This integration falls into the category of combining Description Logics (DL) [5] and Logic Programs (LP) [6].

The structure of this paper is as follows: for the remainder of section 1, the benefits of semantically enhanced policy subjects and targets, as well as the integration of CBPM with DL, and the resulting syntax will be presented. In section 2 a case study based on the semantic encoding of the Trinity College Statutes, together with some examples, will be shown. An intervention analysis as part of a user trial series is presented in section 2.2. Section 3 discusses the paper findings, while in section 4, related and future work will be presented.

1.1 Semantically Enhanced Policy Subjects and Targets

One of the requirements in policy engineering is the need to have an accurate and up-to-date view of the managed resources as well as of the actions that can be performed on these resources. This is particularly challenging in the case when not only these resources are frequently changing but also when their corresponding actions change as well.

This can be achieved by having an up-to-date snapshot of the resources and their corresponding actions in question into a structured, manageable and easy to query knowledge base (KB). This KB can be represented using ontological representations. This is because the use of ontologies can provide a centralized form of knowledge aggregation, when even complex relationships can be described and queried. Using OWL DL [7] for instance, the "equivalent" and "disjoint" relationships between resources and actions can be described. An example application that showcases such a system is presented in [8].

When modelling organizational structures, the flow of authority between subjects and targets needs to be modelled and preserved. Large organizations have decision-making groups, which are usually hierarchical in nature and often delegate the decision authority to other groups lower in the hierarchy. To model such concepts the use of the ontological class-subclass relationship is not adequate. For example, a company's Director has authority over the Project Lead, but the latter is not a subclass of the former. In reality however, decision-making groups are more fluid and overlapping than the classic hierarchical organizational models that underpin role-based schemes. For this purpose, this research will use the Community-based Policy Management (CBPM) [4] framework to model policy subjects, which has been shown to model the organizational aspects of policy subject models in a way that makes the resulting policy base more robust to organizational change [9].

CBPM uses the notion of Community as the primary grouping abstraction with the aim of allowing groups within the organization itself to define communities to naturally reflect the changing nature of decision making (i.e. policy setting) authority. Communities towards the top of the hierarchy have the wider membership and more general function, while those toward the bottom have more narrow membership and more specific function. The hierarchy is designed to support agility and autonomy, allowing new sub-communities to be formed and encouraging the delegation of decision making authority as far down the hierarchy as possible.

Using CBPM enables us to have a powerful permissions scheme in place, since the flow of authority from one community to another is explicitly defined. This scheme grants or denies permission to perform actions on resources and it can also be used to define obligations.

In this paper we therefore aim to investigate how different combinations of DL and LP impact on the overall system expressiveness and on the cognitive load placed upon the requirements engineer, while aiming for a similar level of policy engineering robustness for target models, which CBPM delivers for subject models.

We will now present in more detail, the inner workings of the integration between CBPM and OWL DL.

1.2 CBPM and OWL DL Integration

The type of integration between CBPM and OWL DL presented in this paper, is of the hybrid type [3], which means that there is no creation of a new logical language. The resulting framework is called CBPM+DL. CBPM+DL is formed as follows: a hybrid knowledge base K = (S,R) is a finite set S of DL axioms in the ontology language S and a finite set of hybrid rules R over R and S, including non-DL atoms. DL atoms are classes, properties, individuals. This definition follows the hybrid integration definition presented in [3].

Since the CBPM+DL method follows this hybrid approach, the semantics entailed in the hybrid language are derived from the semantics of its constituent components, namely CBPM and OWL DL. A hybrid rule r is obtained from the CBPM rule r′, by adding DL constraints in the body. The latter are in fact additional constraints on the use of the rule r′, which have to be satisfied so that the rule r′ is fired.

B CBPM+DL Syntax

The syntax of CBPM+DL is formed by extending the existing CBPM syntax with ontological predicates. A rule rendering of the existing CBPM syntax will now be presented, in which a rule has the form: antecedent → consequent

The antecedent is a conjunction of atoms written $a_1 \wedge a_2 \wedge \ldots \wedge a_n$. Variables are indicated using a question mark as a prefix (e.g. ?x).

1) Community Membership Rule

The membership rule defines the restrictions, which need to be satisfied, so that a person P is a member of Community C.

Property is of type ontological object property restriction. This means that user-specific properties can be created.

2) Authorization Rule

Authorization is about granting or denying access (negative authorization) to a subject to perform an action on a resource. In this research we use the notion of resource authority, which is introduced in CBPM. A resource authority specifies what action on which resource, access is sought for.

The resource authority is defined as follows:

Action(?a) ∧ Resource(?r) → ResourceAuthority(?ra)

The resource authority is used as a predicate in the authorization rule:

comm(?comm) ∧ isa(?x, ?y) → posAuth(?comm, ?ra)

The isa(?x, ?y) denotes the OWL is-a relationship, and in this case it means that x is a subclass of y. This rule can be read that as long as the "isa" relationship is satisfied, the community comm, is authorized to perform an action on a resource, as specified in the resource authority restriction.

3) Implies authority rule

The implies authority relationship describes the authority which is exercised by a Community, an Action, and a Resource on a Community′, an Action′ and a Resource′ respectively, which are located lower in the hierarchy of authority.

Community ImpliesAuthorityOver Community′
Action ImpliesAuthorityOver Action′
Resource ImpliesAuthorityOver Resource′

′: lower in the hierarchical tree

For example, the Community called *Directors* ImpliesAuthorityOver the Community′ called *Projects_Leads*. The Action *Read_All* ImpliesAuthorityOver *Read _File*. The Resource *Servers* ImpliesAuthorityOver *Database_Servers*.

2 The TCD Statutes Case Study

For the last 3 years the Trinity College Dublin (TCD) Statutes Review Working Party has been commissioned with the task of reviewing and rewriting the Trinity College Dublin Statutes. The main task for the Party is not to introduce new policies but to update and rewrite the existing ones. This well documented, real-world problem of statutes definition and interpretation in order to minimize the number of conflicts poses a big challenge. We decided to study the combination of CBPM and OWL DL techniques in order to tackle the encoding of existing statutes and the detection of conflicts when enforcing new ones. Some of the challenges this effort has, is to capture the complexity of the organizational structure of the College, with the policies and obligations pertaining to each College body.

For this purpose the use of the CBPM was chosen as a candidate to model policy subjects. Also, due to the reasoning power of OWL DL, static conflict analysis could be performed and this was one driving factor for using ontologies.

However, the main challenge that we faced was finding the right balance between the use of CBPM and OWL DL in terms of system expressivity without incurring a big penalty on the user cognitive load. Since it is envisaged that such a system would be used not only by experts in policy management, but also by domain experts (i.e. experts in law specification in this case), the creation of easy to use, custom tools that support these encodings is considered of great importance.

Such a tool has already been developed and usability studies are being performed. In the remainder of this paper however, we will not focus on the tool support, but on the task of finding a working balance between the various encoding options, during the encoding of the existing statutes. For that purpose, a user trial has been conducted which did not involve the use of any particular tool. This was done with the intention of not skewing the results with potential usability issues the tool has.

Five carefully selected participants took part in the user trial, each with a strong background on either policy-based management, ontologies, or logic programming. This is because the users would be asked to encode statutes using such techniques. The user trial involved a training session introducing all encoding methods, where examples of already encoded statutes were presented, and the actual user statutes encoding session. The whole trial was recorded and in the end the users were asked to fill in a questionnaire.

The users were presented with an "encoding path" which they should follow, in the form of a step-by-step encoding methodology. This was done so that all users irrespective of their background, will consider all encoding options and not jump straight to the one they feel more comfortable with. It was also done so that the users won't be intimidated by the number of encoding options available to them, especially during their first encodings.

The encoding methodology given to the users was the following:

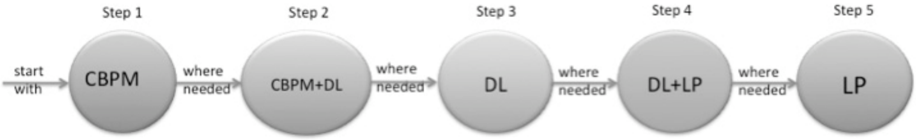


Fig. 1. 5-step encoding methodology

All users started with CBPM, and if it was not possible to encode with CBPM or they were not happy with the outcome, they moved to the next available encoding method, which is CBPM+DL in this case. The same happened until the users traversed all remaining encoding methods, namely OWL DL, OWL DL + LP, and LP.

During the training sessions, the users were informed that there is no single "golden" encoding, and as long as they adhered to the encoding guidelines, they were free to come up with an encoding that could differ with the one suggested by the policy expert.

The OWL DL + LP method was in essence the use of the SWRL language [10]. When selecting LP in this trial, the users had to use the syntax of a generic rule language such as Drools [11].

2.1 Example Statutes Encodings

1st Example

The following example was presented to the users during the training session. The task is to encode the following clause from the TCD statutes:

"The Chancellor shall be a member of the Caput of the Senate and shall preside at its meetings"

Following the encoding methodology, the user has to try to use CBPM first.

The user can search for nouns, which are subjects and as a result can be candidates for being CBPM communities. Likewise, the user can search for targets. For instance, this clause can be broken down as follows:

C NCC C
The Chancellor shall be a member of the Caput of Senate
A R
and shall preside at its meetings.
C:Community, A:Action, R:Resource, NCC: Native CBPM Construct

The "Chancellor" is the subject and therefore it is a Community. We don't know whether the "Chancellor" has authority over the "Caput of the Senate", so the latter cannot be encoded as a Resource. So "Caput of the Senate" is a Community. "Preside" is an Action and "meetings" is a Resource, probably of type "Caput of the Senate" since the clause describes the "Caput of the Senate meetings". The term "Member of" found in this clause, is an indicator that the "Chancellor" can be a member of the "Caput of the Senate" Community, and therefore can be regarded as a Native CBPM Construct.

We are now ready to encode the first part of the clause, namely "The Chancellor shall be a member of the Caput of Senate". We will use the CBPM membership rule to achieve this:

Person(?P) ∧ memberOf(?P, Chancellor) → memberOf(?P, CaputOfSenate)

This clause can be read as follows: For Person P and P is a member of the Chancellor Community, it implies that P is a member of the CaputOfSenate Community.

We can now proceed to encode the second part of the clause, "(The Chancellor) shall preside at its meetings". We can use a CBPM positive authorization rule:

First we define the resource authority, which does not mention anything about the subject:

Resource Authority:
Action(Preside) ∧ Resource(meetings) → ResourceAuthority(presides_meetings)
The rule that uses the given resource authority is the following:
comm(Chancellor) ∧ ResourceAuthority(presides_meetings) → cl.posAuth(Chancellor, presides_meetings)

This rule can be read as follows:

For the Chancellor Community and for the presides_meetings Resource Authority, it implies that the Chancellor is given positive authorization to preside in Senate's meetings.

The encoding of this clause is a good fit for CBPM, because both the membership rule and the authorization rule can be applied easily. The same clause can also be encoded with the other remaining methods in the encoding methodology. For instance, for the OWL DL encoding, the clause breakdown is shown below:

OC OP OC
The Chancellor shall be a member of the Caput of Senate
OP OC
and shall preside at its meetings.

OC:Ontological Class, OP:Ontological Property

Following the OWL DL modelling, “Chancellor”, “Caput of Senate” and “meetings” can be ontological classes. “Member of” and “preside” can be ontological object properties.

Thus, so far we have encoded the same TCD statute clause using two encoding methods, CBPM and OWL DL. The encoding choice is simply a matter of personal preference. However, in some type of clauses, such as those that have conditionals, the encoding process is not obvious, as user trials have shown.

2nd Example

This clause includes a conditional:

“The Chancellor shall act as head of the University on ceremonial occasions”

“On ceremonial occasions” is the conditional.

Following a similar breakdown as in the previous clauses, the “Chancellor” and the “University” can be encoded as Communities and “head of” can be an Action. The question that arises is how to encode the “ceremonial occasions”.

Let’s assume that the condition “Ceremonial occasions” is encoded as a Community. This means that a new Community called Ceremonial is created and will be utilized on Ceremonial occasions.

We can use the CBPM authorization rule to encode this clause. First we create the Resource Authority:

Action(heads) ∧ Resource(University) → ResourceAuthority(heads_university)

The authorisation rule reads:

comm(Chancellor) ∧ memberOf(Chancellor, Ceremonial) ∧ ResourceAuthority (heads_university) → cl.posAuth(Chancellor, heads_university)

The meaning of this rule is that the Chancellor, who belongs to the Chancellor Community and is a member of the Ceremonial Community and for the given Resource Authority, it implies that the Chancellor is authorized to head the University.

The encoding of this rule has shown that by restricting membership into a new Community, such as the “Ceremonial” in this case, statute conditionals can be represented.

However, during the presentation of the same example in the training session, all users agreed that the “Ceremonial” Community is an odd concept, and did not feel natural. Therefore, they were urged to seek alternative methods to encode this clause.

Using CBPM+DL, this clause would be encoded as follows:

cl.comm(Chancellor) ∧ isa(Ceremonial, Event) → cl.posAuth(Chancellor, heads_university)

Using such an encoding there is no need to create the "Ceremonial" community. Instead, using the "is-a" relationship, a more "user-friendly" Ceremonial Class is created, and we define the Ceremonial Class as a subclass of the Class Event.

This CBPM+DL encoding has thus proven to be an efficient encoding method for cases like this.

It is worth noting that using OWL DL to encode and infer conditionals is very hard, and all users avoided that.

3rd Example

Although OWL DL cannot encode complex conditionals, there is one type of statute clause, where it was particularly useful.

"There shall be up to six Pro-Chancellors, who shall be members of the Senate ex officio"

This clause is hard to encode in CBPM. It would require the introduction of a meta-rule (a rule about a rule) that would count the number of existing Pro-Chancellors. The CBPM syntax which is available for this user trial, does not cater for the use of meta-rules.

On the other hand, this is a clause where the OWL max cardinality restriction is ideally suited for its encoding:
Pro-chancellors is max 6

4th Example

The users had to encode the following clause:

"The Chancellor shall be the primary Visitor of the College and University, and in the event of a disagreement between him and the other Visitor the opinion of the Chancellor shall prevail"

One way of encoding this rule is to use a similar analysis as in the previous examples. However, a more holistic view of this clause reveals that it would be a good fit for the CBPM implies authority relationship.

In fact some users selected this method as their preferred one:
Chancellor ImpliesAuthorityOver Other Visitor.

The users have used the "implies authority" relationship to describe the relationship between the "Chancellor" and the "Other Visitor". By doing so, they have ensured that under all circumstances, such as a disagreement between the "Chancellor" and the "Other Visitor", the "Chancellor" always prevails. This encoding has saved the users from encoding every single predicate of the clause. For instance, the term disagreement is not encoded.

This encoding has proven than CBPM when used in such scenarios, can be very beneficial both in saving rule editing, as well as resulting in smaller encodings, which can be more easily understood by a third party.

The use of the "implies authority" relationship in this case, is an elegant solution of avoiding the painstaking task of having to analyse complex clauses.

2.2 User Trials

For this research, to evaluate the user cognitive load caused by policy engineering tasks, a number of metrics are currently used:

- the user prior-knowledge of the encoding method in question.
- the time required to complete all tasks.
- an intervention analysis, which studies the type of interventions performed by the policy expert, so that the users complete all tasks successfully.

These metrics are an indirect indication of the user cognitive load, since they do not include the use of any type of biometric sensor. The metrics currently used were selected on the basis that they represent an easy and unobtrusive method of collecting data. In this paper, only part of the intervention analysis is shown.

2.2.1 Intervention Analysis

There were two main actors when conducting the user evaluation experiments: the users –being experts in either ontologies, CBPM, or rules- were asked to perform a series of tasks and complete a questionnaire in the end, and the policy/domain expert who conducted the experiment. The intervention analysis studies the type of interventions performed by the policy/domain expert, during the experiment, as a means to assist users. The number and type of interventions can be an indication of the user cognitive load. For this user trial, which involved the encoding of TCD college statutes using a number of different encoding methods, there were 5 types of domain expert intervention.

I1: The user doesn't know what to do, and the domain expert gives a hint

I2: The user asks for clarification about the syntax and the domain expert gives a hint

I3: The user asks for clarification about conceptualizing the problem and the domain expert gives a hint

I4: The user doesn't ask for clarification, the domain expert intervenes

I5: The user doesn't ask for clarification, the domain expert doesn't intervene

The intervention made by the domain expert for every user and for each task is shown in Figure 2. This graph shows that during the first task, most of the users didn't know what to do and as a result, the domain expert intervened a lot. This is in stark contrast with the 5th task, where the domain expert intervened the least and three users completed the encoding without any intervention. This has to do with the fact that by the 5th task, the users had up to a certain degree acclimatized themselves with all the different encoding methods. This highlights the importance of user training.

The 1st task was about the application of the CBPM syntax, and as a result most users needed time to familiarize themselves with its syntax.

The encoding of the 2nd task was not an obvious selection, and the users had to decide on their favourite method. This is depicted by the fact that almost all users needed assistance in order to conceptualize the problem, hence the Intervention type 3 is shown as the highest for this task.

The 3rd task was about discovering that CBPM does not support meta-rules in its given syntax, and that users can use the OWL DL cardinality relationship. This is why the intervention type 2, which is related to clarifications about syntax, is shown as the highest.

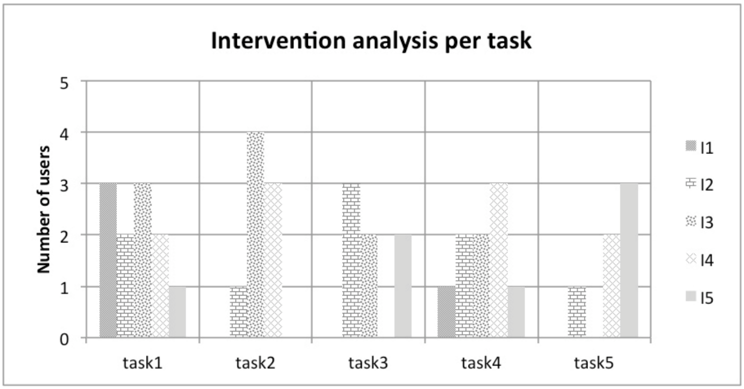


Fig. 2. Intervention analysis per task

The 4th task was about the encoding of a conditional. OWL cannot be used to encode and infer conditionals, but all other methods can be used. Most intervention types were used in this task, since it was the first time the users had to encode a conditional.

The 5th task builds on top of the 4th one, since again it describes a conditional, and that is why there are considerably less interventions needed compared to task 4.

3 Discussion

What has become obvious from analyzing the user responses, is that there is no one-size-fits-all encoding method, which doesn't cause a rule clutter. Obviously, nearly every clause can be expressed in Logic Programs (LP) in the form of generic rules such as [11], but firstly, not all users are comfortable in using and interpreting LP rules and secondly the outcome being in the order of hundreds or thousands of rules, would be hard to maintain. Also, the benefit of having static rule conflict analysis that OWL DL provides, would be missing from an LP only case.

The 5-step encoding methodology used in the trial, does not necessarily describe a path from low expressivity to high expressivity, or from a given set of semantics to a richer set of semantics. This is because in some cases the semantics involved are just different, such as those between CBPM and OWL, and it is generally hard to agree on which one is semantically the richest.

What has been an interesting point of discussion is whether CBPM needs to be the first option for the encoding. This was done primarily because only after repeated use, could the users better utilize CBPM's inherent principles. For instance, some of them used the CBPM "implies authority" relationship to describe a complex conditional.

Describing and inferring conditionals can be regarded as one of OWL DLs weakness, and thus alternative encoding methods must be selected.

Since the aim of the experiment was to showcase the strengths and weaknesses of each encoding method in a controlled environment, the syntax of each of the methods was explicitly defined and was available for the experiment participants to use. For CBPM, meta-rules were specifically omitted, since this is more akin to rule-based systems. Therefore, an alternative encoding method needs to be selected, when meta-rules are required, e.g. the use of OWL DL as shown in the example. Also, user-defined properties were left out of the CBPM syntax, since this is more tailored to OWL DL based encodings. The introduction of CBPM+DL encoding method can be seen as a bridge between two different worlds. Using CBPM+DL in the trials, the users were able to create custom properties and intuitive encodings in the cases where CBPM produced counter-intuitive ones, according to user responses. More on the user's preferences regarding the various encoding methods can be found in [8].

4 Conclusions and Future Work

This research aims to assist policy experts and domain experts in the task of policy engineering, and especially in cases where the complexity of the subjects, targets and actions used in rules is high. The modelling of the Trinity College Dublin statutes poses such a challenge and for that purpose the use of semantically enhanced techniques has been suggested as a viable solution. This modelling involves the creation of a new encoding method called CBPM+DL, which uses CBPM to model organizational subjects and model the flow of authority between different parties. CBPM+DL also suggests the use of OWL DL to model changing policy resources (targets) mainly due to OWL's ability to offer a taxonomical representation of complex resource relationships as well as static policy conflicts checking. For example, OWL DL can express equivalence, and when contradicting policies are enforced onto a class, its equivalent classes can also be signalled as 'in conflict'. This can facilitate conflict detection, since there is no need for the policy author to monitor for conflicts in all equivalent classes.

This paper has shown that the use of CBPM can be beneficial in terms of increasing the overall system expressivity, but unless systematic user training is performed, it is difficult to use in practice, due to increased user cognitive load. For this user trial, the users were informed that there is no 'golden' encoding method, and as long as the outcome is the one intended, any method can be used. Potentially, this can be an issue in collaborative environments, where more than one policy authors is involved and maintaining consistency when encoding similar rules is a requirement.

Currently further trials are being performed, where the user's prior knowledge in relevant technologies is correlated to the user's performance in policy engineering tasks, such as policy conflict detection and resolution. Further evaluations are made to test the efficiency of a custom-built policy engineering tool, capable of modelling in the CBPM+DL form. This tool is based on the Graphical Modelling Framework [12], and its efficiency is compared against Drools-based graphical editors [11].

Related work can be found in [13] where Rei, a policy framework that permits the specification, analysis and reasoning about declarative policies is presented. Rei adopts OWL Lite to specify policies and can reason over any domain knowledge expressed in either RDF or OWL. KAoS [14] defines basic ontologies for actions, actors, groups, places, resources and policies. Using an inference engine, KAoS can reason about policy disclosure, conflict detection, and domain structure. SWRL [10] combines OWL with Horn-like rules. What differentiates the research presented in this paper from these approaches, is the use of Community modelling (CBPM), to model policy subjects, and their interactions, such as the delegation of authority between Communities. CBPM+DL also emphasizes in producing encodings, which are –in some cases- more intuitive than plain CBPM, as user trials have shown, thus assisting in the reduction of the user cognitive load. Since RuleML [15] is based on an XML mark-up language, it is tailored mostly for web use, rather than for the business level. Finally, SBVR [16] can be used in the future to model CBPM+DL concepts, such as Communities, the authority transfer between Communities, as well as the formation of Community Federations. In CBPM, parts of an existing or of different organisations can form a Federation, which is delegated authority over resources.

Acknowledgments. The authors would like to acknowledge the Centre for Next Generation Localisation (CNGL) (www.cngl.ie), for its support.

References

1. Feeney, K., Tsarouchis, C., Lewis, D.: Policies as Signals in Collaborative Policy Engineering. In: Policy-based Autonomic Computing, PBAC (2007)
2. Batra, D.: Cognitive complexity in data modeling: causes and recommendations. Requirements Engineering 12, 231–244 (2007)
3. Antoniou, G., et al.: Combining Rules and Ontologies. A survey (2005)
4. Feeney, K., et al.: Relationship-Driven Policy Engineering for Autonomic Organisations. In: 6th IEEE International Workshop on Policies for Distributed Systems, POLICY 2005 (2005)
5. Baader, F., et al.: The Description Logic Handbook. C.U. Press (2002)
6. Baral, C., Gelfond, M.: Logic programming and knowledge representation. Journal of Logic Programming 19/20, 73–148 (1994)ss
7. OWL-overview, http://www.w3.org/TR/owl-features/
8. Tsarouchis, C., O'Sullivan, D., Lewis, D.: A study in the expressiveness of semantically different policy modelling schemes. In: 6th IFIP/IEEE International Workshop on Business-driven IT Management BDIM, Dublin, Ireland (2011)
9. Brennan, R., et al.: Policy-based integration of multiprovider digital home services. Netwrk. Mag. of Global Internetwkg 23(6), 50–56 (2009)
10. SWRL, http://www.w3.org/Submission/SWRL/
11. Drools, http://www.jboss.org/drools/
12. GMF, http://www.eclipse.org/gmf/
13. Kagal, L.: Rei: A Policy Language for the Me-Centric Project. HP Labs (2002)
14. Uszok, A., et al.: KAoS Policy and Domain Services: Toward a Description-Logic Approach to Policy Representation, Deconflictions, and Enforcement. In: IEEE 4th International Workshop on Policies for Distributed Systems and Networks (2003)
15. RuleML, http://www.ruleml.org/-Scope
16. SBVR. Semantics of Business Vocabulary and Rules, http://www.omg.org/spec/SBVR/

Sensing Presence (PreSense) Ontology: User Modelling in the Semantic Sensor Web

Amparo-Elizabeth Cano, Aba-Sah Dadzie, Victoria Uren, and Fabio Ciravegna

Department of Computer Science, The University of Sheffield,
Sheffield, United Kingdom
firstinitial.surname@dcs.shef.ac.uk

Abstract. Increasingly, people's digital identities are attached to, and expressed through, their mobile devices. At the same time digital sensors pervade smart environments in which people are immersed. This paper explores different perspectives in which users' modelling features can be expressed through the information obtained by their attached personal sensors. We introduce the PreSense Ontology, which is designed to assign meaning to sensors' observations in terms of user modelling features. We believe that the *Sensing Presence (PreSense)* Ontology is a first step toward the integration of user modelling and "smart environments". In order to motivate our work we present a scenario and demonstrate how the ontology could be applied in order to enable context-sensitive services.

Keywords: Linked data streams, semantic sensor web, user modelling, smart objects.

1 Introduction

Digital sensors have pervaded the modern world, and increasingly make up the majority of connected devices, in, e.g., intelligent buildings, traffic lights and in particular in mobile devices. These advances have resulted in "smart" environments, marking an evolution in the generation of information, and the interaction between humans, smart and ordinary devices and sensors. Human-computer interaction now extends to everyday objects attached to the end user or located in their changing environment [1]. Users produce data streams through their mobile devices, wearable and implantable microsensors (e.g., GPS tracklogs, heart rate monitors). These devices now frequently act as the gateway to cyberspace, which is increasingly becoming an extension of the lives of humans in the real, physical world. Therefore, these can provide information regarding a user's physical context (e.g. location, physiological state), in addition to their digital environment (e.g. adding new friends to an online social network, tweeting on an evolving event). This leads to a bond between the user and their mobile devices and sensors, in which the latter act as an extension of the user's identity, providing real-time information that can reveal important user and environmental characteristics.

This provides motivation to explore new techniques for combining current user modelling methods, that depict the digital identity of a given person, with sensor information distributed across the online and physical worlds.

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 253–268, 2012.

The contributions of this paper are as follows: we explore different perspectives in which the attachment of sensor data to user models can impact the derivation of tailored services that feed into users' interaction with smart objects and environments, by providing real-time contextualisation. We propose the *Sensing Presence* (PreSense) ontology as an approach to modelling the attachment of sensor data streams to a user profile, allowing rich, semantic, real-time change in a user's representation. This enables also the integration of observable user features to the linked data cloud [2].

The paper is structured as follows: in section 2 we present the motivation for our work by discussing different perspectives on the use of sensor data for user modelling; in section 2.3 we introduce a scenario that highlights challenges and benefits that the attachment of sensor data to user profiling presents; section 3 discusses existing ontologies that consider sensor data in user modelling; in section 4 we introduce a set of requirements for modelling the attachment of sensor data streams to a user profile; in section 5 we introduce the Sensing Presence Ontology (PreSense). Section 6 describes the application of the ontology in relation to our scenario; finally, section 7 discusses the potential of this work, plans for evaluation, and concludes the paper.

2 Motivation

The relevance of users to the Sensor Web has been explored from the perspective of users acting as collective sources of information. Goodchild [3] highlights the relevance of the Social Web in Volunteered Geographic Information, where users have created a mesh of global information. Projects like SensorBase[1] and SensorPedia[2] provide a platform for sharing online sensor information within user communities. However, little attention has been paid to the importance of users' sensors as gateways for personal feature information. This section motivates our work by introducing different perspectives from which users engage with the physical and online worlds through sensor data.

2.1 Mobility in the Digital Society

In the past few years, users' online activities, including web browsing, online shopping, and social web media use [4,5], have served as information sources for user modelling. Further, the emergence of compelling social web platforms (e.g. Facebook[3], Twitter[4]) have encouraged users to proactively participate, shaping their online personae and influencing their perception about how they are viewed by others (a.o., [6]).

Social studies on the adaptation of users to online technologies highlight that users appropriate telecommunications technologies in ways that fit their social groups, life stages, sociability and activities [7]. Since mobility has become a central aspect of the digital society, the introduction of location-aware services in social web platforms for mobile devices has received considerable attention from researchers in recent years. Research in this area includes scenarios for emergency response, tracking, navigation,

[1] http://sensorbase.org
[2] http://www.sensorpedia.com
[3] http://www.facebook.com
[4] http://twitter.com

billing and social networking [8,9,10]. Part of the success of these applications is the user's increased dependency on mobile devices; which have become, for some, an indispensable tool. While the use of sensors for registering users' features (e.g. location) has proved to be fundamental in these applications, transient, sensor-based information has, to date, not been considered as an inherent component in user modelling.

2.2 Sensors and Users' Context

Sensors refer not only to physical sensor devices but also to values computed as a result of the composition of indirect or abstract measurements derived from multiple, distributed, often heterogeneous data streams [11,12,13]. Such sensors are usually referred to as *virtual sensors*; they allow the abstraction of data collection away from a fixed set of physical objects. A virtual sensor may define a number of valid sources of information, allowing it to poll for and retrieve information from different sources and at varying levels of granularity.

Following this definition, we consider a *web-based sensor* as an extension of the concept of virtual sensors, in which the measuring computation involves data streams generated from web resources. A web-based actuator may be regarded as a reactive computation that produces a response to a specified event. E.g. NASA Hurricane[5] on Twitter is a data stream of instantly updated information generated from the continuous monitoring of different devices sensing meteorological conditions for predicting hurricanes and tropical cyclones all over the world.

In the same way, personal data streams may be regarded as gateways reporting relevant information for user modelling. The information embedded in these streams involves different users' context. User context is built on static, stable and dynamic contexts. A user's static and stable contexts represent information about or related to a user that does not, or rarely, changes in time, e.g. the relation between a user and their hometown or work place. A user's dynamic context, in contrast, reflects highly changing information, which is often influenced by the environment in which a user is immersed; this includes, e.g., changes in position, anxiety levels while in a traffic jam.

Advances in intelligent, context-aware systems promote a vision of increasingly autonomous and ubiquitous applications that act on proactive knowledge to provide tailored services to individuals. These smart systems must not only support users in static, pre-defined environments, but also adapt to users' changing context and evolving goals. However, the integration of user context and the user's immersed environmental context, taking into account tempo-spatial restrictions, still requires research. We present next a scenario highlighting the role of sensor data streams in a user profile.

2.3 Scenario

Imagine Alice, a *Doctor* working at a public hospital, and Bob, a *Patient* suffering from Type II diabetes and obesity. Bob's treatment combines regular insulin injections with a diet plan. His nutritionist works with Alice to monitor how well he follows the plan, his physical activity and the impact both have on his overall health. Periodic reviews will

[5] NASAHurricane: `http://twitter.com/NASAHurricane`

take this information into account in updating his treatment. Alice must also monitor Bob's blood glucose levels, to determine the suitability of both his diet and medication.

This scenario requires Bob to wear multiple sensors that communicate with Alice over a network. More precisely, it requires the attachment of sensor information to Bob's personal attributes. Given the emergence of sensor-enabled mobile devices, we can imagine that Bob owns a device that connects to the Internet and monitors his location and health [14,15]. In this context, Alice accesses, in real-time, the data generated by Bob's sensors. Should Bob's blood sugar reach a dangerous level, Alice must be able to dispatch emergency assistance to Bob in the most efficient way. Information on his diet is not critical, so is only uploaded periodically.

Let us consider a weekday when Bob is returning to his office after inspecting a construction site with a client. His sensors have recorded higher than usual physical activity and that he missed his usual mid-morning snack. His smartphone warns him of the danger of his blood glucose levels dropping too low. Since it is close to lunchtime his nutrition monitor (NutrApp) polls for suitable eateries between his current location and his office (see Fig. 1**A**). It also checks Bob's online social network for recommendations by friends he often eats out with. The NutrApp polls for the ingredients of meals and portion sizes from virtual sensors, and determines suitability by matching with the requirements of his diet plan. Time to cook is also important – his calendar has posted a reminder about an early afternoon meeting he must prepare for (Fig. 1**B**). By merging online information with GPS the NutrApp will try to locate members of Bob's social network in the neighbourhood, whose calendars or status information show they are available – if any are found Bob will receive a suggestion to invite them to join him.

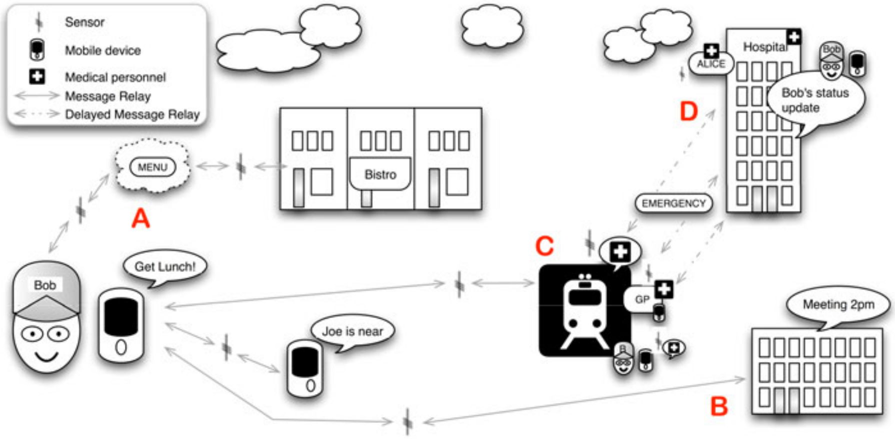


Fig. 1. Schematic for the `PreSense` scenario, illustrating the exchange of data streams between sensors, and the interaction between human actors and other entities as a result

To illustrate an emergency in which situational context is communicated to external actors via sensors, let us consider what happens if Bob ignores the alerts he receives to stop for lunch, because he forgot to carry his medication. He decides to return directly to his office, 45 minutes away via the subway. However, due to a signal failure Bob's train is held stationary just outside a station 1 hour later (Fig. 1C). The stressful situation, combined with more exercise than normal and the time since his last meal, result in a large drop in his blood sugar. Without the sensors recording Bob's blood sugar he may not recognise symptoms of hypoglycaemia till they become severe. In our scenario his smartphone warns him to consume food or drink with high sugar content urgently. His sensors attempt to warn Alice when his blood sugar reaches a critical threshold; however with the train stuck in a tunnel Bob's sensors are unable to connect to Alice's.

Bob's sensors also attempt to locate nearby resources that can help to alleviate his symptoms. A *General Practitioner* (GP) in the next carriage receives the emergency alert. Virtual sensors apply a context-sensitive filter to Bob's medical information (some of which is held on his personal devices). Another virtual sensor calculates Bob's location using GPS and a schematic of the train and transmits this to the GP's mobile device over a (local) wireless network. The GP locates Bob, and armed with the information needed to attend to the semi-conscious patient, successfully handles the emergency.

When the train exits the tunnel the delayed emergency alert is relayed to Alice (Fig. 1**D**), with a timestamp that indicates that it has now expired. An update with more current, valid information on Bob's status is also relayed to Alice over the Internet.

To be effective, this scenario implies the need to connect different information streaming sources in time- and location-constrained situations, via (context-sensitive) virtual sensors. Particularly, it illustrates the demands of attaching streaming information to real world entities such as people – the GP must be able to identify the patient via sensor stream ownership. Wireless networks also play a role in information exchange; in the emergency situation this is how the virtual and physical sensors communicate.

3 Related Work

Ontologies for user modelling follow two paradigms: standardisation- and mediation-based modelling [16]. The first is based on a top-down approach in which ontologies for user modelling are designed to be domain-independent (top-level ontologies), or still high-level but domain-specific (upper ontologies) in order to be reusable by multiple systems. The second is a bottom-up approach which proposes an integrated user model for a specific goal within a specific context [17].

The Friend of a Friend (FOAF) ontology[6] is a top level ontology that models generic information about a user, including their name, social graph, interests and location. However, current FOAF profiling is based on the *static* representation of, in some cases, highly changing data, such as the temporal location of a user, or their current position in the world (à la *foursquare*[7]). Since many of these highly changing user properties can be observed through sensors, different ontologies for considering sensor data in user modelling have emerged. The Service-Oriented Context-Aware Middleware (SOCAM) ontology [18] is an upper ontology which introduces concepts like `Activity`, `Location`, `ComputationalEntity` and `Time` under the umbrella concept of `ContextEntity`. Although it models sensors using the `Device` concept,

[6] FOAF ontology: http://xmlns.com/foaf/spec

[7] http://www.foursquare.com

it does not provide a link between a sensor and its owner, nor a relation between the sensor's observations and user properties.

The General User Model Ontology (GUMO) is a top-level ontology introduced in 2005 [19]. GUMO is based on the User Mark-up Language (UserML) [20] and considers dimensions including personality, demographics, emotional and physiological state. The use of sensors in GUMO is considered particularly for users' physiological state; Heckman et al. [19] suggest the use of wearable bio-sensors to register users' body conditions such as pupil dilation and blood pressure. Although they consider the use of UbisWorld[8] for integrating users in ubiquitous environments, there is no clear definition of the way in which a sensor's relationship with a user's properties could be addressed.

The Ontonym ontologies[9] [21] are a collection of seven upper ontologies for pervasive computing. including the *Sensor*, *Device* and *Person* ontologies. They are designed to allow the definition of ownership between a `Sensor` and a `Person` through the `Device` class's `owns` and `ownedBy` properties. However, Ontonym requires the definition of a new ontology to map each sensor observation to user properties. For example, to add a relationship between a user's mobile device GPS's location observations and the user's location, the *Location* ontology is defined to declare the `LocatableEntity` and `LocatableFeature` classes (and associated properties). Ontonym is, to the best of our knowledge the only user modelling ontology available online.

Work done in sensor data integration into ontology-based user modelling following a bottom-up approach includes the Mobile Ontology-based Reasoning and Feedback System (MORF) [22], which defines a set of domain-specific ontologies which include classes such as `Patient`, `Doctor` and `HeartRateSensorData`. Their model allows monitoring and transmitting a patient's data through a mobile device. However, restrictions due to domain-specific design prevent MORF and other such bottom-up ontologies from being extensible to generic user modelling.

Relevant components of standard ontologies are discussed in the requirements identified in section 4 and revisited in 5.4 where we assess the extent to which these are met.

4 Requirements

The scenario presented in section 2.3 highlights not only the relevance of the identification of sensors and their observations as meaningful web resources, but also the importance of addressing the generated data streams as users' feature properties. In this section, we identify requirements for associating sensor data to user modelling.

Identification and Addressability: To uniquely identify and dereference sensor resources. In our scenario, Alice should be able to identify Bob's sensors, as well as the potential relations among these sensors. For example, by exposing Bob's physical activity and sugar levels, through the definition of his pedometer, as well as his glucose sensor as web resources, health care services could react in a contingency situation, in which external entities such as nearby emergency medical services could respond according to Bob's physical location (see section 6.2).

[8] UbisWorld can be tested at: `http://www.ubisworld.org`

[9] Ontonym ontologies: `http://ontonym.org`

Sensor Ownership and Provenance: To establish the sources of information, including entities and processes involved in the generation of measurements from observed stimuli. Provenance in sensor data is crucial for assessing trust judgements on information. An `Entity`, in particular a `Person`, should be able to address a sensor as its own – Bob, for instance, should be able to associate sensors with himself. Given a sensor data stream, it should be possible to access the sensor publishing the given stream and identify the sensor's owner. In the scenario, Alice and the GP should be able to identify the streams they are consuming as Bob's.

Association of Sensor Data and Profile Information: To map explicitly, a user's property characterised by a stimulus with the sensor that observes this stimulus. In the scenario, Alice must be able to associate Bob's (continuously changing) location with, e.g., Bob's `current_location` property, observed by his GPS.

Privacy in Data Streams: To consider how identity information should be exposed and to whom: (1) The consumer of a data stream should be guaranteed that no other service has impersonated the sender; (2) The owner of a data stream should be able to establish authentication methods so only authorised consumers have access to it. E.g., besides Bob, Alice should be authorised to access Bob's health information, as well as the closest emergency doctor who treats Bob at the scene (the latter will have access to a filtered view).

Sensor Data Expiration: To enable a data stream to declare an estimation of the period of time in which its data should be considered valuable. In our scenario, Alice must be able to tell if the received information is still valid, e.g., Alice must know the latest (valid) position of Bob and the time beyond which it is no longer valid.

Interaction with Smart Entities: To allow the representation of collective stimuli in which different entities, including the user, are involved. With collective stimuli we refer to the aggregation of common detectable changes in observable properties. E.g., Bob's location-based proximity social graph is a property derived from the collective stimulus of being located in the vicinity of Bob, within a radius of 5km.

Integrate Physical and Virtual Presence Stimuli: To identify and incorporate virtual and physical stimuli as part of a user's presence. This integration would bridge the user's physical and online personae. In the scenario, the NutrApp would make use of Bob's online social network to obtain the references of those entities to be monitored for physical presence proximity.

5 The Sensing Presence Ontology

In this section we introduce concepts related to users' presence and present the Sensing Presence (`PreSense`) Ontology[10]. It defines key concepts and properties required to describe users' features in terms of virtual sensor observables. `PreSense` models users as entities whose presence is the aggregation of online and physical properties. It represents sensors' observations for deriving presence properties and particular features of interest, following the Stimulus-Sensor-Observation (SSO) ontology design pattern [23]. Fig. 2 illustrates the structure of the `PreSense` ontology, focusing on the relationships between its core components.

[10] `PreSense` ontology available at: `http://purl.org/net/preSense/ns`

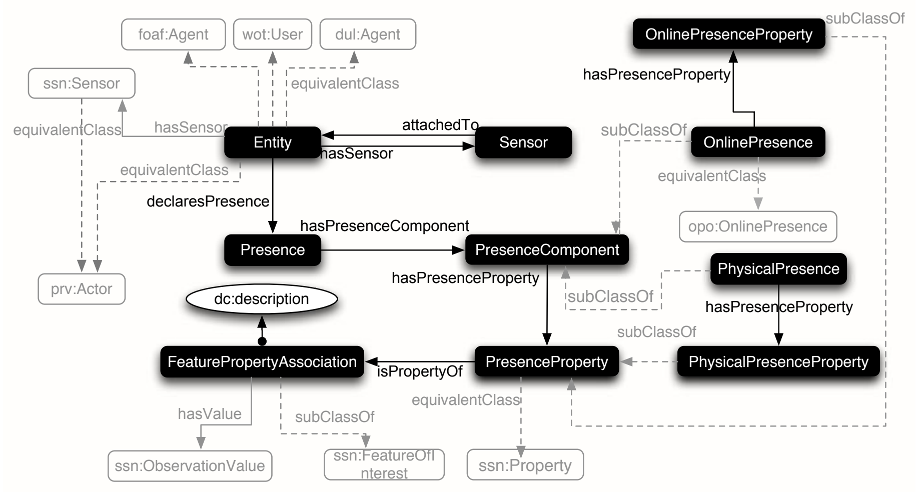


Fig. 2. Sensing Presence Ontology (`PreSense`) Overview

5.1 User Modelling Based on Personal Sensors

Based on the definition of virtual sensors (in section 2.2) and the SSO design pattern [23], we introduce the concept of "*personal sensors*" to refer to both physical sensor devices and compositions of computations or procedures that measure a user's properties. The information embedded in the data produced by personal sensors includes the users' *online* and *physical presence contexts*. By *online presence context*, we refer to the information provided by the aggregation of personal data streams (e.g. microblog posts, emails, text messages) generated by a user within a window of time. We consider the *physical presence context* as the abstraction of physical features, measured by sensor devices, regarding a user's state of existence or being present in a place or a thing (e.g. the user's location, body temperature). Both online and physical presence interweave dynamically with a user's surrounding environmental context, which can include other entities like people, places and things (e.g. members of the user's social graph who are close by, or local points of interest – POIs).

5.2 Imported Ontologies

Specifications on how to exchange sensor data and their observations have been defined by the Open Geospatial Consortium (OGC). In particular the OGC's Sensor Web Enablement[11] (SWE) suite is a broad standardisation initiative which comprises models such as the Sensor Model Language[12] (SensorML) and the Observation & Measurement[13] (O&M) standards, and services such as the Sensor Observation Service (SOS)

[11] `http://www.opengeospatial.org/projects/groups/sensorweb`

[12] `http://www.opengeospatial.org/standards/sensorml`

[13] `http://www.opengeospatial.org/standards/om`

[23]. However, sensor data sharing and discovery expose different challenges involving semantic heterogeneity and integration. The Semantic Sensor Web (SSW) [13] approaches these challenges by providing an ontological platform that defines a machine-readable specification of the conceptualisations that underlie this sensor data.

There are over twelve sensor ontologies [11] for declaring a specification of sensing devices; some include sensors' domain definitions and their relation to observations and measurements. The need for a domain independent and end-to-end model for sensing applications led to the creation of the W3C's Semantic Sensor Network Incubator Group[14] (SSN-XG), who developed the Sensor and Sensor Network (SSN) ontology[15] [24,25]. Taking into account available standards such as the OGC's SWE, the SSN ontology merges sensor-, observation- and system-focused views. The ontology describes sensors following the SSO ontology design pattern [23] and considers spatial provenance properties through the SSN's Deployment module.

Following ongoing research and standardisation efforts, we use the SSN ontology to represent sensors in PreSense. Further, we use the Provenance Vocabulary[16] (PRV) [26] to extend provenance-related metadata regarding both sensors and their owners through prv:Actor. For modelling an Entity asserting the ownership of a sensor, we use foaf:Agent. According to the FOAF specification, a foaf:Agent can refer to a person, a group, software or a physical artifact. The Web of Trust[17] (WOT) ontology is used to ensure that the ownership of a sensor cannot be falsified by a third party, thus providing a solid base for valid sensor attachment. PreSense models a user to be equivalent to a wot:User. This equivalence allows a user to assert a digital signature to a web resource, which ensures that: (1) The provenance of the resource cannot be falsified easily; (2) The resource cannot be modified without revoking the provenance of the information.

From the Online Presence Ontology[18] (OPO), we reuse opo:OnlinePresence to model users' online presence properties. Finally, from the Dolce Ultralight Ontology (DUL)[19] we reuse dul:Agent to align existing properties of SSN with a preSense:Entity (abbreviated prefix ps: used hereafter), and dul:Situation, in defining the contextual setting of an entity's ps:Presence.

5.3 Core Components

Table 1 summarises the requirements fulfilled by each of the core components of the PreSense ontology, which we discuss next:

Entity. An entity is modelled to be equivalent to foaf:Agent, wot:User, dul:-Agent and prv:Actor. The function of the Entity class is twofold: (1) to describe the identity of an individual (not only persons but entities in general) to whom the sensor data should be attached; and (2) to avoid provenance falsification

[14] SSN Incubator Group: http://www.w3.org/2005/Incubator/ssn/wiki

[15] SSN Ontology: http://purl.oclc.org/NET/ssnx/ssn

[16] Provenance Vocabulary: http://purl.org/net/provenance

[17] WOT Ontology: http://xmlns.com/wot/0.1

[18] Online Presence Ontology: http://online-presence.net/opo/spec

[19] Dolce Ultralight Ontology : http://www.loa-cnr.it/ontologies/DUL.owl

Table 1. Match of core PreSense ontology components to requirements

	Ident. & Address-ability of Sensors	Sensor's Ownership & Prove-nance	Sensor & User Profile Assoc.	Privacy	Data Expiration	Interaction with Smart Objects	Integration of Phys. & Virt. Pres. Stimuli
Entity	●	●	–	○	–	–	–
Sensor	–	–	●	–	●	●	–
Presence	–	–	–	–	–	●	●
PhysicalPresence	–	–	–	–	–	●	●
OnlinePresence	–	–	–	–	–	–	●
FeaturePropAssoc.	–	–	●	–	–	–	–

Legend: ● Yes. ○ Limited. – No.

through the use of digital signatures in wot:User. The Entity class considers the property hasSensor for attaching a sensor to an entity (its inverse property is attachedTo). Entity is skos:closeMatch with ssn:Platform, which is considered to be an Entity to which a System of sensors is attached. However, SSN considers a Platform to be a dul:PhysicalObject which is disjoint with dul:SocialObject.

Sensor. A sensor is defined by the ssn:Sensor class and refers to a physical object that detects, observes and measures a stimulus. The ps:attachedTo property is used to assert that a ps:Entity owns this sensor (its inverse property is ps:hasSensor). In order to extend provenance metadata of a sensor and its observations, we model the ssn:Sensor to be equivalent to a prv:Actor.

Presence. A Presence refers to the state or fact of existing or being manifest in a place or a thing. We consider that a Presence is an aggregation of an Entity's online and physical manifestations, that occur within a situation or setting. Following DUL, a situation is defined as a "*relational context*" created by an observer on the basis of a description frame.

Physical Presence. This is the abstraction of the aggregation of physical properties featuring a quality of an entity. These properties are derived by sensors observing physical stimuli. The ps:PhysicalPresence class manifests an entity to be in a state of existing or being present in a place or a thing. These physical presence properties can be broken down into different modules regarding different dimensions in which users' properties can be linked to sensor data.

Online Presence. This is equivalent to opo:OnlinePresence; it refers to the abstraction of the aggregation of online properties featuring a quality of an Entity, e.g., a user. These properties are derived by virtual sensors observing stimuli involving this Entity, e.g., the detection of a user's change of status on a social network site through the ps:OnlineStatusStream.

Feature Property Association. Following the SSO ontology design pattern we introduce this class to bridge a sensor's observed stimulus and the feature that this stimulus characterises in the user model. It is a subclass of ssn:FeatureOfInterest; which being an abstraction of real world phenomena, proxies a stimulus through a quality that can be observed by (an ssn:Property of) a sensor, and the

PreSense property describing this quality (i.e., ps:PresenceProperty). The ps:FeaturePropertyAssociation class establishes a relation with the ps:Presence through the property ps:hasPresenceProperty, and by declaring ps:PresenceProperty to be owl:sameAs ssn:Property.

5.4 Fulfilment of the Requirements

The PreSense ontology addresses all of the requirements identified in section 4. Table 2 summarises the differences between the PreSense Ontology and the existing upper ontologies introduced in section 3.

Table 2. The PreSense Ontology, compared to existing, standard models

	Ident. & Address-ability of Sensors	Sensor's Ownership	Sensor's Prove-nance	Sensor & User Profile Assoc.	Privacy	Data Expiration	Interaction with Smart Objects	Integration of Phys. & Virt. Pres. Stimuli
FOAF	–	–	–	–	–	○	–	○
SOCAM	○	–	○	–	–	○	●	○
GUMO	–	●	–	–	–	○	○	○
Ontonym	–	●	●	–	–	●	–	–
PreSense	●	●	●	●	○	●	●	●

Legend: ● Yes. ○ Limited. – No.

PreSense uses the SSN:Sensor ontology to model sensors and sensor data. Entity acts as the bridge through which sensor data and profile information can be associated. By reusing the FOAF, WOT and PRV ontologies, entities and sensor ownership can be uniquely identified. The use of WOT partially covers privacy issues. However questions still remain about the correct structure for introducing privacy settings within data streams; we aim to tackle this in future work. Sensor data expiration can be handled using ssn:observationSamplingTime. PreSense allows the representation of physical and online presence and their corresponding properties by enabling a bridge between a user's properties and the sensors observing these properties.

6 Applying PreSense

This section revisits the scenarios presented in section 2.3 and provides an overview on how to represent different information with the PreSense core ontology.

6.1 Extending PreSense Core Ontology with Modules

PreSense modules are extensions to the PreSense core vocabulary that provide additional information regarding a specific type of property. Currently PreSense has two modules, the *spatial properties* module and the *health properties* module. The spatial property module includes Location, which is a spatial quality of an entity; this property is linked to a sensor by the `ps:FeaturePropertyAssociation` whose value is the observation of, in this case, a GPS sensor. The health properties module considers the `PhysiologicalState` class and its subclasses (Fig. 3).

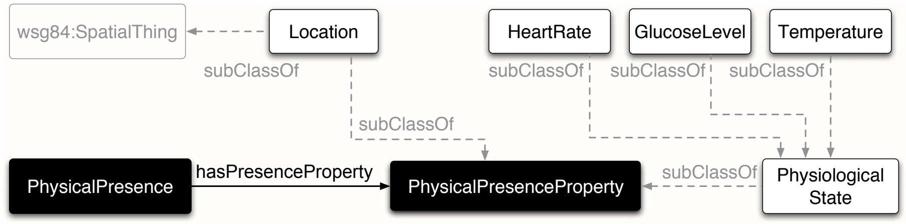


Fig. 3. `PreSense` modules for handling features related to `Location` and `Physiological State`

6.2 Scenario with PreSense

In this scenario Bob's levels of glucose can be monitored as part of his profile. This could be modelled with the `PreSense` ontology as:

```
 @prefix ps: <http://purl.org/net/preSense/ns\#>  .
 @prefix physioState: <http://purl.org/net/preSense/physioState/ns\#>  .
@prefix prvTypes: <http://purl.org/net/provenance/types#> .
 @prefix prv: <http://purl.org/net/provenance/ns>  .
 @prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn\#>  .
 <http://my.identity.org/Bob> a ps:Entity, a foaf:Person;
 ps:hasSensor <http://my.identity.org/Bob/sensors/glSen1/>.
 ps:declaresPresence _:p1.

 _:p1 a ps:Presence;
   ps:hasPresenceComponent _:phyPr.

 _:phyPr a ps:PhysicalPresence;
   ps:hasPresenceProperty _:prop1.

 _:prop1 a physioState:GlucoseLevel;
   ps:hasPresenceProperty _:glucoseLevel.
   ps:isPropertyOf _:bloodGlucose .

 <http://my.identity.org/Bob/sensors/glSen1/>
   a ssn:Sensor, prv:Actor, prvTypes:Sensor;
   prv:operatedBy <http://my.identity.org/Bob> .
   prv:observedBy <http://my.identity.org/Bob/sos/observations/glSen1/>.
 <http://my.identity.org/Bob/sos/observations/glSen1/> a ssn:Observation;
   ssn:observedProperty _:glucoseLevel.
 _:glucoseLevel   a ssn:Property, ps:PresenceProperty;
   ssn:isPropertyOf _:bloodGlucose.
 _:bloodGlucose   a ps:FeaturePropertyAssociation
```

In this example, Bob registers his glucose level measuring sensor (`glSen1`) and his physical presence (`phyPr`). His physical presence considers in this example the health properties module; in particular the glucose level property (`glucoseLevel`). This property corresponds to the property observed by his `glSen1` sensor. This sensor observes changes in his blood sugar levels, (`bloodGlucose`), which is the feature of interest. This association enables Alice to monitor Bob's sugar levels. Following the URI scheme for linked sensor data proposed by Janowicz et al. [27], Alice could refer to, e.g., `http://my.identity.org/Bob/sos/observations/glSen1/mgPerdL`, which is a reference to all observations gathered by `glSen1` corresponding to the feature of interest `bloodGlucose` for the observed property, milligrams per decilitre, `mgPerdL`. In a similar way the `PreSense` ontology could be applied for registering Bob's heart rate micro-sensor.

The scenario also considers the attachment of virtual sensors to the user's profile. Bob could allow other systems to consume his online status stream (e.g., tweet streams) as:

```
<http://my.identity.org/Bob> a ps:Entity, a foaf:Person;
ps:hasSensor <http://my.identity.org/Bob/sensors/stSen1/>.
ps:declaresPresence _:p1.

_:p1 a ps:Presence;
  ps:hasPresenceComponent _:onlPr.

_:onlPr a ps:OnlinePresence;
  ps:hasPresenceProperty _:prop2.

_:prop2 a ps:OnlineStatusStream;
  ps:hasPresenceProperty _:personalStatusStream.
  ps:isPropertyOf _:twitterStatusStream .


<http://my.identity.org/Bob/sensors/stSen1/>
  a ssn:Sensor, prv:Actor, prvTypes:Sensor;
  prv:operatedBy <http://my.identity.org/Bob> .
  prv:observedBy <http://my.identity.org/Bob/sos/observations/stSen1/>.
<http://my.identity.org/Bob/sos/observations/stSen1/> a ssn:Observation;
  ssn:observedProperty :personalStatusStream.
_:personalStatusStream   a ssn:Property, ps:PresenceProperty;
  ssn:isPropertyOf _:twitterStatusStream.
_:twitterStatusStream   a ps:FeaturePropertyAssociation
```

In this case, Bob declares his personal status stream as a property of his online presence `onlPr`. This property is a proxy for generated contingency tweets on behalf of Bob, and is observed by the virtual sensor `stSen1`. In this case all observations regarding generated tweets could be obtained through `http://my.identity.org/Bob/sos/observations/stSen1/status`. Data derived from his health monitoring devices could trigger an alert when Bob is facing a health contingency situation. This alert could be proxied through Bob's `stSen1` sensor; which could alert, e.g., a particular list of Bob's followers in his physical environment about his need for medical attention. They in turn could, on validating the information and its provenance, notify health services about the impending emergency.

7 Conclusions

The `PreSense` ontology is designed to extend people's digital identities through the information obtained by their attached personal sensors. It provides a first step toward the integration of user modelling and "smart environments". `PreSense` distinguishes between the notions of physical presence, e.g., location data obtained from digital sensors, and virtual presence, provided, for instance, by the aggregation of personal data streams, but affords equal status to both. Moreover, `PreSense` allows the assignment of meaning to sensors' observations in terms of user modelling features.

Future work includes the development of `PreSense` modules addressing interaction with smart entities and environments, by mapping a user's location to that of other nearby entities (*NearByPOI* and *NearByFriends* modules). We are also testing the application of the `PreSense` ontology in real world scenarios, starting with the exploration of new environments and ongoing events.

We are finalising plans for a two-part evaluation of `PreSense`. The first session will monitor `PhysicalPresence` in an indoor, *smart* environment, by tracking the interaction between *person* `Entities` to which *RFID tags* (`Sensors`) are attached, and fixed objects to which sensor readers will be attached (e.g., a *printer* – `POI`), and other sensor-enabled devices (e.g., *smart robotic dispensers* – `Sensor/Entity`), in a *research laboratory* – `POI`, over a fixed period of time. In this phase we aim to observe and measure *physical* interaction and `FeaturesOfInterest` in the smart environment as daily working activities take place.

A second evaluation will focus on end users' `OnlinePresence`, during the Tramlines Festival[20] in Sheffield at the end of July 2011. Bearing in mind privacy restrictions, we will record only the content of the information exchanged, with associated properties such as `Time` and `Location` via the Twitter public stream at selected events and POIs. The information collected will be modelled using `PreSense`, in order to build a database that maps event type to POIs, and measure the degree of online social interaction during different events. This will allow us to measure how `PreSense` may be used to recommend information to end users based on their profiles and that of their (physical and virtual) social circles both in real-time and over different periods of time.

Acknowledgements. A.E. Cano is funded by CONACyT, grant 175203. A.-S. Dadzie and V. Uren are funded by the European Commission (EC) 7th Framework Programme (FP) project SmartProducts (EC grant ICT-231204). The work reported in this paper was also supported by the EC 7th FP project WeKnowIt (EC grant ICT-215453). The authors thank Gregoire Burel and Matthew Rowe for their valuable comments.

References

1. Estrin, D., Culler, D., Pister, K., Sukhatme, G.: Connecting the physical world with pervasive networks. IEEE Pervasive Computing 1(1), 59–69 (2002)
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems (2009)
3. Goodchild, M.F.: Citizens as sensors: the world of volunteered geography. Geo. Journal 69, 211–221 (2007)
4. Hogg, T.: Inferring preference correlations from social networks. Electronic Commerce Research and Applications – Special Issue: Social Networks and Web 2.0 9(1), 29–37 (2010)

[20] Tramlines Festival 2011: `http://www.tramlines.org.uk`

5. Kamis, A., Stern, T., Ladik, D.: A flow-based model of web site intentions when users customize products in business-to-consumer electronic commerce. Information Systems Frontiers 12, 157–168 (2010)
6. Rowe, M.: The credibility of digital identity information on the social web: a user study. In: Proc. 4th ACM Workshop on Information Credibility on the Web, pp. 35–42 (2010)
7. Green, N., Harper, R.H.R., Murtagh, G., Cooper, G.: Configuring the mobile user: Sociological and industry views. Personal and Ubiquitous Computing 5, 146–156 (2001)
8. Hong, D., Chiu, D., Shen, V., Cheung, S., Kafeza, E.: Ubiquitous enterprise service adaptations based on contextual user behavior. Information Systems Frontiers 9, 343–358 (2007)
9. Küpper, A.: Location-Based Services: Fundamentals and Operation. Wiley (2005)
10. Yamada, T., Kaneko, M., Katou, K.: A Mobile Communication Simulation System for Urban Space with User Behavior Scenarios. In: Yang, L.T., Rana, O.F., Di Martino, B., Dongarra, J., et al. (eds.) HPCC 2005. LNCS, vol. 3726, pp. 979–990. Springer, Heidelberg (2005)
11. Compton, M., Henson, C., Neuhaus, H., Lefort, L., Sheth, A.: A survey of the semantic specification of sensors. In: SSN 2009: Proc. 2nd International Workshop on Semantic Sensor Networks at ISWC 2009, pp. 17–32 (2009)
12. Kabadayi, S., Pridgen, A., Julien, C.: Virtual sensors: Abstracting data from physical sensors. In: WoWMoM 2006: Proc. International Symposium on a World of Wireless, Mobile and Multimedia Networks, pp. 587–592 (2006)
13. Sheth, A.P., Henson, C.A., Sahoo, S.S.: Semantic sensor web. IEEE Internet Computing 12(4), 78–83 (2008)
14. Bonato, P.: Wearable sensors and systems. IEEE Engineering in Medicine and Biology Magazine 29(3), 25–36 (2010)
15. Konstantas, D.: An overview of wearable and implantable medical sensors. Yearbook of Medical Informatics 66(9) (2007)
16. Viviani, M., Bennani, N., Egyed-zsigmond, E., Liris, I.: A Survey on User Modeling in Multi-Application Environments. In: Third International Conference on Advances in Human-Oriented and Personalized Mechanisms, Technologies and Services (2010)
17. Baumgartner, N., Retschitzegger, W.: A survey of upper ontologies for situation awareness. In: Proc. 4th IASTED International Conference on Knowledge Sharing and Collaborative Engineering, pp. 1–9 (2006)
18. Gu, T., Wang, X.H., Pung, H.K., Zhang, D.Q.: An ontology-based context model in intelligent environments. In: Proc. Communication Networks and Distributed Systems Modeling and Simulation Conference, pp. 270–275 (2004)
19. Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., von Wilamowitz-Moellendorff, M.: GUMO – the General User Model Ontology. In: Ardissono, L., Brna, P., Mitrović, A. (eds.) UM 2005. LNCS (LNAI), vol. 3538, pp. 428–432. Springer, Heidelberg (2005)
20. Heckmann, D., Krueger, A.: A user modeling markup language (UserML) for ubiquitous computing. In: Proc. 9th International Conference on User Modeling, pp. 393–397 (2003)
21. Stevenson, G., Knox, S., Dobson, S., Nixon, P.: Ontonym: a collection of upper ontologies for developing pervasive systems. In: CIAO 2009: Proc. 1st Workshop on Context, Information and Ontologies, pp. 1–8 (2009)
22. Benlamri, R., Docksteader, L.: MORF: A mobile health-monitoring platform. IT Professional 12(3), 18–25 (2010)
23. Janowicz, K., Compton, M.: The Stimulus-Sensor-Observation ontology design pattern and its integration into the Semantic Sensor Network ontology. In: SSN 2010: Proc. 3rd International Workshop on Semantic Sensor Networks at ISWC (2010)

24. Compton, M., Neuhaus, H., Tran, K.N.: Reasoning about sensors and compositions. In: 2nd International Semantic Sensor Networks Workshop (2009)
25. Neuhaus, H., Compton, M.: The semantic sensor network ontology: A generic language to describe sensor assets. In: AGILE Workshop: Challenges in Geospatial Data Harmonisation (2009)
26. Hartig, O.: Provenance information in the web of data. In: LDOW 2009: Proc. Linked Data on the Web Workshop at WWW 2009 (2009)
27. Janowicz, K., Broering, A., Stasch, C., Everding, T.: Towards meaningful URIs for linked sensor data. In: FIS Workshop: Towards Digital Earth: Search, Discover and Share Geospatial Data (2010)

TUMS: Twitter-Based User Modeling Service

Ke Tao, Fabian Abel, Qi Gao, and Geert-Jan Houben

Web Information Systems, Delft University of Technology
{k.tao,f.abel,q.gao,g.j.p.m.houben}@tudelft.nl

Abstract. Twitter is today's most popular micro-blogging service on the Social Web. As people discuss various fresh topics, Twitter messages (tweets) can tell much about the current interests and concerns of a user. In this paper, we introduce TUMS, a Twitter-based User Modeling Service, that infers semantic user profiles from the messages people post on Twitter. It features topic detection and entity extraction for tweets and allows for further enrichment by linking tweets to news articles that describe the context of the tweets. TUMS is made publicly available as a Web application. It allows end-users to overview Twitter-based profiles in a structured way and allows them to see in which topics or entities a user was interested at a specific point in time. Furthermore, it provides Twitter-based user profiles in RDF format and allows applications to incorporate these profiles in order to adapt their functionality to the current interests of a user. TUMS is available via: http://wis.ewi.tudelft.nl/tums/

Keywords: user modeling, twitter, semantic enrichment, service.

1 Introduction

Applications that aim for personalization and would like to adapt their functionality to the current interests and demands of a user require information about their users [1]. User-adaptive systems suffer from cold-start and sparsity problems [2]. For example, when systems have to deal with new users or fresh content, those systems require user profile information that allows for estimating the interests of the users. In this paper, we present a Twitter-based user modeling service (TUMS) that exploits users' Twitter activities to infer semantically meaningful user profiles.

Since Twitter was launched in 2007, it became the most popular micro-blogging service, with 190 million users who post more than 65 million posts every day[1]. As people are not limited to a certain domain, they can discuss various topics of which most of the topics are related to news which could also be found in mainstream news articles [3]. There is also an active research community studying Twitter in terms of topics like social network analysis [3,4], community and user influence mining [5,6,7], recommendations of URLs [8] or

[1] http://techcrunch.com/2010/06/08/twitter-190-million-users/

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 269–283, 2012.

sentiment analysis [9,10]. Inferring user profiles from individual Twitter activities is a hard problem as tweets are limited to 140 characters which makes the deduction of semantics difficult. Rowe et al. [11] propose to exploit the context in which tweets have been published. In particular, they propose to map tweets to events and exploit semantic descriptions of these events to clarify the semantic meaning of tweets. In previous work, we proposed strategies to automatically map Twitter messages to related news articles which allows us to exploit the news articles to enrich the semantics of individual tweets [12]. This enrichment builds the basis for the user modeling strategies [13] that are made available via the TUMS service which we present in this paper.

The TUMS service allows end-users to inspect Twitter-based profiles and enables other applications to re-use these profiles. People can overview their personal Twitter activities or profiles of other users to explore the topics those users were concerned with in the past. Entity-based, topic-based and hashtag-based tag clouds allow to further visualize the profiles. Visualizing the profiles is important for getting an insight into the Twitter activities: for example, individual people can thus become aware of what can be inferred from their Twitter activities and perhaps then reconsider how they publish tweets on Twitter. TUMS is also of interest for other services on the Social Web as it enables them to consume Twitter-based profiles in RDF format. Profiles can be used for personalization and are particularly interesting for other applications that suffer from sparsity problems (e.g. services that cannot collect sufficient data about their users) and services that are interested in "realtime" or very fresh profile information.

In the following section we summarize related work before we introduce TUMS in detail in Section 3. Then we outline the architecture of our service, the user modeling strategies featured by TUMS and present the graphical user interface as well as the service API. Insights on applying TUMS within an application that aims for personalization is given in Section 4. Finally, we conclude and give an outlook on our plans for future work in Section 5.

2 Related Work

With the advent of Semantic Web technologies and appropriate vocabularies such as FOAF [14], SIOC [15] or the Weighted Interest vocabulary [16], re-use of user profiles is becoming easier nowadays. Research on generic user modeling services [17], mediating user models [18], identifying users across system boundaries [19] and cross-system user modeling and personalization [20,21] further supports the re-use of user profiles in different application contexts. In this paper, we introduce a service that generates user profiles by exploiting Twitter and allows for applying these profiles in other applications.

Research on Twitter often focuses on analyzing large fractions of the Twitter network to study information propagation patterns [3,22,23] or to identify influential users [6,7]. Dong et al. [24] exploit Twitter to detect and rank fresh URLs that have possibly not been indexed by Web search engines yet. Lately, Chen et al. conducted a study on recommending URLs posted in Twitter messages and compare strategies for selecting and ranking URLs by exploiting the social network of a user as well as the general popularity of the URLs in Twitter [8]. Yet, there exists little research on analyzing the semantics of individual tweets and exploiting Twitter as a source for modeling user interests. Rowe et al. [11] proposed to exploit contextual information to enrich the semantics of Twitter messages(tweets). In previous work, we followed this suggestion and linked tweets to related Web resources [12]. Based on this, we consider on the technical and semantical aspects of the services Given this semantic enrichment, we proposed user modeling strategies that allow for recommending news articles. In this paper, we make those enrichment and user modeling strategies available to the public and present TUMS, a Twitter-based User Modeling Service.

3 Architecture, Stategies, and Implementation

TUMS is a user modeling service that exploits the tweets of individual users to infer and provide user interest profiles. TUMS targets two types of consumers: (i) end-users who would like to overview and inspect their profiles (or profiles of friends) in a structured way and (ii) applications that would like to incorporate the RDF-based user profiles in order to adapt their functionality to the current interests of a user. In this section, we first present the architecture of TUMS; then we describe user modeling strategies in TUMS; finally, we present both graphical and machine-processable user profiles generated by TUMS.

3.1 Architecture

TUMS aims to to deliver both a human and a machine readable representation of the user characteristics that can be deduced from the posts a user published on Twitter. The only input required by TUMS is the Twitter username of the person that should be profiled. Figure 1 depicts the architecture of the TUMS service. TUMS is composed of three modules: (i) the *Crawler and backend*, (ii) the *User Modeling* module, (iii) the *Graphical User Interface and RDF Endpoint*.

Crawler and Back-end. To start using the TUMS functionality is easy as TUMS just requires a Twitter username as input. Given a username, TUMS initiates a pipeline for crawling, processing, storing and analyzing the Twitter data that is publicly available for the corresponding user. TUMS features a tweets and a news crawler: the tweets crawler aggregates Twitter posts of the given user and stores the tweets in the TUMS data repository; the news crawler continuously monitors traditional news media. At the moment, TUMS monitors 62 RSS feeds published by mainstream news publishers such as CNN, New York Times, and BBC and aggregates the full content of each news article by making use of Boilerpipe[2]. Once a new username

[2] `http://code.google.com/p/boilerpipe/`, a Java library that detects supplemental text (e.g. advertisements or menus) in a website and allows for the extraction of the main content.

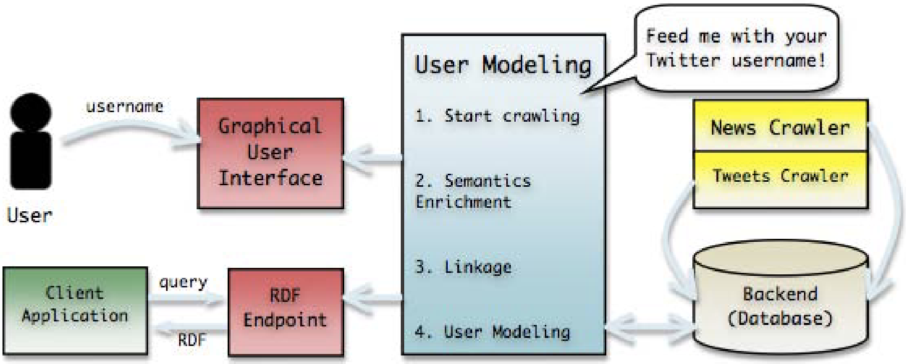


Fig. 1. The architecture of TUMS

is provided through either the graphical interface or the RDF endpoint, TUMS will continuously start monitoring the tweets of the corresponding user. Whenever a new tweet is observed, the tweet will get processed to enrich the semantics of the tweet (see below). Tweets, news articles and semantic metadata which is extracted from tweets and news, are stored in the data repository which builds the basis for user modeling.

User Modeling. Given the raw data crawled from Twitter, TUMS aims to infer user interest profiles that adhere to the Friend-Of-A-Friend (FOAF) [14] vocabulary and the Weighted Interest vocabulary [16]. Hence, given the raw text of tweets, TUMS will output profiles which specify how much a user is interested in a certain topic. The example output can be found in Section 3.4. In order to infer interests of a user from the tweets this user posted on Twitter, TUMS performs four main steps: (1) it notifies the tweet crawler to start collecting tweets, (2) it enriches the semantics of the user's tweets by categorizing the topic of a tweet and extracting hashtags and other entities (e.g. persons, locations) from the tweet, (3) if possible, it links the tweet to related news articles to further enrich the semantics of a tweet with semantics extracted from the news article and (4) it applies different user modeling strategies (see Section 3.2) to generate user profile information.

The first three steps will be executed whenever a new tweet is observed while the last step, the actual user modeling, is executed on query time, i.e. when a user (via the graphical user interface) or client application (via the Web service) requests profile information.

Graphical User Interface and RDF Endpoint. Given the data aggregated, processed and stored by the above modules, the Twitter-based user profiles can be retrieved by visiting the graphical user interface or by invoking the TUMS Web service API. The graphical interface enables end-users to overview their profiles by means of tag clouds (topic-based, hashtag-based and entity-based clouds) and diagrams (see Section 3.3). The REST-like Web service interface outputs user profiles in RDF using FOAF [14] and Weighted Interest vocabulary [16] as depicted in the above example.

Table 1. Design space of Twitter-based user modeling strategies

design dimension	**design alternatives**
profile type	(i) hashtag-based, (ii) topic-based, or (iii) entity-based
weighting schemes	(i) term frequency (TF) or (ii) $TF \times IDF$
enrichment	(i) tweet-only-based enrichment or (ii) linkage and exploitation of external news articles (propagating entities/topics)
temporal constraints	(i) specific time period(s), (ii) temporal patterns (*weekend*, *night*, etc.), or (iii) no constraints

3.2 User Modeling Strategies

The goal of TUMS and the user modeling strategies in particular is to generate user interest profiles that conform to the following model.

Definition 1 (User Profile). *The profile of a user $u \in U$ is a set of weighted concepts where with respect to the given user u for a concept $c \in C$ its weight $w(u, c)$ is computed by a certain function w.*

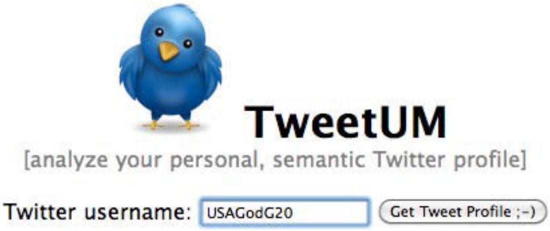
$$P(u) = \{(c, w(u, c))|c \in C\}, u \in U$$

Here, C and U denote the set of concepts and users respectively.

To generate such profiles, we developed a set of user modeling strategies [13] that vary in four design dimensions: (i) the type of profiles created by the strategies, (ii) the weighting scheme, (iii) the data sources exploited to further enrich the Twitter-based profiles, and (iv) temporal constraints that are considered when constructing the profiles (see Table 1). The generic model for profiles representing users is specified in Definition 1.

As listed in Table 1, TUMS allows for three *types* of profiles that differ with respect to the type of concepts $c \in C$ for which we specify an interest weight (see Definition 1): entity-based, topic-based, and hashtag-based profiles. For entity-based profiles, we differentiate the weights between the entities extracted from the tweets. The topic-based profiles are rather broad and abstract from the concrete content as we map the tweets to 19 static topics such as sports, politics, or music.

TUMS allows for different methods as weighting scheme $w(u, c)$. For example, using term frequency (TF), the weight of a concept is determined by the number of Twitter activities in which user u refers to concept c. In a hashtag-based profile, $w(u, \#\mathit{technology}) = 5$ means that u published five tweets that mentioned "#technology". Other weighting methods such as $TF \times IDF$ are possible as well and a detailed comparison of different weighting schemes is planed for future work. The resulting user profiles will be normalized so that the sum of all weights in a profile is equal to 1: $\sum_{c_i \in C} w(u, c_i) = 1$. With $\boldsymbol{p}(u)$ we refer to $P(u)$ in its vector space model representation, where the value of the i-th dimension refers to $w(u, c_i)$.



Twitter username: USAGodG20

Fig. 2. The start page of TUMS in the graphical user interface

Tweets posted by a user u may refer to external resources. A user can explicitly link to other Web resources in her tweet or she could discuss topics and events that are, for example, discussed in mainstream news articles as well. TUMS aims to also take these external resources into account when constructing entity-based and topic-based user profiles (see semantic enrichment in Table 1). In particular, profiles are enriched with entities and topics extracted from news articles that are linked with tweets. In previous work, we have done the study on selecting appropriate news articles for enriching tweets [12]. We revealed that for tweets which do not explicitly link to external Web resources we can find related news articles that report about the same event as the tweet with a high precision of more than 70%.

Temporal constraints are considered as the fourth dimension of the profiles (see Table 1). By specifying temporal constraints, client applications can, for example, retrieve the latest profile of a user or a profile that is only based on Twitter activities which a user performed we revealed.

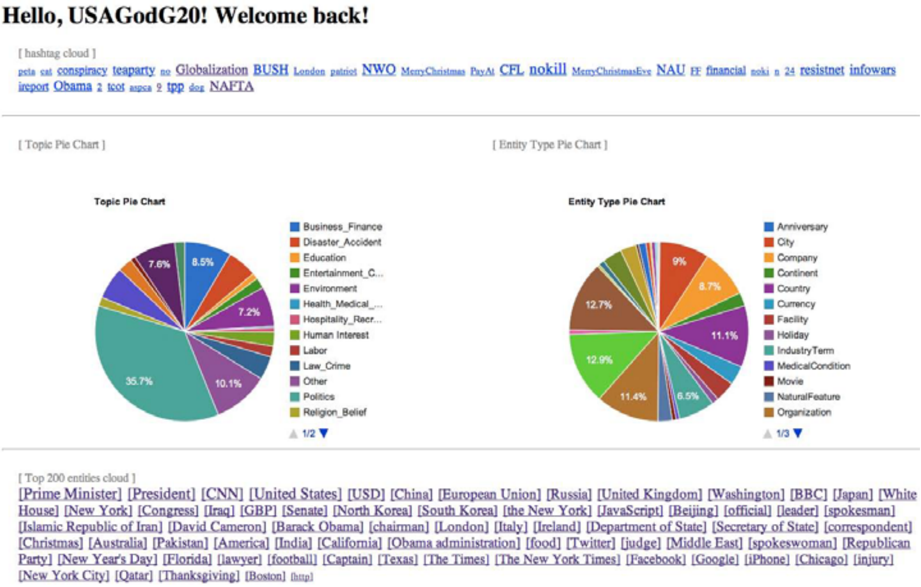
By selecting and combining the different design dimensions and alternatives, TUMS can exploit different user modeling strategies and thus generate different profiles, in the form of visualized charts and RDF triples.

3.3 Graphical User Interface of TUMS

End-users can easily access TUMS via its graphical user interface which requires only a Twitter username as input. Figure 2 shows the start screen that is displayed when users are accessing TUMS. In the given example, the profile of a user named “USAGodG20” should be returned. By clicking on the “Get Tweet Profile” button the user will be directed to the next page that allows the user to overview and inspect the profile of “USAGodG20” (see Figure 3).

TUMS can represent user profiles by means of different visualizations, including hashtag-based clouds, pie charts visualizing topic-based profiles and entity types referenced by a user, and entity-based clouds (see Figure 3). In the following, we examine the implemented visualization features of TUMS and describe them by example results that we obtain for requesting the profile of the user “USAGodG20”.

Hashtag-Based Clouds. The hashtag-based clouds represent the hashtag-based profile of a user and thus visualize the hashtags which the corresponding user



Hello, USAGodG20! Welcome back!

[hashtag cloud]

peta cat conspiracy teaparty no Globalization BUSH London patriot NWO MerryChristmas PayAt CFL nokill MerryChristmasEve NAU FF financial noki n 24 resistnet infowars ireport Obama 2 tcot aspca 9 tpp dog NAFTA

[Top 200 entities cloud]

[Prime Minister] [President] [CNN] [United States] [USD] [China] [European Union] [Russia] [United Kingdom] [Washington] [BBC] [Japan] [White House] [New York] [Congress] [Iraq] [GBP] [Senate] [North Korea] [South Korea] [the New York] [JavaScript] [Beijing] [official] [leader] [spokesman] [Islamic Republic of Iran] [David Cameron] [Barack Obama] [chairman] [London] [Italy] [Ireland] [Department of State] [Secretary of State] [correspondent] [Christmas] [Australia] [Pakistan] [America] [India] [California] [Obama administration] [food] [Twitter] [judge] [Middle East] [spokeswoman] [Republican Party] [New Year's Day] [Florida] [lawyer] [football] [Captain] [Texas] [The Times] [The New York Times] [Facebook] [Google] [iPhone] [Chicago] [injury] [New York City] [Qatar] [Thanksgiving] [Boston] [http]

Fig. 3. Overview of user profile page of TUMS

mentioned in her Twitter messages. Different hashtags will be displayed in different font size depending on their weight within the hashtag-based profile (cf. $w(u, c)$ in Definition 1). The higher the weight $w(u, c)$ within the profile $P(u)$, the bigger the font size.

For example, at the top of Figure 3 we see the hashtag cloud of "USAGodG20". The visualization of the hashtag-based profile tells us that "USAGodG20" is particularly interested in the news about globalization, NAFTA, and political leaders in the United States, e.g. Bush, Obama.

Topic-Based Piecharts. One of the reasons for the popularity of Twitter lies in the idea that anyone can contribute and discuss her opinion about any topic. TUMS generates topic-based profiles and the TUMS user interface provides a pie chart that displays the relative importance of a topic within a user profile. Topic-based profiles feature 19 different dimensions that correspond to broad topics like politics or education. Given the pie chart visualization of a topic-based profiles, we can see about which topics a user has published most.

The example topic-based piechart of USAGodG20 shows that 35.7% of her tweets are concerned with politics. This observation is consistent with the results we got from the hashtag-based cloud.

Entity Type-Based Piecharts. During the user modeling process, entities are extracted from tweets and external sources for enriching the semantic of tweets. TUMS extracts 39 types of the entities. The pie chart of entity types shows the importance of the entity types for a given user and can be inferred from the entity-based profiles.

The example pie chart generated based on the entity-based profile of USAGodG20 shows that USAGodG20 is often referring to persons (12.9%), organizations (11.4%), countries (11.1%) and cities (9%). Natural features (e.g. beach, ocean) or movies are less frequently mentioned.

Entity-Based Clouds. With entity-based clouds, TUMS visualizes the entity-based profile of a user and thus lists the top entities mentioned by a user. It refines the entity type based pie chart overview and shows particular persons, locations, etc. in which a user is interested. As for the hashtag-based cloud, entities are represented in different font size according to their weight $w(u, c)$ in the entity-based profile (cf. Definition 1).

At the bottom of Figure 3 we see the most important entities within the entity-based profile of USAGodG20. Entities with larger font size in the cloud, such as “Prime Minister”, “President”, “United States”, “China”, and “European Union” are words that are quite common within the political domain which confirms again the findings from the above profile visualizations.

3.4 RDF Endpoint of TUMS

User profiles are furthermore made available via a Web service in RDF format. Via HTTP content negotiation[3] or by specifying the format in the HTTP query string (*format=rdf*), client applications are enabled to request the different types of profiles generated by TUMS in RDF. The URL pattern that represents a query for profile information is defined as follows:

```
.../profile/[username]/[profile type]/[weighting]/[enrichment]/?[temporal
constraints]
```

Hence, in addition to the *username*, which refers to the Twitter username of the person whose profile should be returned, there are four types of parameters which can be customized in the pattern. These four parameters correspond to the four dimensions in Table 1. The format and possible values of these parameters are explained below.

`profile type`. The profile type parameter enables applications to specify which type of profile they would like to retrieve: hashtag-based, topic-based or entity-based. Given a user profile $P(u) = \{(c_1, w(u, c_1)), (c_2, w(u, c_2)), ...\}$ where $c_i \in C$, the profile type thus specifies the type of the concepts c_i. While topics and entities are represented via a URI by nature and therefore explicitly describe the semantic meaning of a topic and entity respectively, we try to clarify the semantic meaning of hashtags by referring to *#tagdef*[4]. Given the three types of profiles, the corresponding parameter values that can be used in the HTTP request are the following: *hashtag*, *topic*, and *entity*. For example, the hashtag-based profile can be retrieved as follows:

```
.../profile/USAGodG20/hashtag/
```

The RDF-formatted response to this HTTP request might be the following:

[3] http://tools.ietf.org/html/rfc2616#section-12

[4] http://tagdef.com

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix wi: <http://purl.org/ontology/wi/core#> .
@prefix wo: <http://purl.org/ontology/wo/core#> .
@prefix tums: <http://wis.ewi.tudelft.nl/tums/rdf/> .
@prefix dbpedia: <http://dbpedia.org/resource/> .

<http://twitter.com/USAGodG20>
  a foaf:Person;
  foaf:name "John of God";
  wi:preference [
     a wi:WeightedInterest ;
     wi:topic dbpedia:Conspiracy_(political) ;
     wo:weight [
        a wo:Weight ;
        wo:weight_value 0.0757 ;
        wo:scale tums:Scale
        ]
     ] ;
  wi:preference [
     a wi:WeightedInterest ;
     wi:topic dbpedia:Globalization ;
     wo:weight [
        a wo:Weight ;
        wo:weight_value 0.1576 ;
        wo:scale tums:Scale
        ]
     ] .
```

Hence, the hashtag-based profile of USAGodG20, which is in the graphical user interface represented by means of a tag cloud (see above), is represented using FOAF [14] and the Weighted Interest vocabulary [16]. For each concept(*wi:topic*), which is represented by a DBpedia URI, in the profile, the weight is clearly specified using *wo:weight_value*. TUMS normalizes the weights in a profile vector so that the sum of all weights in the profile is equal to 1. Correspondingly, the scale (*tums:Scale*) on which the weights are specified ranges from 0 to 1 (*wo:min_weight = 0*, *wo:max_weight = 1*, cf. Weighting Ontology[5]).

`weighting schemes.` TUMS supports different weighting schemes for computing the weights in a user profile (cf. $w(u, c)$ in Definition 1). At the moment, TUMS supports (i) term frequency (TF) and (ii) TF multiplied by inverse document frequency ($TF \times IDF$). The corresponding parameters are *tf* and *tfidf* so that a request may have the following format:

```
.../profile/USAGodG20/hashtag/tf/
```

`enrichment.` Currently, the possible options for this parameter are: (i) *tweet* or (ii) *tweetnews*. By default, TUMS extracts semantics from *tweet*. However, developers can specify whether external sources should also be exploited to further enrich the semantics of tweets when constructing the profile (*tweetnews*).

`temporal contraints.` Temporal constraints of profiles can be specified in the URL-based query as well. For specifying a certain time period, developers can use `start` and `end` to specify the start and end of the period that should be considered when generating the profiles. Moreover, it is possible to specify pattern-like temporal constraints like weekend or night. They can be

[5] http://purl.org/ontology/wo/core

specified by adding further parameters to the query string. At the moment, TUMS allows for a `wk` parameter which can be *weekend* (Twitter activities on Saturdays or Sundays) or *weekday* (Monday till Friday) and a `dk` parameter which can be *night* (from 6pm till 6am) or *day* (from 6am till 6pm). An example query combing these two features look like follows:

```
.../hashtag/?start=2010-12-01&end=2011-02-28&wk=weekend&dn=night
```

4 Application of TUMS

TUMS can be adopted by developers that aim at personalized applications. As Twitter allows for capturing daily activities, TUMS is of particular benefit for applications such as news recommendation systems that require information about the current concerns of a user. Moreover, it is beneficial for systems that suffer from sparsity problems such as the cold-start problem.

4.1 Developing Personalized Applications with TUMS

The cold-start problem is one of the issues that every personalized application developers have to solve. If a new user registers to the system then there is usually no or little profile information available. Therefore, it becomes difficult to provide personalization right after a new user registers to the system. Like stumbleupon[6], systems could decide to ask end-users to fill a long list of topic interests for starting knowledge. Contrary to this, TUMS provides a solution that enables systems to automatically obtain user interest profiles without requiring any further user interaction. Various personalized applications can be developed based on the TUMS service. For example, a recommender can easily know about a user with the generated profile in order to provide fresh contents like news articles, video clips, pictures, etc.

Liu et al. [25] analyzed a content-based recommender for Google News. They showed that interests in news topics such as technologies, politics, etc. change over time. Hence, information about the latest interests of a user is essential for personalized systems such as news recommender. As TUMS keeps collecting users' tweets in real-time, the profile can be generated based on the latest information.

End-users might ask themselves for the reason of such recommended items. Since the visualization of the profiles is already provided by TUMS, it can be applied to explain why recommendations have been generated. Such explanations may foster acceptance of recommendations. For example, for some e-shopping websites like Amazon[7] it is common to give an explanation why a product was recommended. Using TUMS, recommender systems can link to the (particular fragment of the) user profile that they used in order to compute the recommendation.

[6] `http://www.stumbleupon.com/`

[7] `http://www.amazon.com`

4.2 Personalized News Article Recommendations with TUMS

Let us consider the following scenario: Bob is a PhD student majoring in Semantic Web and interested in cycling during the weekend. He is a passioned Twitter user and a news junkie. First, let's look at what Bob has posted on Twitter. Here are two of his tweets:

- Reading interesting papers about a service aiming at helping personalized application developers to do user modeling based on twitter.
- Cycling day http://bit.ly/m71kUb

It would be great if a news recommending application can know Bob's interests from these tweets. However, without further processing these tweets are still in plain text. With TUMS, personalized application can benefits from these tweets, as the information in plain text can be enriched with more concepts not only from the tweets themselves, but also the external resources which can be refered from the URLs in the tweets. The tweets enriched by TUMS look as follows:

```
@prefix sioc: <http://rdfs.org/sioc/spec/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix dbpedia: <http://dbpedia.org/resource/> .

<http://twitter.com/#!/Bob/statuses/40435486531137536> a <sioc:Post> ;
     dcterms:created "2011-02-23T15:39:06+00:00" ;
     sioc:content "Reading interesting papers about a service aiming at
            helping personalized application developers to do user modeling
            based on twitter."@en ;
     sioc:topic dbpedia:Personalization;
     sioc:topic dbpedia:Twitter;
     sioc:topic dbpedia:Service-oriented_architecture;
     sioc:has_creator <http://twitter.com/#!/Bob/>;

<http://twitter.com/#!/Bob/statuses/42513338076381184> a <sioc:Post> ;
     dcterms:created "2011-02-27T19:46:06+00:00" ;
     sioc:content "Cycling day http://bit.ly/m71kUb"@en ;
     sioc:topic dbpedia:Cycling;
     sioc:topic dbpedia:Segregated_cycle_facilities;
     sioc:topic dbpedia:Mountain_bike;
     sioc:has_creator <http://twitter.com/#!/Bob/> ;
     sioc:links_to <http://bit.ly/m71kUb> .
```

Base on these enriched tweets of Bob, the semantically meaningful profiles can be generated by TUMS:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix wi: <http://purl.org/ontology/wi/core#> .
@prefix wo: <http://purl.org/ontology/wo/core#> .
@prefix tums: <http://wis.ewi.tudelft.nl/tums/rdf/> .
@prefix dbpedia: <http://dbpedia.org/resource/> .

<http://twitter.com/#!/Bob>
  a foaf:Person;
  foaf:name "Bob Green";
  wi:preference [
     a wi:WeightedInterest ;
     wi:topic dbpedia:Personalization ;
     wo:weight [
        a wo:Weight ;
        wo:weight_value 0.4407 ;
```

```
         wo:scale tums:Scale
         ]
      ] ;
   wi:preference [
      a wi:WeightedInterest ;
      wi:topic dbpedia:Cycling ;
      wo:weight [
         a wo:Weight ;
         wo:weight_value 0.1413 ;
         wo:scale tums:Scale
         ]
      ] .
```

Since Bob was doing research about personlization on weekdays, and cycled during the weekend, the profile above includes the entities of *Personalization* and *Cycling*. In the case that it is Sunday evening, the Women's World Cup is dominating the news, the Wimbledon final and the Tour de France are also ongoing, a news recommender can consider to deliver some personalized news articles. Then TUMS can help as the weekend profile can be retrieved. And the profile is understandable for the application because it is described in RDF:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix wi: <http://purl.org/ontology/wi/core#> .
@prefix wo: <http://purl.org/ontology/wo/core#> .
@prefix tums: <http://wis.ewi.tudelft.nl/tums/rdf/> .
@prefix dbpedia: <http://dbpedia.org/resource/> .

<http://twitter.com/#!/Bob>
   a foaf:Person;
   foaf:name "Bob Green";
   wi:preference [
      a wi:WeightedInterest ;
      wi:topic dbpedia:Cycling ;
      wo:weight [
         a wo:Weight ;
         wo:weight_value 0.1729 ;
         wo:scale tums:Scale
         ]
      ] ;
   wi:preference [
      a wi:WeightedInterest ;
      wi:topic dbpedia:Segregated_cycle_facilities ;
      wo:weight [
         a wo:Weight ;
         wo:weight_value 0.0823 ;
         wo:scale tums:Scale
         ]
      ] .
```

The concept *Cycling* in the weekend profile, is the word in the tweet that Bob posted on Sunday; while the concept of *Segregated cycle facilities* (a.k.a. bike lanes) comes from analyzing the external resources. TUMS analyzed the content of the blog post with the URL of "http://bit.ly/m71kUb" in Bob's tweet, and found that concept. Based on the weekend profile, recommedations can be made by computing and ranking the similarity between Bob's profile and the concepts in the news articles. Then, when Bob was going to use the news recommender application in a weekend, his weekend profile will be adopted to make the recommendation. As a result, a news article with the title of "Cycling-Tour rivals profit from Contador's misfortune", which was about the Tour de France, can be selected for him to read. In the meanwhile, Bob can also figure out why this news article is recommended to him through the indication of the concepts matched concepts in both his tweets or external resources mentioned in them, and the content of the news.

5 Conclusion and Future Work

In this paper, we present TUMS, a Twitter-based User Modeling Service, that generates semantic user profiles by exploiting tweets. Based on functionality for enriching the semantics of tweets, TUMS features a variety of user modeling strategies that produce entity-based, topic-based or hashtag-based profiles. TUMS makes these profiles available to other applications via an RDF endpoint and allows end-users to explore profiles visually. Given a Twitter username, TUMS aggregates tweets from Twitter and starts monitoring the Twitter user. All tweets published by the user will be processed by the semantic enrichment module of TUMS which, for example, extracts entities from tweets and links tweets to related external Web resources to further enrich the semantics of a tweet. Given the semantically enriched tweets, TUMS can be queried to return semantic profiles for specific periods in time.

TUMS is available online and ready for being used by platforms that aim for personalization based on users' Twitter activities. Evaluations regarding the characteristics and quality of profiles that are generated by TUMS are reported in [13] and show, for example, that entity-based user profiles generated by TUMS allow for high precision when recommending news articles. Our ambition for future work is to further investigate for what other types of personalization tasks what type of TUMS profiles are appropriate so that we can further advice developers who are using TUMS how they can customize TUMS user modeling strategies to optimize their personalization quality.

Acknowledgements. This work is partially sponsored by the EU FP7 projects ImREAL (http://imreal-project.eu) and GRAPPLE (http://grapple-project.org).

References

1. Jameson, A.: Adaptive interfaces and agents. The HCI handbook: fundamentals, evolving technologies and emerging applications, pp. 305–330 (2003)
2. Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.): Adaptive Web 2007. LNCS, vol. 4321. Springer, Heidelberg (2007)
3. Kwak, H., Lee, C., Park, H., Moon, S.: What is twitter, a social network or a news media? In: Proceedings of the 19th International Conference on World Wide Web (WWW 2010), pp. 591–600. ACM, New York (2010)
4. Krishnamurthy, B., Gill, P., Arlitt, M.: A few chirps about twitter. In: Proceedings of the first workshop on Online social networks. In: WOSP 2008, pp. 19–24. ACM, New York (2008)

5. Java, A., Song, X., Finin, T., Tseng, B.: Why we twitter: understanding microblogging usage and communities. In: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis. WebKDD/SNA-KDD 2007, pp. 56–65. ACM, New York (2007)
6. Weng, J., Lim, E.P., Jiang, J., He, Q.: Twitterrank: finding topic-sensitive influential twitterers. In: Davison, B.D., Suel, T., Craswell, N., Liu, B. (eds.) Proceedings of the Third International Conference on Web Search and Web Data Mining (WSDM 2010), pp. 261–270. ACM, New York (2010)
7. Cha, M., Haddadi, H., Benevenuto, F., Gummadi, P.K.: Measuring User Influence in Twitter: The Million Follower Fallacy. In: Cohen, W.W., Gosling, S. (eds.) Proceedings of the Fourth International Conference on Weblogs and Social Media (ICWSM 2010). The AAAI Press, Washington, DC, USA (2010)
8. Chen, J., Nairn, R., Nelson, L., Bernstein, M., Chi, E.: Short and tweet: experiments on recommending content from information streams. In: Proceedings of the 28th International Conference on Human Factors in Computing Systems (CHI 2010), pp. 1185–1194. ACM, New York (2010)
9. Gaffney, D.: #iranElection: quantifying online activism. In: Proceedings of the WebSci10: Extending the Frontiers of Society On-Line (2010)
10. Diakopoulos, N.A., Shamma, D.: Characterizing debate performance via aggregated twitter sentiment. In: Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI 2010, pp. 1195–1198. ACM, New York (2010)
11. Stankovic, M., Rowe, M., Laublet, P.: Mapping Tweets to Conference Talks: A Goldmine for Semantics. In: Passant, A., Breslin, J., Fernandez, S., Bojars, U. (eds.) Workshop on Social Data on the Web (SDoW 2010), co-located with ISWC 2010, Shanghai, China, vol. 664. CEUR-WS.org (2010)
12. Abel, F., Gao, Q., Houben, G.J., Tao, K.: Semantic Enrichment of Twitter Posts for User Profile Construction on the Social Web. In: García-Castro, R., et al. (eds.) ESWC 2011 Workshops. LNCS, vol. 7117, pp. 269–283. Springer, Heidelberg (2011)
13. Abel, F., Gao, Q., Houben, G.J., Tao, K.: Analyzing User Modeling on Twitter for Personalized News Recommendations. In: Konstan, J.A., Conejo, R., Marzo, J.L., Oliver, N. (eds.) UMAP 2011. LNCS, vol. 6787, pp. 1–12. Springer, Heidelberg (2011)
14. Brickley, D., Miller, L.: FOAF Vocabulary Specification 0.91. Namespace document, FOAF Project (November 2007), http://xmlns.com/foaf/0.1/
15. Bojars, U., Breslin, J.G.: SIOC Core Ontology Specification. Namespace document, DERI, NUI Galway (January 2009), http://rdfs.org/sioc/spec/
16. Brickley, D., Miller, L., Inkster, T., Zeng, Y., Wang, Y., Damljanovic, D., Huang, Z., Kinsella, S., Breslin, J., Ferris, B.: The Weighted Interests Vocabulary 0.5. Namespace document, Sourceforge (September 2010), http://purl.org/ontology/wi/core#
17. Kobsa, A.: Generic user modeling systems. User Modeling and User-Adapted Interaction 11(1-2), 49–63 (2001)
18. Berkovsky, S., Kuflik, T., Ricci, F.: Mediation of user models for enhanced personalization in recommender systems. User Modeling and User-Adapted Interaction (UMUAI) 18(3), 245–286 (2008)
19. Carmagnola, F., Cena, F.: User identification for cross-system personalisation. Information Sciences: an International Journal 179(1-2), 16–32 (2009)
20. Abel, F., Henze, N., Herder, E., Krause, D.: Interweaving Public User Profiles on the Web. In: De Bra, P., Kobsa, A., Chin, D. (eds.) UMAP 2010. LNCS, vol. 6075, pp. 16–27. Springer, Heidelberg (2010)

21. Mehta, B.: Learning from What Others Know: Privacy Preserving Cross System Personalization. In: Conati, C., McCoy, K., Paliouras, G. (eds.) UM 2007. LNCS (LNAI), vol. 4511, pp. 57–66. Springer, Heidelberg (2007)
22. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, pp. 851–860. ACM, New York (2010)
23. Lerman, K., Ghosh, R.: Information contagion: an empirical study of spread of news on digg and twitter social networks. In: Proceedings of 4th International Conference on Weblogs and Social Media, ICWSM (May 2010)
24. Dong, A., Zhang, R., Kolari, P., Bai, J., Diaz, F., Chang, Y., Zheng, Z., Zha, H.: Time is of the essence: improving recency ranking using twitter data. In: WWW 2010: Proceedings of the 19th International Conference on World Wide Web, pp. 331–340. ACM, New York (2010)
25. Liu, J., Dolan, P., Pedersen, E.R.: Personalized news recommendation based on click behavior. In: Rich, C., Yang, Q., Cavazza, M., Zhou, M.X. (eds.) Proceeding of the 14th International Conference on Intelligent User Interfaces (IUI 2010), pp. 31–40. ACM, New York (2010)
26. Passant, A., Hastrup, T., Bojars, U., Breslin, J.: Microblogging: A Semantic Web and Distributed Approach. In: Bizer, C., Auer, S., Grimnes, G.A., Heath, T. (eds.) Proceedings of the the 4th Workshop Scripting For the Semantic Web (SFSW 2008) co-located with ESWC 2008, Tenerife, Spain, vol. 368. CEUR-WS.org (2008)

Author Index