SEMAPRO 2014

The Eighth International Conference on Advances in Semantic Processing

ISBN: 978-1-61208-355-1

August 24 - 28, 2014

Rome, Italy

SEMAPRO 2014 Editors

Alexey Cheptsov, High Performance Computing Center Stuttgart (HLRS), Germany

Constandinos Mavromoustakis, University of Nicosia, Cyprus

SEMAPRO 2014

Forward

The Eighth International Conference on Advances in Semantic Processing (SEMAPRO 2014), held on August 24 - 28, 2014 - Rome, Italy, considered the complexity of understanding and processing information. Semantic processing considers contextual dependencies and adds to the individually acquired knowledge emergent properties and understanding. Hardware and software support and platforms were developed for semantically enhanced information retrieval and interpretation. Searching for video, voice and speech [VVS] raises additional problems to specialized engines with respect to text search. Contextual searching and special patterns-based techniques are current solutions.

With the progress on ontology, web services, semantic social media, semantic web, deep web search /deep semantic web/, semantic deep web, semantic networking and semantic reasoning, SEMAPRO 2014 constituted the stage for the state-of-the-art on the most recent advances.

We take here the opportunity to warmly thank all the members of the SEMAPRO 2014 technical program committee as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and efforts to contribute to the SEMAPRO 2014. We truly believe that thanks to all these efforts, the final conference program consists of top quality contributions.

This event could also not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the SEMAPRO 2014 organizing committee for their help in handling the logistics and for their work that is making this professional meeting a success. We gratefully appreciate to the technical program committee co-chairs that contributed to identify the appropriate groups to submit contributions.

We hope the SEMAPRO 2014 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in semantic processing.

We hope Rome provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city.

SEMAPRO 2014 Chairs

SEMAPRO Advisory Chairs
Wladyslaw Homenda, Warsaw University of Technology, Poland
Bich-Lien Doan, SUPELEC, France
Alexey Cheptsov, High Performance Computing Center Stuttgart (HLRS), Germany
Shu-Ching Chen, Florida International University, USA
Sule Yildirim Yayilgan, Gjøvik University College, Norway
Jesper Zedlitz, Christian-Albrechts-Universität Kiel, Germany
Soon Ae Chun, City University of New York, USA
Fabio Grandi, University of Bologna, Italy
David A. Ostrowski, Ford Motor Company, USA
Andrea Perego, European Commission - Joint Research Centre, Ispra, Italy

SEMAPRO Industry/Research Liaison Chairs
Riccardo Albertoni, IMATI-CNR-Genova, Italy
Panos Alexopoulos, iSOCO S.A., Spain
Sofia Athenikos, IPsoft, USA
Isabel Azevedo, ISEP-IPP, Portugal
Sam Chapman, The Floow Limited, UK
Daniele Christen, Parsit Company, Italy
Frithjof Dau, SAP Research Dresden, Germany
Thierry Declerck, DFKI GmbH, Germany
Alessio Gugliotta, Innova SpA, Italy
Peter Haase, Fluid Operations, Germany
Shun Hattori, Muroran Institute of Technology, Japan
Xin He, Airinmar Ltd., UK
Tracy Holloway King, eBay Inc., USA
Lyndon J. B. Nixon, STI International, Austria
Zoltán Theisz, evopro Innovation LLC, Hungary
Thorsten Liebig, derivo GmbH - Ulm, Germany
Michael Mohler, Language Computer Corporation in Richardson, USA
Michael Schmidt, fluid Operations AG, Germany

SEMAPRO Publicity Chairs
Felix Schiele, Reutlingen University, Germany
Bernd Stadlhofer, University of Applied Sciences, Austria
Ruben Costa, UNINOVA, Portugal
Andreas Emrich, German Research Center for Artificial Intelligence (DFKI), Germany

SEMAPRO 2014

Committee

SEMAPRO Advisory Chairs

Wladyslaw Homenda, Warsaw University of Technology, Poland
Bich-Lien Doan, SUPELEC, France
Alexey Cheptsov, High Performance Computing Center Stuttgart (HLRS), Germany
Shu-Ching Chen, Florida International University, USA
Sule Yildirim Yayilgan, Gjøvik University College, Norway
Jesper Zedlitz, Christian-Albrechts-Universität Kiel, Germany
Soon Ae Chun, City University of New York, USA
Fabio Grandi, University of Bologna, Italy
David A. Ostrowski, Ford Motor Company, USA
Andrea Perego, European Commission - Joint Research Centre, Ispra, Italy

SEMAPRO Industry/Research Liaison Chairs

Riccardo Albertoni, IMATI-CNR-Genova, Italy
Panos Alexopoulos, iSOCO S.A., Spain
Sofia Athenikos, IPsoft, USA
Isabel Azevedo, ISEP-IPP, Portugal
Sam Chapman, The Floow Limited, UK
Daniele Christen, Parsit Company, Italy
Frithjof Dau, SAP Research Dresden, Germany
Thierry Declerck, DFKI GmbH, Germany
Alessio Gugliotta, Innova SpA, Italy
Peter Haase, Fluid Operations, Germany
Shun Hattori, Muroran Institute of Technology, Japan
Xin He, Airinmar Ltd., UK
Tracy Holloway King, eBay Inc., USA
Lyndon J. B. Nixon, STI International, Austria
Zoltán Theisz, evopro Innovation LLC, Hungary
Thorsten Liebig, derivo GmbH - Ulm, Germany
Michael Mohler, Language Computer Corporation in Richardson, USA
Michael Schmidt, fluid Operations AG, Germany

SEMAPRO Publicity Chairs

Felix Schiele, Reutlingen University, Germany
Bernd Stadlhofer, University of Applied Sciences, Austria
Ruben Costa, UNINOVA, Portugal
Andreas Emrich, German Research Center for Artificial Intelligence (DFKI), Germany

SEMAPRO 2014 Technical Program Committee

Nasser Alalwan, King Saud University - Riyadh, Saudi Arabia
Riccardo Albertoni, IMATI-CNR-Genova, Italy
José F. Aldana Montes, University of Málaga, Spain
Panos Alexopoulos, iSOCO S.A., Spain
Mario Arrigoni Neri, University of Bergamo, Italy
Sofia Athenikos, IPsoft, USA
Isabelle Augenstein, University of Sheffield, UK
Isabel Azevedo, ISEP-IPP, Portugal
Bruno Bachimont, Universite de Technologie de Compiegne, France
Ebrahim Bagheri, Ryerson University, Canada
Khalid Belhajjame, Université Paris-Dauphine, France
Helmi Ben Hmida, FH MAINZ, Germany
Jorge Bernardino, ISEC - Polytechnic Institute of Coimbra, Portugal
Christopher Brewster, Aston University - Birmingham, UK
Volha Bryl, University of Mannheim, Germany
Dilletta Romana Cacciagrano, University of Camerino, Italy
Ozgu Can, Ege University, Turkey
Tru Hoang Cao, Vietnam National University - HCM & Ho Chi Minh City University of Technology, Vietnam
Nicoletta Calzolari, CNR-ILC (Istituto di Linguistica Computazionale del CNR), Italy
Delroy Cameron, Wright State University, USA
Sana Châabane, ISG - Sousse, Tunisia
Sam Chapman, The Floow Limited, UK
Shu-Ching Chen, Florida International University, U.S.A.
Alexey Cheptsov, High Performance Computing Center Stuttgart (HLRS), Germany
Dickson Chiu, University of Hong Kong, Hong Kong
Smitashree Choudhury, UK Open University - Milton Keynes, UK
Sunil Choenni, Ministry of Security and Justice, Netherlands
Daniele Christen, Parsit Company, Italy
Soon Ae Chun, City University of New York, USA
Paolo Ciancarini, Università di Bologna, Italy
Ruben Costa, UNINOVA - Instituto de Desenvolvimento de Novas Tecnologias, Portugal
Frithjof Dau, SAP Research Dresden, Germany
Geeth Ranmal De Mel, University of Aberdeen - Scotland, UK
Cláudio de Souza Baptista, Computer Science Department, University of Campina Grande, Brazil
Thierry Declerck, DFKI GmbH, Germany
Jan Dedek, Charles University in Prague, Czech Republic
Gianluca Demartini, University of Fribourg, Switzerland
Chiara Di Francescomarino, Fondazione Bruno Kessler - Trento, Italy
Alexiei Dingli, The University of Malta, Malta
Christian Dirschl, Wolters Kluwer, Germany
Bich Lien Doan, SUPELEC, France
Milan Dojčinovski, Czech Technical University in Prague, Czech Republic
Laura Dragan, National University of Ireland, Ireland
Raimund K. Ege, Northern Illinois University, USA
Enrico Francesconi, Institute of Legal Information Theory and Techniques, Italy

Shahab Mokarizadeh , Royal Institute of Technology (KTH) - Stockholm, Sweden
Anne Monceaux, Airbus Group Innovations, France
Alessandro Moschitti, Qatar Computing Research Institute, Qatar
Mir Abolfazl Mostafavi, Université Laval - Québec, Canada
Ekawit Nantajeewarawat, Sirindhorn International Institute of Technology / Thammasat University, Thailand
Vlad Nicolicin Georgescu, SP2 Solutions, France
Lyndon J. B. Nixon, STI International, Austria
Csongor Nyulas, Stanford Center for Biomedical Informatics, USA
David A. Ostrowski, Ford Motor Company, USA
Vito Claudio Ostuni, Polytechnic University of Bari, Italy
Peera Pacharintanakul, TOT, Thailand
Andrea Perego, European Commission - Joint Research Centre, Ispra, Italy
Livia Predoiu, University of Oxford, UK
Hemant Purohit, Wright State University, USA
Jaime Ramírez, Universidad Politécnica de Madrid, Spain
Isidro Ramos, Valencia Polytechnic University, Spain
Werner Retschitzegger, Johannes Kepler University Linz, Austria
German Rigau, IXA NLP Group. EHU, Spain
Juergen Rilling, Concordia University, Canada
Tarmo Robal, Tallinn University of Technology, Estonia
Sérgio Roberto da Silva, Universidade Estadual de Maringá, Brazil
Alejandro Rodríguez González, Centre for Biotechnology and Plant Genomics, UPM-INIA, Spain
Marco Rospocher, Fondazione Bruno Kessler (FBK), Italy
Thomas Roth-Berghofer, University of West London, U.K.
Michele Ruta, Politecnico di Bari, Italy
Gunter Saake, University of Magdeburg, Germany
Melike Sah, Trinity College Dublin, Ireland
Satya Sahoo, Case Western Reserve University, USA
Adriano A. Santos, Universidade Federal de Campina Grande, Brazil
Minoru Sasaki, Ibaraki University, Japan
Felix Schiele, Hochschule Reutlingen, Germany
Michael Schmidt, fluid Operations AG, Germany
Kinga Schumacher, German Research Center for Artificial Intelligence (DFKI) - Berlin, Germany
Wieland Schwinger, Johannes Kepler University Linz, Austria
Floriano Scioscia, Politecnico di Bari, Italy
Giovanni Semeraro, University of Bari "Aldo Moro", Italy
Kunal Sengupta, Wright State University - Dayton, USA
Luciano Serafini, Fondazione Bruno Kessler, Italy
Md. Sumon Shahriar, Tasmanian ICT Centre/CSIRO, Australia
Sofia Stamou, Ionian University, Greece
Vasco Soares, Instituto de Telecomunicações / Polytechnic Institute of Castelo Branco, Portugal
Ahmet Soylu, University of Oslo, Norway
Mari Carmen Suárez-Figueroa, Universidad Politécnica de Madrid (UPM), Spain
Lars G. Svensson, German National Library, Germany
Cui Tao, Mayo Clinic - Rochester, USA
Saïd Tazi, LAAS-CNRS, Université de Toulouse / Université Toulouse1, France
Zoltán Theisz, evopro Innovation LLC, Hungary

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Semantic Web GIS Services for Cultural Heritage Domain

Caner Güney
Geomatic Engineering
Istanbul Technical University
İstanbul, Turkey
guneycan@itu.edu.tr

***Abstract*—World is a collection of objects of cultural and natural heritage resources. Every human activity happens somewhere and sometimes. Each application projected in the Cultural Heritage sector takes a different view of the time period of the Cultural Heritage resource. In other words, projects belonging to the same time period but different geographical locations could be correlated with each other. However, users from all over the world are still faced with the perennial problem of finding those resources that will be most relevant to any particular research project. Cultural Heritage documentation is definitely going digital, but this trend may not be able to solve the problems arising when it is desired to perform e-heritage solutions in order to share Cultural Heritage knowledge. On the other hand, Cultural Heritage is a promising application domain for semantic web technologies due to the semantic richness and heterogeneity of cultural content. In this study, a coherent and standardized architectural framework -'GeoGCHEAF- has been designed as a "Semantic Geospatial Information System (GIS) Services" and proposed to the Cultural Heritage domain.**

Keywords-cultural heritage; geospatial informatics; semantic web technologies; ontologies.

I. INTRODUCTION

Clues from the past life styles and habits of the mankind have always been interesting and valuable to people who are living on the same land at different time. Documentation and protection of the historical places and structures is not only a local or national issue, but also a global interest to keep the memory of the past of the mankind. Discovering and comprehending habitats and creations of mankind in the past, not only adds to knowledge, but also unfold rich heritage setting conservation responsibilities for societies. The expectations from the local, national and international authorities are highly increased to protect the historical areas for the next generations. Currently, numerous Cultural Heritage (CH) recording, documentation, conservation, restoration, reconstruction, renewal, rehabilitation, digital preservation projects, etc., are in progress [1].

Archaeologists may be committed to studying the past, but their use of technology is quite up-to-date so that digital heritage, e-heritage, digital archaeology, virtual archaeology and open archaeology are fast-moving fields. In the digital age with ever-increasing quantities of CH data being collected, stored, and distributed in computer-readable forms, interconnection of information is becoming essential.

Organizations from across the CH sector have been able to take advantage of Information and Communication Technology (ICT) to offer new forms of access to their resources for users. They are creating geoservices, moving from data sharing to sharing resources, such as maps, models, data, content, knowledge. Many web-based GIS applications of CH resources are being designed and implemented all over the world. However, users from all over the world are still faced with the perennial problem of finding those resources that will be most relevant to any particular research project. Furthermore, it is difficult for the data/information/content/application/service to be integrated because it is stored in stove-piped systems or because two CH communities use different terminology to describe the data/information.

To conclude this, the "Geo-enable Global Cultural Heritage Enterprise Architecture Framework (GeoGCHEAF)", a global internet-connected spatially-enabled CH sharing network based upon the open standards and semantic technologies, has been specifically designed to expand communication and dissemination of the CH data, information, knowledge, content, applications and services to the different levels of users and the public.

The aim of this study is to promote the digital preservation, integrate the heterogeneous sources using semantic web technologies and make them available primarily to a wider audience of researchers, specialists and decision makers but also to the general public in order to foster wider understanding of the past.

The rest of this article is structured as follows: Section II discusses why Semantic Web technologies are needed in the CH domain. Section III explains the use of ontologies in the CH field. The article concludes by presenting conclusions and recommendations.

II. MOTIVATION

CH data, information, knowledge, content management and research are inherently distributed among many users, projects, organizations, systems, enterprises, applications and services. Each CH organization develops some, but not all, of its data/information content. At least some of the resources come from outside the organization. Moreover, many of today's CH organizations rely on digital ICT to gather, organize, interpret, and disseminate data, information and knowledge relating to their various projects. In many cases, this involves applications and services that were

 ISBN: 978-1-61208-355-1

created at different times and designed for heterogeneous hardware and software platforms. The challenge now faced by these organizations is not only data, information, content distributed management, but also the CH organizations increasingly face the challenge of providing efficient and effective methods, such as integrating various distributed open web services, loosely-coupled applications, geoinformation technologies and infrastructures for CH resources into a single semantic interoperable framework, by which these disparate technologies can work together to achieve academic and/or commercial objectives that are constantly evolving [2].

Not only spatial data/information sets, topographic and thematic maps, vector and raster layers but also demographic data, geo-demographic data, census data, archaeological, architectural, historical information (including date of recording, recording by, structural changes (e.g., shape, size, width, length, height), construction date-material, technique, archaeological finds (e.g., ceramic, lithic, metal, textiles, bone) and other geodata/geoscience data (e.g., geomorphological information, earthquakes, fault zone maps, climate change information), and GIS files/contents are shared via open standard data and information formats, such as Extensible Markup Language (XML), Geographic Markup Language (GML), compact GML (cGML), CityGML, Keyhole Markup Language (KML), Geospatial JavaScript Object Notation (GeoJSON), Web Ontology Language (OWL) and services on the web through a geoweb portal among CH scientific community, decision-makers, NGOs, field teams, authorities and public. This is because long-term conservation depends on the involvement of people from all levels, from government structures to experts to public.

III. SEMANTIC ARCHITECTURE

A. *Semantic Approach*

The key to faster, better, discovery of CH information is metadata, which can be quickly and thoroughly searched by computers and presented in an understandable form to users. Therefore, the CH domain needs standardized metadata entries (e.g., Resource Description Framework, RDF) and a standard metadata framework or frameworks (e.g., RDF Schema, RDF-S). CH spatiotemporal data, information, content and application are encoded and presented with a structured XML document along with its standard CH-specific & CH community-wide defined schema, such as XML CIDOC-CRM, MidasXML, OCHRE (formerly XSTAR), ArchML, Dublin Core or combinations of these, rather than a common XML schema, that can be validated to ensure data integrity and coherence without the need for human interaction. The benefits of an ontology-driven database search are potentially enormous. Effective XML-based data/information integration among the distributed heterogeneous systems, applications, databases, web portals/portlets, data providers and CH specialists are performed through ontologies (e.g., OWL).

When conducting online portlet-based research, aggregating information from these searches across the different datasets, and making data available from different sites in different locations for different user groups need dynamic interoperable data sharing on a global scale for the CH domain via XML-formatted datasets. Whichever method is used to support technical interoperability, including data, information, application, services, process, policy and rules interoperability, web-portals with portlet specifications/protocols (e.g., Java Specification Request (JSR), Web Services for Remote Portlets (WSRP)) also need to achieve semantic interoperability between databases to return useful sets of results to its users to share information on the semantic web.

Ontologies play a critical role in associating meaning with data such that computers can understand enough to meaningfully process data automatically. Compared to syntactic means, the semantic approach leads to high quality and more relevant information for improved decision-making. Equally important is the use of ontologies to achieve shared understanding. Ontologies are also evolving as the basis for improving data usage, achieving semantic interoperability, developing advanced methods for representing and using complex metadata, correlating and integrating information, knowledge sharing and discovery.

Ultimately, ontologies can be an important tool in expediting the advancement of related sciences, and they can reduce the cost by improving sharing of information and knowledge. In such an architecture, distributed repositories can be searched and relevant information according to user specified criteria are found and merged by means of an intelligent web agent or web services through the semantic web. For instance, a sort of specific artifact in the Ottoman fortresses of “Seddülbahir” and “Kumkale” belongs to 17th century can be searched in different CH projects’ databases and portals, digital archives, museum collections and old antiquarian reports [3].

The goal of the semantic architecture of “GeoGCHEAF” is to develop a semantic solution for providing a great level of geospatial semantic interoperability, enabling knowledge sharing and geospatial information integration at different levels of granularity. This open and interoperable semantic solution based on the explicit use of geo-ontologies through geospatial semantic web also provides a cooperative human-computer environment for the composition of spatial- and context-aware semantic web applications in a dynamic and flexible manner within the Internet-connected CH domain. While such a semantic approach facilitates geospatial information storage, search processes, query formulations and retrieval models on the heterogeneous distributed repositories, the ultimate goal of this architecture is to develop knowledge-based spatial information web services for the CH domain [3].

If different web sites that contain CH information share and publish the same underlying ontology of the terms they all use, then computer agents can extract and aggregate information from these different sites. The agents can use this aggregated information to answer user queries or as input data to other applications. This enables the communication and collaboration inside the CH domain and

 ISBN: 978-1-61208-355-1

among the domains and improves understanding of how different CH enterprises exchange geospatial information.

B. Ontology Design Methodology

There is no single correct way or methodology for designing ontologies. Ontology design is a creative process of modeling the given domain, by choosing the most important concepts and identifying the most relevant relations between them. Hence, no two ontologies designed by different modelers would be the same. The potential applications of the ontology and the designer's understanding and view of the domain will undoubtedly affect ontology design choices. The quality of the ontology can be assessed only by using it in applications for which it was designed. Thus, an iterative process has been addressed to ontology design methodology in this research. It is started with a rough first pass at the ontology. Then, the evolving ontology is revised and refined, and filled in the details. Along the way, the modeling decisions that a designer needs to make, as well as the pros, cons, and implications of different solutions are discussed. That is, deciding what the ontology is going to use for, and how detailed or general the ontology is going to be will guide many of the modeling decisions down the road. Among several viable alternatives, it is needed to determine which one would work better for the projected task, be more intuitive, more extensible, and more maintainable. It is also needed to remember that an ontology is a model of reality of the world and the concepts in the ontology must reflect this reality. After the initial version of the ontology is defined, users can evaluate and debug it by using it in applications or problem-solving methods or by discussing it with experts in the field, or both. As a result, it is almost certainly needed to revise the initial ontology. This process of iterative design will likely continue through the entire lifecycle of the ontology [4].

For the purposes of this research an ontology that helps to present objectivity as agreement about subjectivity is a formal explicit semantic definition of set of concepts and relations in the CH domain. The methodology for designing ontologies attempts to establish the types of objects (e.g., fortress, mosque, tower); relations (e.g., Hadice Turhan Sultan built the fortress, commander managed the fortress); events (e.g., World War I, repairment of the structures); and processes (e.g., deterioration, architectural changes) at different levels of scale and granularity, from out of which the geospatial domain is constituted. In order to resolve the conceptual and terminological incompatibilities on case-by-case basis, developing such an ontology, once and for all, includes:

- Underlying conceptualization (conceptual ontologies): Determination of what is wanted to model, checking whether existing ontologies can be reused, if there is, drafting the ontology by making use of existing one. Embracing conceptual issues concerning what would be required to establish an exhaustive ontology of the geospatial and CH domains. Establishment of explicit formal and consensual specification of the concepts with their definitions and the relations among them populating in the CH domain. Underlying conceptualization can be performed by interacting with the specialists in the area of the application, such as scientists/researchers/ontologists from CH domain.
- Ontological commitment to this conceptualization (CH domain-specific logical ontologies): Determination of what certain terms mean in the CH domain and what terms the CH community uses for certain concepts. Preparation of the robust, comprehensive and shared taxonomies (canonical reference taxonomy) of the terms exisiting in the CH domain, which are sufficiently detailed to capture the semantics of the CH domain, and definition of classes and properties.
- Geo-ontological commitment to the abstraction (Geospatial domain-specific logical ontologies): Representations of classification of geospatial entities of real world/spatial phenomena (canonical formalization), their properties and relations within geospatial domain.
- Logical statements (semantic relations): Generation of the rich (thematic-spatial-temporal) relationships among the classes within the CH domain and between geographic entities/features within the geospatial domain.
- Associative relations: Synonymy, hyponymy, hypernymy, and antonymy are semantic relations defined between related words and word senses. Synonymy (syn same, onyma name) is a symmetric relation between word forms. Hyponymy (sub-name) and its inverse, hypernymy (super-name), are transitive relations between sets of synonyms. Antonymy (opposite name) is synonymous with opposite.
- Hierarchical relations (subclass–superclass relations)
- Inheritance or generalization/specialization or taxonomic relationships (superordinate-subordinate relationships): "is-a-kind-of" and "has-kinds" relationships
- Aggregation or partonomic relationships (part-whole relations): "is-a-part-of" and "has-parts" relationships
- Quantitative spatial relations
- Distance: quantitative distance relations (within a specified distance), for instance, space distance, such as "withinMetersOf", or time distance, such as "withinMinutesOf."
- Qualitative spatial relations (vague spatial relationships): includes relative locational properties of objects, such as containment, distance and directions, (near, north, between, inside, outside, in front of (a mosque), along (a street))
- Distance: qualitative distance relations
- Direction: the 8-sector model to express the cardinal directions North, NorthEast East, SouthEast, South, SouthWest, West, NorthWest, or isNorthOf, isLeftOf, isBehindOf
- Topological relationships: the OpenGIS Simple Features Specification of topological relations based on the Dimensionally Extended 9-Intersection Model Based on Components (DE-9IMBC), the ontology includes the following eight relations: equals, disjoint, intersects, touches, crosses, within, contains, and overlaps.
- Mereological relationships: Parthood relations, e.g., "isWholeOf", "isPartOf". Region Connection Calculus

(RCC-8) abstractly describes regions by their 8 basic relations: disconnected, externally connected, equal, partially overlapping, tangential proper part, tangential proper part inverse, non-tangential proper part, non-tangential proper part inverse.

- Semantic mediators (semantic translators): Connection between CH domain ontologies and geo-ontologies is made by semantic mediators.
- Formalizing ontologies (translating ontologies into ontology-derived classes): An object-oriented mapping of multiple ontologies to the system classes. Translation of the ontologies specified in a standard, system-independent form into classes that are specific computer language representations, that is, machine-interpretable definitions of the concepts. Ontology-derived classes are software components that can be used to develop applications and they are fully functional classes with all the operations that can be applied to entities.
- Defining slots and describing allowed values for these slots: Each class describes various features and attributes of the classes and instances.
- Implementation: Establishment of the mapping between information sources and the common ontology. Mapping the ontology into the basic data models and representations necessary for scientific computing about CH domain and geospatial phenomena, and semantic associations in applications that integrate data, metadata, and knowledge queries.
- Creating a knowledgebase: Creating a knowledgebase by defining individual instances of these classes. Filling in specific slot value information and additional slot restrictions for instances.
- Validation: Definition of test cases in the CH domain to validate the ontology being developed.

C. Implementation Methodology for Building Ontologies

Protégé [5] has been chosen to use as an ontology-development environment to specify the ontologies since it is free, open source, and supports a wide variety of plugin and import formats, such as Web Ontology Language (OWL) [6] and RDF-S [7]. In addition, OWL has been adopted as a web-based ontology language to present ontologies and represent knowledge. Semantic web contents and declarative frame-based ontologies in this research are being currently developed using the Protégé-OWL plugin. Protégé-OWL editor is able to present conceptual modeling of the CH domain, edit ontologies developed, create classes, slots, facets, and instances.

The Geographic Markup Language (GML) provides a syntactic approach to encoding geospatial information through a language in which symbols need to be interpreted by users, because associated behavior is not accounted for [8]. GML can be viewed as an alternative not just to geography in RDF, but to RDF itself. These are the differences, data model and type system. GML is built on the XML data model and the XML Schema type system. RDFMap and RDFGeom are built on the RDF data model, and RDF Schema or OWL can be used to express typing information. OWL is the appropriate choice for this job, since its expressiveness corresponds more closely to that of XML Schema. The application of RDF to geography is at an early stage, whereas GML is a mature effort. RDFMap combined with the companion RDFGeom language cover only a fraction of the ground covered by GML3 [9]. In this research, the GML of geospatial instances obtained from the spatial datasets has been translated into OWL using XSLT style sheet.

Fortress is a term in the CH domain ontology and "Seddülbahir Fortress" is an instance of the fortress that is-a-kind-of a CH site. The renovation project directors consider the fortress as a high-level ontology that a consensus can be reached about which are the basic properties of the fortress. From the point of view of this ontology, the fortress is an Ottoman architectural monument belongs to 17th century and built by Hadice Turhan Sultan at the entrance of the Dardanelles. The fortress can be seen differently by different systems, such as architecture sub-domain, archaeology sub-domain, art-historian sub-domain, etc. For the architecture sub-domain the fortress can be building. For the archaeology sub-domain it is an excavation site. For the art history sub-domain it is a recreation site. Although the conceptualization of the architectural sub-domain of the fortress is derived from higher level, architecture sub-domain has a view (building concept of the fortress, e.g., the structural characteristics of the buildings of the fortress) that is more detailed than the previous higher level. This is done using inheritance. Architecture sub-domain will have all the basic properties defined in the higher level ontology plus the add-ons that the architects think are relevant to their concept of fortress. The same happens with the other sub-domains. Inside the archaeology sub-domain, the section in charge of the excavation will have an even more detailed view of the fortress. If all sub-domains inherit from higher level ontology, they will be able to share complete information at this level only, although they can share partial information at lower levels. The users have the means to share information through the use of common classes derived from ontologies. The level of detail of the information is related to the level of detail of the ontology.

The semantic architecture of "GeoCHEAF" stores entities and their associated relationships in the knowledgebase, classifying them according to a hierarchical entity class tree. A given entity can belong to multiple entity classes, that is, there are classes of concepts, which constitute a hierarchy with multiple inheritances. Figure 1 shows an example of a graph representing the ontology of buildings of a fortress as an OWL. The class 'building' is a subclass of the class 'fortress', and the class 'tower' is subclass of the class 'building'. Other branches of the class tree contain buildings with subclasses Turkish bath, wall, mosque, and military barrack. Classes typically have instances, for instance, a specific tower is an instance of the 'tower' class, such as Algerian Tower, Cannon Tower, South Tower, Poyraz Limanı, Lodos, Tophane of the "*Seddülbahir Fortress*". A 'tower' class/entity must have a geometric shape, for example, the round or polygonal plan of the tower. Conceptually, a tower can be placed on both types of tower figure; however, a specific tower can only reside on either.

 ISBN: 978-1-61208-355-1

For example, the instance of Algerian Tower is-a prominent rounded seaside tower or South tower is-a hexagonal tower-hexagonal tower ako tower.

The use of multiple inheritances allows an application developer to make use of the existing ontologies to build new classes. The application developer can combine classes from diverse ontologies and create new classes that represent user needs. These new classes represent objects that have diverse characteristics [10][11][12].

For instance, towers have geometric characteristics along with alphanumeric attributes. Instead of having a single class that needs to include information on the geometric shape of the tower, as well as associated information about construction date, construction material, construction techniques, and so on, multiple inheritance is utilized in this research by inheriting geometric characteristics and methods from a geometric/spatial class of spatial ontology, such as polygon, and inheriting/descending application-specific characteristics and methods from a more generic Tower class (parent class) of CH domain ontology. In the first group, all necessary representational and locational data can be handled by inherited methods, while in the other information on the semantics and behavior of the tower are inherited from CH specific ontology-derived classes. The views can be combined enabling the user to have a geometric/alphanumeric view. An example of the use of this combined view is a "point-and-click" operation over a tower that highlights its shape and shows its alphanumeric data.

In the knowledge generation phase of the semantic architecture of "GeoCHEAF", the ontology editor stores a formal representation of the ontologies and provides a translation of the ontologies from multiple independent data sources into software components (e.g., Java classes) to be used in a semantic web applications, such as information retrieval, web mining. These classes are linked to geospatial information sources through the use of mediators. The application developer can combine classes from diverse ontologies and create new classes that represent the user needs. For the knowledge use phase, the ontologies are available to be browsed by the end user using ontology browser at different levels of detail depending on the ontology level used, and they provide semantic metadata on the available information. The ontology browser can be used during ontology specification by users who wish to collaborate in composing a shared ontology. Once the ontology has been specified, the ontology browser is used to show the available geospatial entities to the users. Hence, the user can query and update the ontologies using remote applications on the Internet. The query processor matches the terms in the user ontology to the system component ontologies. The information about ontologies is provided by the ontology server that holds a standard catalog of ontologies for the user to search and browse, using mappings between ontologies and the structures in data repositories. The connection with the information sources is done through mediators that are pieces of software with embedded knowledge. Mediators connect instances of the entities available in the ontology server to features in spatial databases and translate them into a format understandable for the end user. Figure 2 shows the proposed framework, in terms of its components and their intuitive relations.

For instance, a researcher wants to make cross-archive searches on distributed digital archives encoded in RDF/OWL using the CIDOC-CRM ontology in order to retrieve information, or execute a complex query about the CH data/information on the web. First, the researcher browses the ontology server looking for the related classes. After that, the ontology server starts the mediators that look for the information and return a set of objects of the specified class. The results can be displayed or can undergo any valid operation, such as CH analysis. This ontology-based approach allows CH researchers to associate geospatially referenced data to any other non-spatial information related to the geospatial feature that is expressed on the semantic web.

Existing web service technologies (Remote Procedure Call (RPC) or Representational State Transfer (REST)) are only at the syntactic level and fail to capture enough semantic data, there are semantic gaps in cross-domain resource discovery, heterogeneous resource query, resource translation from one domain to another at the semantic level [13]. Semantic web technology can alleviate this limitation and Semantic Web technologies have been widely used to support automatic service composition. Semantic Web Services deal with such limitation by augmenting the service description with a semantic layer in order to achieve automated discovery, composition, monitoring, and execution, which are all highly desirable processes [14].

The concrete GI services which meet those conceptual needed GIS data and function can be automatically discovered in the semantic repository. Discovered GI services can also be automatically composed as a workflow (service chain) to generate an initial result for users to evaluate. Ontology Web Language for Services (OWL-S) was chosen as a proper workflow description language to enable automatic web service discovery, invocation, composition into a workflow, and interoperation. Automatic workflow chaining utilizes business logic in integrating applications to construct a new application and executes an OWL-S composite process with service groundings.

IV. CONCLUSION AND FUTURE WORK

The geo-ontologies embodied in the geospatial semantic web approach provide a shared understanding and conceptualization of relevant aspects of the CH domain applications. Independent applications that interpret and process CH data with respect to these ontologies can achieve a much higher level of interoperability and information/knowledge sharing. This proposed Knowledge-based Spatial Information Systems and Services and services can play an important role in enabling geospatial-based information and knowledge sharing in the world of interoperable knowledge-based distributed environments.

As ontology development technology evolves, the benefits of ontology use will outweigh the costs of developing them. With the success of this technology, large-scale repositories of ontologies will be available in diverse

 ISBN: 978-1-61208-355-1

disciplines, and this work has been developed based upon this assumption.

To share and integrate data, information, and knowledge among the constituents of the CH domain, standardized communication protocols, standardized metadata contents, and interoperable programming interfaces are essential for the success of 'the future of the past'.

In addition to developing technical solutions, a series of recommendations and effective management are required for the frictionless workflow of adaptive information, from local fieldworker to regional heritage curator to national agencies and the public, such as how fieldworkers could report surveys, excavations.

REFERENCES

[1] C. O. Kıvılcım, "Architectural Survey for Documentation of Cultural Heritage with New Sensor Technologies", Master Thesis, Istanbul Technical University, Turkey, 2006.

[2] C. Guney, 2007. "Towards Conceptual Design Of 'GeoHistory'", EPOCH Publications.

[3] C. Güney, "A conceptual design for the development of a customizable framework for the cultural heritage domain," Ph.D. thesis, Istanbul Technical University, Turkey, 2006.

[4] N. F. Noy and D. L. McGuinnes, "Ontology Development 101: A Guide to Creating Your First Ontology, Technical Report," Knowledge Systems Laboratory, Stanford University, 2001. Available online at: www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html, June 30th, 2014

[5] Protege, http://protege.stanford.edu/

[6] OWL, http://www.w3.org/2001/sw/wiki/OWL

[7] RDF-S, http://www.w3.org/wiki/RDFS

[8] M. J. Egenhofer, "Toward the Semantic Geospatial Web," 10th ACM International Symposium on Advances in Geographic Information Systems, 8-9 November 2002, Virginia, USA (New York: ACM), pp. 1-4.

[9] C. Goad, 2004, RDF versus GML. Available online at: www.mapbureau.com/gml, June 30th, 2014

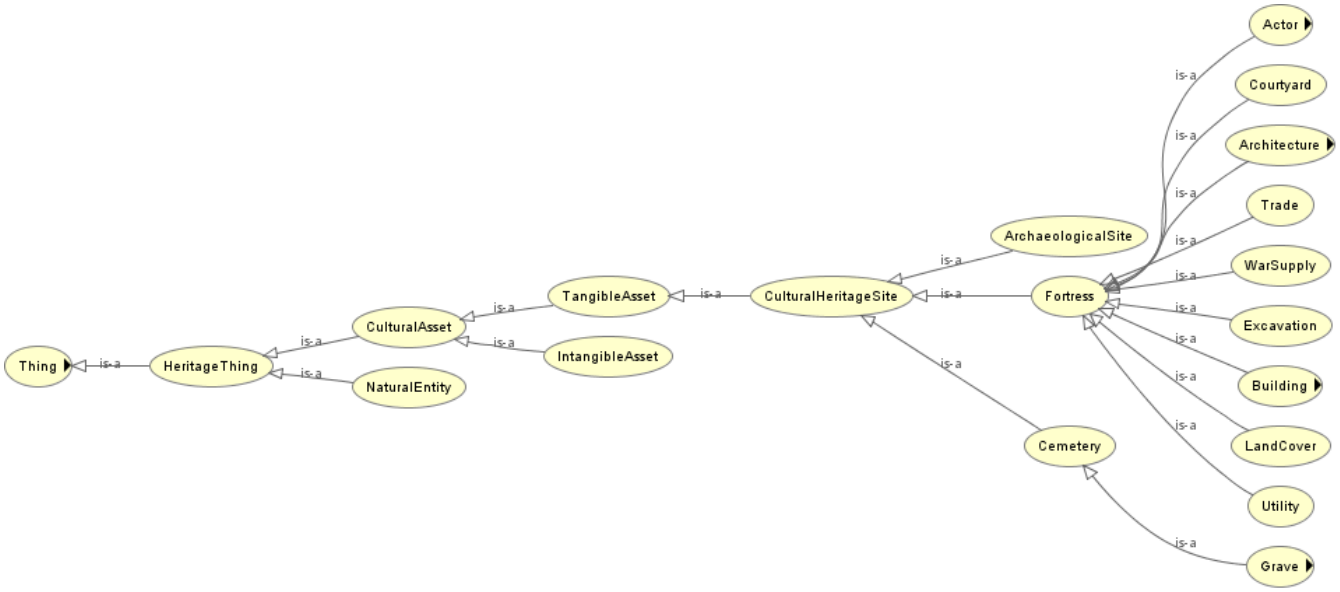
[10] F. T. Fonseca and M. J. Egenhofer, "Ontology-Driven Geographic Information Systems," 7th ACM Symposium on Advances in Geographic Information Systems, 2-6 November 1999, Kansas City, USA (ACM), pp. 14-19.

[11] F. T. Fonseca, M. J. Egenhofer, and C. Davis, "Ontology-Driven Information Integration," AAAI-2000 Workshop on Spatial and Temporal Granularity, August 2000, Austin, USA.

[12] F. T. Fonseca, M. J. Egenhofer, C. A. Davis, and K. A. V. Borges, "Ontologies and Knowledge Sharing in Urban GIS", Computer, Environment and Urban Systems, vol. 24, no. 3, 2000, pp. 232-251.

[13] Y. Gang, "A Research on Semantic Geospatial Web Service Based REST", International Forum on Computer ScienceTechnology and Applications, 5-27 December 2009 China, pp. 208-210.

[14] G. Antoniou and F. Harmelen, "A Semantic Web Primer," Cambridge: The MIH Press, 2008.



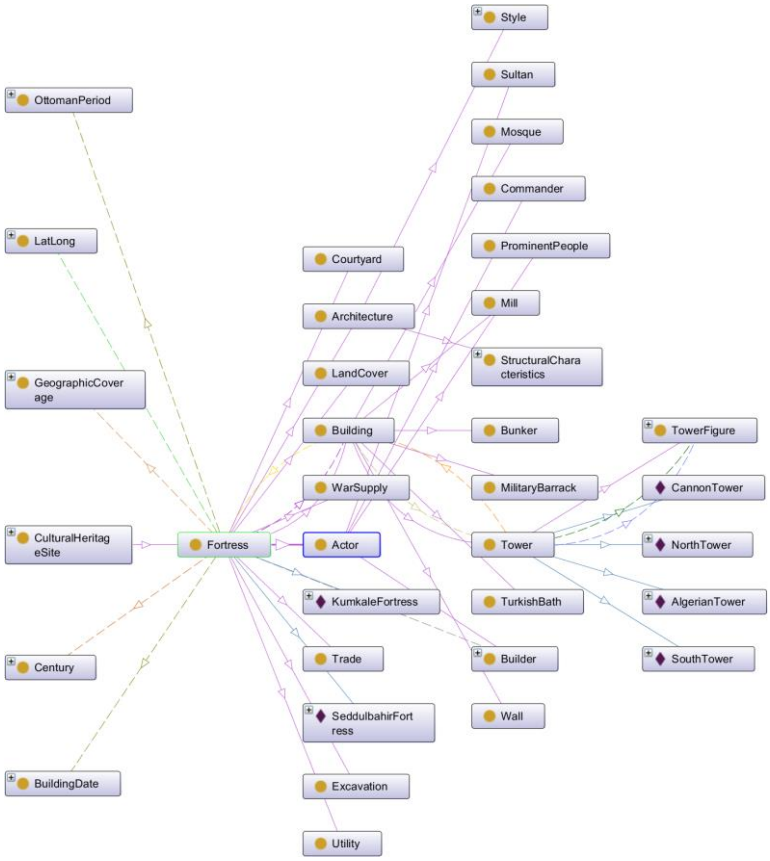


Figure 1. Screen shots of the Protégé-OWL ontology development environment.

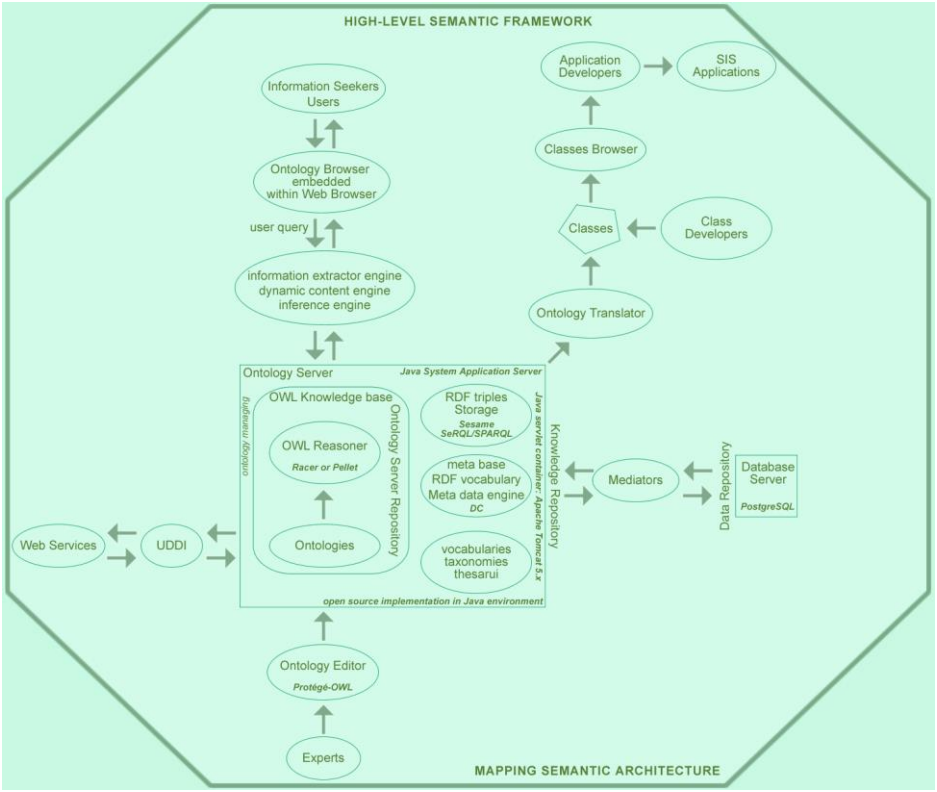


Figure 2. Semantic Architecture of "GeoGCHEAF".

Spacetime: a Two Dimensions Search and Visualisation Engine Based on Linked Data

Fabio Valsecchi and Marco Ronchetti
DISI, Università degli Studi di Trento
Povo di Trento, Italy
fabiovalse@gmail.com, marco.ronchetti@unitn.it

Abstract— **DBpedia is one of the most interesting projects in the arena of the Semantic technologies. However, being able to extract useful information from it is not a trivial task for a user without a specific competence. We present the Spacetime, a two dimensions search engine that provides a simple visualization user interface for making it easy for a generic user to perform certain types of queries on the DBpedia body, and having results shown in a graphic and animated form. Spacetime has been equipped with various features, such as heat maps, time sliding animations, map aggregations, icon map customization and map saving and loading.**

Keywords-DBpedia; Wikipedia; Visualization; GUI.

I. INTRODUCTION

DBpedia is a community effort to extract structured information from Wikipedia, the well known collaboratively edited, multilingual, free Internet encyclopaedia supported by the non-profit Wikimedia Foundation. Wikipedia has over 25 million articles in various languages, written collaboratively by volunteers around the world. Over 4 million articles are present in the English Wikipedia alone. The DBpedia project extracts a subset of information from Wikipedia, and allows asking sophisticated queries on it [1]. DBpedia is also at the core of the Linked Data project [2]. The strength of DBpedia is the capability of answering complex user requests, such as, e.g., "Which European countries have a capital with more than 3 million people in which flows a river longer than 300 km?". The weakness is the difficulty in formulating such queries, due to the complexity of the huge schema that underlies the data. Even though DBpedia has built a "light" ontology for classifying data, the problem still remains extremely difficult. Several approaches to the problem have been developed, over the last few years, to provide user interfaces that attempt to deal (at least partially) with this issue. None of them is fully satisfying.

We identified a subset of the problem, which deals with space-temporal queries, and wrote a user interface, which enables users to perform queries in a simple way, and to get a response in graphical form. Our work is described in the present paper.

This paper is organized as follows: in Section II we shortly present more details of the problem; in Section III we review the attempts to solve it; in Section IV we present our "Spacetime" solution; in Section V we discuss our approach and we draw our conclusions.

II. FORMULATING QUERIES TO DBPEDIA

Which European countries have a capital with more than 3 million people in which flows a river longer than 300 km? Though Wikipedia does not have a page that directly describes this complex set, it contains all the data required for retrieving it. Wikipedia contains information written in natural language, that is very hard for a computer to extract meaning from, but it also contains some tables, called Infoboxes, which present structured information. It is exactly this information that is most valuable source of knowledge harvested by DBpedia, which stores it in a large database. The idea of using this source of information is due to Auer and Lehmann [3]. DBpedia also extracts data from other sources, such as the title of the Wikipedia articles, their categories, interlinks (i.e., the links that connect equivalent articles in different languages), geo-coordinates, redirects (strings used by Wikipedia to identify synonymous terms) and disambiguation (pages that explain the different meanings of homonyms), etc. Also a short and a long abstract are kept for each Wikipedia article.

The DBpedia Knowledge Base (DBKB) contains more than 2.6 million entities [4]. Each entity is defined by a Uniform Resource Identifier (URI), which is described by the common pattern http://dbpedia.org/page/Name, where *Name* is taken from the corresponding Wikipedia article URL. Each entity is composed of a set of Resource Description Framework (RDF) triples. DBKB is composed of around 274 million RDF triples which have been extracted from 35 different Wikipedia language versions. This information is represented using an OWL-based ontology, which was manually created by the members of the community, even though there have been attempts to automatically refine the ontology (see e.g., [5]). The ontology is composed of a large number of classes and properties. The need of having an ontology comes from the fact that the Wikipedia Infobox template system evolved without a central schema for describing entities and their properties. This leads to a situation in which, for instance, the entity Person has an attribute for describing his/her place of birth that can be either "birthplace" or "placeofbirth". The ontology allows centralizing the equivalent property names in a unique property label.

 ISBN: 978-1-61208-355-1

Since the data in DBKB are stored as RDF triples, a natural way to extract information is to perform SPARQL queries. SPARQL is in fact a well-known query language designed specifically to query RDF databases [6]. Although this is in principle enough to solve the problem of extracting information from DBpedia, it is in no way a practical road. First of all, it requires the user to be familiar with SPARQL. This would be equivalent to saying that any Chief Executive Officer can know everything about the company he manages, because s/he only needs to run a SQL query against the company database(s). Although the statement is in principle true, it is not a viable solution.

In order to extract information from a database, one needs to be familiar with its schema. It is necessary to know which entities are represented, which attributes they have and which relations are stored. Hence, to be able to run a query onto DBpedia one needs to be familiar with its ontology, which is composed of 359 classes and 1775 properties. For instance, the class *Person* has properties like *first name, surname, age, birth date, death date, hometown,* etc., while the class *Organisation* has *name, foundation date, hometown, founders,* etc. A deep familiarity with the ontology is needed to be able to write queries. The ontology in itself presents shortcomings. In first place, it was not really "designed", but it is rather the outcome of choices taken by individuals or groups who collaborated in writing the corresponding Wikipedia pages. The ontology is hence partial. For instance, trade fairs do not have an Infobox in Wikipedia pages, hence DBpedia does not have *TradeFair* class in its ontology. Furthermore, it is unbalanced. In fact, some classes have a deeper structure than the others. For instance, the class *Event* has a number of subclasses (*Convention*, *Election*, *FilmFestival*, *MilitaryConflict*, *SpaceMission* and *SportsEvent*). Some are much more developed than others. For instance, *SpaceMission* has 40 attributes while *Election* and *FilmFestival* have respectively 9 and 15 properties. Moreover, some (like *Convention*) are much more general than other (like *SpaceMission*), while many other (such as *TradeFair*) are missing.

Another class of problem is related to the quality of the results rather than to complexity of formulating queries. One of them is related to the completeness of the data. When writing the Wikipedia page, some fields in the Infoboxes can be left blank. This makes sense from the point of view of Wikipedia, which allows progressive evolution of the pages, so that even stubs are allowed. Of course the lack of completeness of the data harms the quality of the query results. As an example, we mention that not all the movies belonging to the *Film* class have the attribute *releaseDate*. At the time of our work this property had a value (in Wikipedia) only on 30943 resources out of the 71715 belonging to the *Film* class (43%). Therefore, any query involving the *releaseDate* attribute will miss over half of the target population, simply because the data are not there.

In the same category also falls the misclassification problem. For example, sometimes a "thing" (represented by a Wikipedia page) is not classified in the correct class but rather in a superclass. For instance, the rock band Pink Floyd is generically classified as *Organisation* rather than as *Band* (*Band* is subclass of *Organisation*). It is worth remarking that when we speak of classification we mean the classification which is provided by the Infoboxes rather that the one given by Wikipedia Categories, which by the way have their own set of problems. For instance, in addition to sharing most of the DBpedia ontology shortcomings we just mentioned, Wikipedia Categories form a non-acyclic graph.

III. PROPOSED SOLUTIONS

Several attempts have been made to help end users extracting valuable information from DBpedia. Here, we do not intend to propose an exhaustive review of all available tools, but rather we arbitrarily choose some examples, as representative of the class of solutions they belong to.

Some are front-ends suitable for exploring the sea of RDF triples, and make it possible to interactively run SPARQL queries. An example is OpenLink Virtuoso. It allows performing research starting from keyword, URI or label. The text search requires the insertion of a text pattern to look for. Then a finder shows a list of entities with the text occurring in any literal property value or label. The entity URI lookup is used inserting entity URI that are recognised by the autocomplete feature of the tool.

Although such sort of tools is certainly useful, it is by no means suited for the end user. Apart the exposition of the technicalities of the query language, the "what can be asked" problem (which requires an understanding of at least a portion of the ontology) is far from being solved.

RelFinder [7] is an example of a different class of tools. It starts from instances instead of from queries. The user specifies two entities, and the system explores the RDF graph to find relations that associate the two instances, and shows the resulting graph to the user. Lodlive [8] is somehow similar. This system allows the the user to choose one instance as a starting point, and then to explore the RDF graph by navigating one of the relations the chosen item is involved in.

Somehow similar is gFacet [9]. It also provides the possibility of navigating the data, but the starting point is now a class instead of an instance. The starting class can be selected by writing a text (part of its name); all classes which contain as a substring in their name the text provided by the user will be shown, and the user will select one. At this point the list of instances is presented to the user. Through class relations, the user can then select a second class, which is linked to the one that was chosen in first place. Selecting an instance in the second class will put a restriction on the first, implicitly solving a query. As an example, if the first class is "Italian Actors" and the relation is "birth place", by choosing Rome in the second class, the box of first class will show all the Italian actors, which were born in Rome. Further restrictions can then be added.

DBpedia mobile [10] takes a different approach, since it starts from a query based on the context. Given the user location, it searches entities, which are nearby geo-located, and shows them on map to the (mobile) user. It is not meant

 ISBN: 978-1-61208-355-1

to explore the whole set of data, but it is rather aimed at providing a location-aware service.

Sgvizler [11] looks at the last part of the process of data retrieval. It provides a way to present the results of a query in graphical form.

Faceted Wikipedia Search (FWS) [12] adopted a faceted search paradigm. This approach enabled users to compose complex questions step by step using facets. A facet is a component shown in the user interface for refining user searches. Facets exploited the properties of an entity to refine the result of a user query. Unfortunately FWS, which had one of the most interesting approaches among the DBpedia-based applications, is now dead since Neofonie, where it was deployed, stopped maintaining the server.

Other (more specific) DBpedia-based applications are listed on a dedicated page on the DBpedia web site [13].

IV. SPACETIME

Spacetime is a tool that aims at making easy for the end user to run a certain subset of queries on the DBKB, and presents the results in graphical form. It considers all the resources in the DBpedia dataset that have at least one spatial and one temporal attribute. This approach allows overcoming the ontology complexity, even though it limits the set of queries that can be formulated. Our requirements for the application were quite simple. We wanted it to be graphically simple and pretty, to be intuitive and simple to use, and able to minimize the knowledge required to the user and its effort in dealing with the interface.

The interface is composed of a control panel for specifying the query parameters, and for saving and loading the resulting maps; a map for visualising the locations of the events found through a search; a timeline for showing the events on the temporal axis (see Figs. 1 and 2).

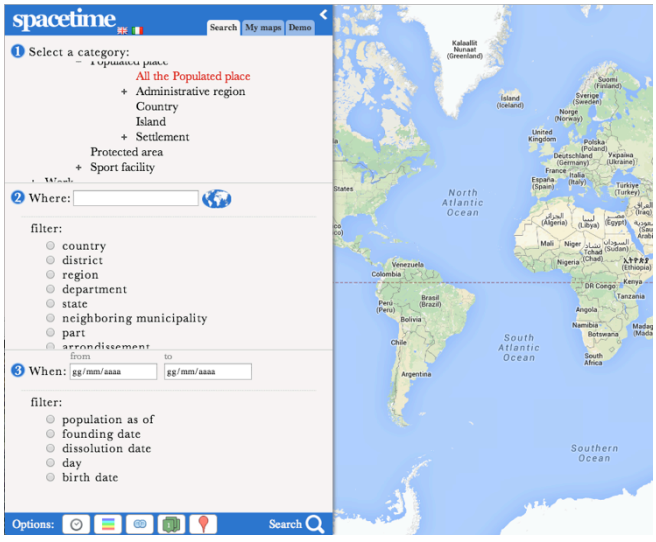


Figure 1. The Spacetime user interface.

To formulate the query, the user has to select "Where", "When" and "What". The "What" comes in the form of what we call "the Spacetime Categories" (SC), i.e., the set of classes that have among their attributes some spatial and temporal data. SC are a (hierarchical) subset of the DBpedia ontology. They are composed of six "top" categories: *Organisation*, *Person*, *Event*, *Place*, *Species* and *Work*. Each of them contains numerous subclasses.

The user starts browsing the SC tree. The user interface is shown in Fig. 1. As it can be seen, there are three sections: Category, Where and When. Once the user selects the suitable class in the upper part, the following sections auto-adapt showing those properties, which have a spatial and temporal dimension. The user can select among those, and put a restriction indicating a geographical place and a time window. When the user fires the request, a SPARQL query is generated and run against the DBKB. The results are rendered on the map in different ways, according to the specific user request, as we shall discuss later.

Behind the scenes, the system performs three types of queries: filter queries, search queries and resources queries.

Filter queries provide the data for the space and time filters in the user interface of the application. These filters are the ones that allow presenting to the user the selectable properties along the space and time dimensions. The time filter selects those attributes which are of data type *xsd:date*. The space filter selects DBpedia properties having a correspondence in latitude and longitude in the Basic Geo (WGS84 lat./long.) Vocabulary. Filter queries are performed as soon as the user selects a category.

Search queries compose the information provided by the user (selected category and properties, and filtering values). An approximation is done in this phase. In fact, the user selects a geographical region by specifying its name, being helped by an auto completion feature. However, geographical data are identified by their coordinates, rather then by logically (or politically) belonging to a geographical entity. Hence, what we do is to use the bounding box of the geographical region specified by the user as a matching filter for the location-dependent properties. This has an obvious problem in terms of precision of the supplied results, as it may include some data belonging to (logically or politically) neighbouring regions.

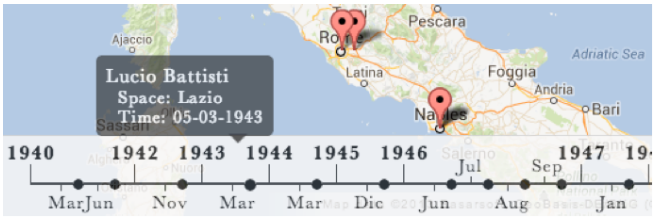


Figure 2. A detail of the shown result set, with a pop-up.

Resource queries are composed in the last phase. Results are graphically shown in a variety of ways. The simplest one (shown in Fig. 2) is in the form of "pins" appearing on the

 ISBN: 978-1-61208-355-1

map, and of dots shown on a timeline. At this point the user can investigate the details; selecting a pin or a dot, information about that specific element of the result set is shown in a pop up.

Typically, the name of the corresponding Wikipedia page is shown, together with the relevant space and time information. Optionally also a short abstract from the corresponding Wikipedia page can be shown. Also links of the resource to Wikipedia and DBpedia are provided, in case the user wishes to obtain additional and more specific information. All this is retrieved by a "resource query", which is a simple SPARQL query asking the DBKB to provide the needed information and the matching abstract, which can be obtained in multiple languages.

Apart of running a query and inspecting the results, the user can save, load, and modify a map. Maps can be saved locally so as to be able to later import them in external resources such as, e.g., a multimedia presentation. Saved maps can be later reimported – e.g., to modify them so as to create a join between the results of two different queries, as we will discuss later.

As we mentioned, Spacetime provides also other types of visualizations. It is possible to create time-sliding animations. Instead of showing all the results at the same time, it is possible to have them ordered along the time axis, and to let them appear in a temporal sequence. As an example, this might show how civilization spread by showing a set of cities in the order in which they were founded.

A second type of presentation shows the events in form of "heat-maps" instead as individual pins (see Fig.3). This is useful in the case one has to show many events: they appear as a density map rather than individually.

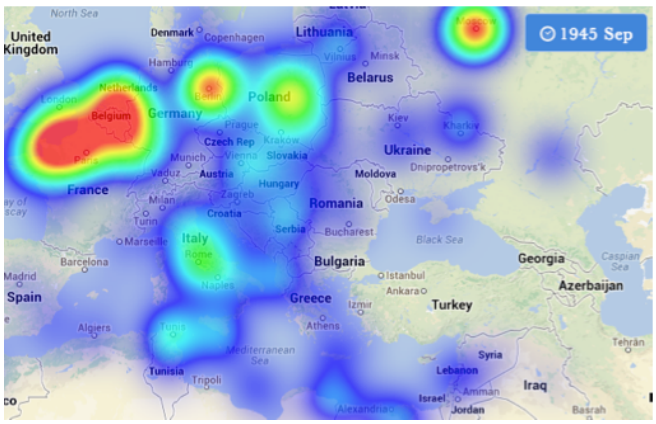


Figure 3. The Europe density map of the military conflict during the Second World War.

Presentations of these two types can be combined, having a density map evolving in time in an animation. One such example is among the demos of the system, which can be seen on the Spacetime web site [13]. It is a query about the battles, which took place during the Second World War. It asks about "the military conflicts occurring in the world between 1939 and 1945 (inclusive)". The density map allows following the evolution of the Second World War, clearly showing how the conflict spread and moved in the world, or, by zooming, in a particular geographical area.

Pin customization allows creating easy-to-understand maps. Pins can be customized by numbering them (in the order they are generated) or by choosing icons among eight categories: colours, numbers, letters, people, culture, events, transportation and sports. Icon colours can also be selected.

Map aggregation allows the creation of more complex maps that include different DBpedia categories. In fact, it is possible to render a map containing resources from categories such as Writer and Book, or different historical events like Election, Military Conflict, and Convention, or to define the historical context when a certain person was born just by aggregating on the map of person birth also a set of historical events. Moreover, the possibility of modifying the icon markers of the resources, allows making the map better in term of meaning and clearness.

An example of pin customization in an aggregated map is shown in Fig.4, which displays the career of the soccer player Zinedine Zidane. Different icons mark the place where he was born, the towns hosting the soccer teams he was playing for, and the places where he won cups. The map is an aggregation of the results of multiple queries.

Figure 4. A map, built as aggregation of the results of multiple queries, shows the career of the soccer player Zinedine Zidane.

All these functionalities were introduced in view of the actual use of the search results: we thought of Spacetime as a tool, which could be used, e.g., when teaching or studying. Hence, it was necessary to think how the user might need to clarify some points, for instance when using those (probably precompiled) results during a lecture, or to incorporate them in a homework.

From a technical point of view, the application is based on five pillars:

• DBpedia: the repository where the knowledge base is kept;

• SPARQL: the query language used for interrogating DBpedia and retrieving the data through a SPARQL endpoint;

 ISBN: 978-1-61208-355-1

• Google Maps: the rendering engine for showing the retrieved data;

• JavaScript: it is at the core of Spacetime, and it contains its application logic. It is responsible of all the interactions between the application components;

• HTML: defines the graphical structure of the Spacetime and the dynamic content of the application.

In more detail, the used technologies are:

•SPARQL Query Language for RDF. Queries are composed by the Javascript engine, and are executed through the SPARQL endpoint;

• JavaScript Object Notation (JSON): the results of the SPARQL queries are produced in this format, which is used for managing the results of the query and for the saving and loading maps;

• Google Maps JavaScript API v.3: the Google Maps API are used for populating a map with the data extracted in the JSON file returned by the SPARQL endpoint;

• JavaScript and JQuery library: the scripting language and its library define a set of functions that are the core of the application. In particular the JQuery library allows the creation of animations inside Spacetime;

•Asynchronous JavaScript and XML (AJAX): this technology is used to have a responsive user interface compliant with the Rich Internet Application paradigm;

• Cascading Style Sheets (CSS): the style sheets language is used for designing the graphical aspect of Spacetime;

• HyperText Markup Language 5 (HTML5): the markup language is used for developing certain part of the application, such as the map saving operation, implemented via the Blob object, and some graphical feature, such as the rounded corners.

V. CONCLUSION

We presented Spacetime, a Rich Internet Application, which deploys the power of Linked Data, and in particular those data, which the DBpedia project gathers from Wikipedia. Our solution does not fully solve the difficult problem of allowing a non-technical user to perform generic queries on the data. However, it provides an easy-to-use interface for a subset of the possible queries. It has been designed to make it possible for a generic user to obtain results that can be embedded in a presentation, or to prepare catching animations.

Spacetime has some limitations. As we mentioned, the geographic selection is made through a bounding box, which might end up in retrieving some data, which are not pertinent to the query. It obviously reflects the weaknesses of Wikipedia and DBpedia in terms of missing information and misclassifications, as discussed in section IV.

The SPARQL endpoint that is used constitutes a single point of failure. If the endpoint has a problem, users cannot perform a search, but they can only load their own maps and work on them. Sometimes errors are generated by the endpoint, as it runs out of its memory pool size. We try to catch these anomalies and to warn the user, but it is not always possible. Unfortunately, the class of problems related to the SPARQL endpoint is out of our control possibilities.

In summary, we think that Spacetime shows in practice the potential of Linked Data, and provides an original solution to part of the problem of building a good and simple interface for the user. Unfortunately, we did not have the time (yet) to run a validation study to support our claims about ease of use and user friendliness.

REFERENCES

[1] S. Auer et al. "Dbpedia: A nucleus for a web of open data." In The semantic web, Springer Berlin Heidelberg, 2007, pp. 722-735.

[2] G. Eason, C. Bizer, T. Heath and T. Berners-Lee, “Linked Data - the story so far.” International Journal on Semantic Web and Information Systems, 5, (3), 1-22, 2009. doi:10.4018/jswis.2009081901 http://dx.doi.org/10.4018/jswis.2009081901 Retrieved Apr, 2014

[3] S. Auer, and J. Lehmann, "What have Innsbruck and Leipzig in common? Extracting semantics from wiki content." In The Semantic Web: Research and Applications, Springer Berlin Heidelberg, 2007, pp. 503-517.

[4] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, U. Becker, R. Cyganiak and S. Hellmann, "DBpedia-A crystallization point for the Web of Data." Web Semantics: Science, Services and Agents on the World Wide Web 7, no. 3, 2009, pp.154-165.

[5] Fei Wu and D. S. Weld, “Automatically refining the wikipedia infobox ontology”. In Proceedings of the 17th international conference on World Wide Web (WWW '08). ACM, New York, NY, USA, 2008, pp. 635-644 DOI=10.1145/1367497.1367583 http://doi.acm.org/10.1145/1367497.1367583 Retrieved Apr, 2014

[6] S. Harris and A. Seaborne, “SPARQL 1.1 Query Language - W3C Recommendation 21 March 2013”. Retrieved April, 2014 from http://www.w3.org/TR/sparql11-query/

[7] P. Heim, S. Hellmann, J. Lehmann, S. Lohmann and T. Stegemann, "RelFinder: Revealing relationships in RDF knowledge bases." In Semantic Multimedia, Springer Berlin Heidelberg, 2009, pp. 182-187.

[8] D. V. Camarda, S. Mazzini and A. Antonuccio, "LodLive, exploring the web of data." In Proceedings of the 8th International Conference on Semantic Systems, ACM, 2012, pp. 197-200.

[9] P. Heim, J. Ziegler and S. Lohmann, "gFacet: A Browser for the Web of Data." In Proceedings of the International Workshop on Interacting with Multimedia Content in the Social Semantic Web (IMC-SSW’08), vol. 417, 2008, pp. 49-58.

[10] B. Becker and C. Bizer, “DBpedia Mobile: A Location-Enabled Linked Data Browser”. 1st Workshop about Linked Data on the Web (LDOW2008), Beijing, China, April 2008.

[11] M. J. Skjæveland, “Sgvizler: A JavaScript Wrapper for Easy Visualization of SPARQL Result Sets”. In: 9th Extended Semantic Web Conference (ESWC 2012), workshop and demo proceedings. Heraklion, Crete, Greece, 2012.

[12] R. Hahn et al. "Faceted wikipedia search." In Business Information Systems, Springer Berlin Heidelberg, 2010, pp. 1-11.

[13] Spacetime. Retrieved Apr, 2014 from http://latemar.science.unitn.it/spacetime/spacetime.html, also reachable from the “DBpedia applications” web site, http://wiki.dbpedia.org/Applications, demos visible from http://latemar.science.unitn.it/spacetime/usecase.htm

 ISBN: 978-1-61208-355-1

Towards Legal Knowledge Representation System Leveraging RDF

Raoul Schönhof, Axel Tenschert, Alexey Cheptsov
High Performance Computing Center Stuttgart,
University of Stuttgart
Stuttgart, Germany
e-mail: raoul.schoenhof@b-f-u.de, tenschert@hlrs.de, cheptsov@hlrs.de

***Abstract*—This paper presents a model usable for a legal system knowledge representation and an implementation of the German Civil Law System as RDF ontology. In this work, different laws are determined in an interconnected structure in order to bridge the gap between computer and social sciences. This model will be created out of natural text, for instance law texts or court decisions, by using a parsing algorithm to build the model, information retrieval tools to extract information and a reasoning algorithm to search and create connections between the particular rules. The focus of this work is to develop the design of the presented model, for an automated reusable entity generation extended by third party knowledgebases.**

Keywords-Knowledge Representation; Law Texts; Ontology; RDF; Big Data; JUNIPER Project.

I. INTRODUCTION

In computer sciences, working with highly unstructured and ambiguous data is a challenge needing to be solved in various research, industrial and social areas. Nonetheless, knowledge is mostly stored implicitly in various formats, e.g., books, articles, websites, data files and so forth. Without an overriding context, these formats contain information. This circumstance and the high complexity leads to the need of improving computer science approaches for enabling social sciences, industries and research to deal with those data. The Resource Description Framework (RDF) [1] syntax allows us to generate relations between instances, consisting of three items: object, subject and predicate. The RDF-Schema (RDFS) enables a mapping of unstructured ambiguous data in a structured manner. Developers are enabled to use RDFS triple stores or ontologies containing logically structured data leading to clearly defined information usable for reasoning tasks. Within the social sciences, there are diverse disciplines like philosophy or political science. The discipline law was chosen because of a well-defined terminology and a clear systematic structure. The thought to exploit legal systems by computer science is old; the first papers about a legal machine were published in the late fifties [2]. Since then, countless approaches have been made. In recent times, there have been several attempts to describe legal knowledge by semantic web languages [3]. Lots of approaches in this area are abstract models. Just a few models were actually generated manually, for example, with the ontology editor Protégé [4]. An automated and realized legal knowledge model for law texts does not exist yet. However, this is necessary; just between 2009 and 2013, Germany resolved 553 federal laws [5] and much more federal state laws.

This work aims to realize a knowledge ontology for the German law system by means of RDF. Center of the law system is the German Civil Code (BGB). It manages and defines fundamental and general issues. The paragraphs are numbered ongoing through the entire BGB. Moreover, most of the single paragraphs are successive subdivided to articles, sub articles and half sentences or numbers. In the scope of this work, German law texts will be explored and structured using RDFS in order to extract information out of this model, being used for automated reasoning processes. By querying the generated RDFS relations, it is possible to comprehend how rules interact and which requirements have to exist to get a legal effect. By matching these requirements with a given case ontology, it could be possible to picture the legal situation of any case. Therefore, this system assists with legal issues by providing legal advice in a fast, user friendly and affordable way.

The paper is structured as follows. Section II gives an introduction into the German legal system and explains briefly, by reference to an example, how different rules can interact together. Section III depicts the system design and shows how legal knowledge ontologies could be generated out of natural texts found in a law book by the use of computer linguistic tools. Conclusively, Section IV deals with the future tasks, as well as the assets and drawbacks.

II. EXEMPLARY SCENARIO

Law texts are not a cluster of isolated rules, but form a complicated network of provision mechanisms and relations. When thinking of relations in law texts, one of the main causes of the complexity of law systems is the aspiration to reduce repetitions as well as the use of an abstract wording. Moreover, the BGB is divided in five chapters. Each chapter manages a special part of possible law issues. The first chapter is called General Part, which is the result of the repetition reduction. It contains mostly definitions and general rules; these are used in the chapters two to five. The second chapter is called Law of

 ISBN: 978-1-61208-355-1

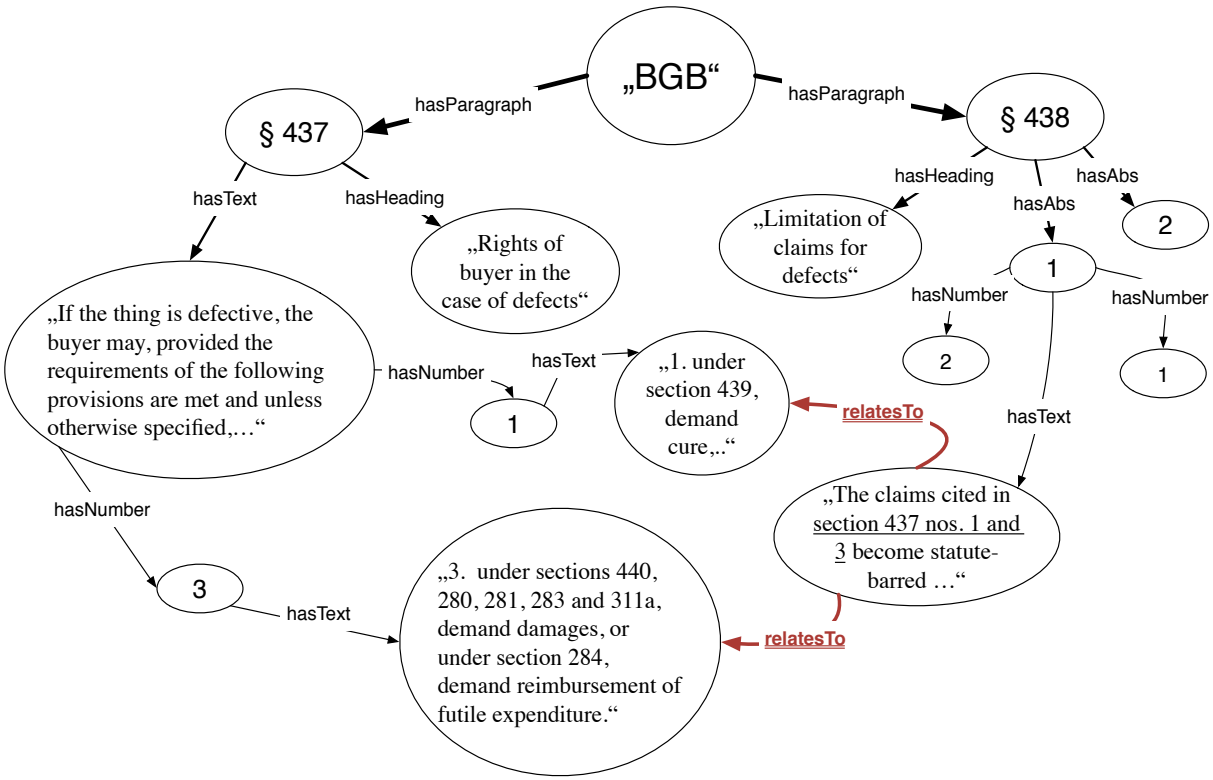


Figure 1: Example of connections in legal text

Obligations. It contains rules to any kind of contract and defines the most common contracts, for example the purchase agreement. This chapter is followed by the Law of Property, the Family Law and the Law of Succession. Especially the separation between general rules and specialized rules makes it possible that two rules regulate one situation in different ways. In such cases, the more general rule is displaced by a more specialized one or a younger rule displaces the older rule. Therefore, rules interact constantly with each other. These mechanisms shall be illustrated based on *§ 437 BGB* and *§ 438 BGB* of the Sales Convention [6]:

§ 437 BGB : “If the thing is defective, the buyer may, provided the requirements of the following provisions are met and unless otherwise specified, 1. under section 439, demand cure, 2. revoke the agreement under sections 440, 323 and 326 (5) or reduce the purchase price under section 441, and 3. under sections 440, 280, 281, 283 and 311a, demand damages, or under section 284, demand reimbursement of futile expenditure. ” [6].

§ 438 I BGB: “The claims cited in section 437 nos. 1 an 3 become statute-barred 1. in thirty years, if the defect consists a) a real right of a third party on the basis of which return of the purchased thing may be demanded, or b) some other right registered in the Land Register, 2. in five years a) in relation to a building, and b) in relation to a thing that has been used for a building in accordance with the normal way it is used and has resulted in the defectiveness of the building, and 3. otherwise in two years.“ [6].

While on the one side, *§ 437 BGB* defines the rights of a buyer in case the purchased object is faulty, *§ 438 BGB* declares on the other side that some of these rights (*§ 437 nr. 1* and *3*) become statue-barred after a certain time [6]. In this example, the rules are connected through named references (see also Figure 1), but it is also common to connect rules through abstract concepts, here for example the word statute-barred which is again defined in *§ 194 BGB*.

The total amount of relations in a legal system is vast, therefore a system is necessary supporting non-jurists by estimating legal issues.

III. System Design

The RDF framework is generated in three consecutive steps, which is shown in Figure 2. In the first step, a parsing algorithm creates an initial RDF ontology out of Extensible Markup Language (XML) files. At this point, the model simply pictures the structure of the law texts. In the second step, additional information are extracted out of the law text by using various computer linguistic tools. This information is added to the RDF model as separated entities. Finally, a reasoning method generates the framework by connecting the extracted concepts and references.

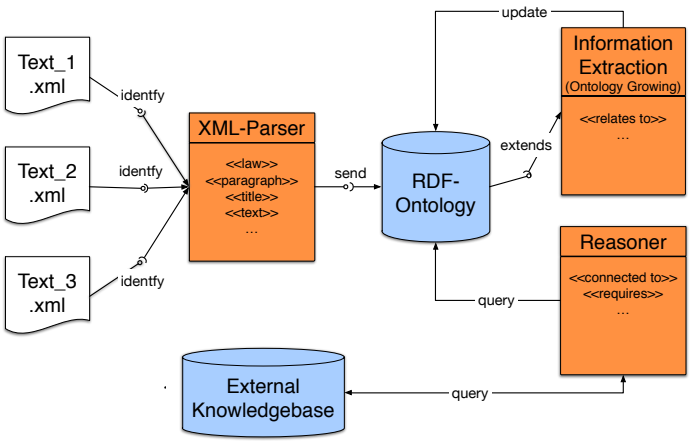


Figure 2: Architecture of the proposed system

A. *Initial RDF Ontology*

The initial model is built by a simple XML-parsing algorithm and creates the hierarchical structure of the law texts in the RDF model. The required XML files with the law texts are open source [6]. The manually provision of XML law files was replaced by an automated crawling algorithm. First, the model contains basic entities, e.g. the law names, the rule numbers and their headings, the particular paragraphs and finally the actual law text. The entities are connected by their own RDF vocabulary called legVoc, which helps to depict the structure of the law texts. Properties of legVoc are for example “hasLaw” to summarize all paragraphs in a law book or “hasSection” in order to connect a paragraph to a superior topic. The structure of the RDF model is illustrated by an extract of *§ 438 BGB* (an example can be found in Figure 3).

```
<rdf:Description rdf:about=”http://gesetzteontologie/BGB/438”>
    <legVoc:hasAbs
   rdf:resource=”http://gesetzteontologie/BGB/438/2”/>
   <legVoc:hasHeading>Limitation of claims for defects
   </legVoc:hasHeading>
   <legVoc:hasAbs rdf:resource=”http://gesetzteontologie/BGB/438/1”/>
</rdf:Description>
<rdf:Description rdf:about=”http://gesetzteontologie/BGB/438/1”>
   <legVoc:hasNumber
      rdf:resource=”http://gesetzteontologie/BGB/438/1/2”/>
   <legVoc:hasNumber
      rdf:resource=”http://gesetzteontologie/BGB/438/1/1”/>
   <legVoc:hasText>The claims cited in section 437 nos. 1 and 3 become
   statute−barred</legVoc:hasText>
```

Figure 3: Listing of RDF extraction

B. *Information Extraction*

After the initial model is generated, information about the content of the given law texts have to be extracted and added to the model, which is one of the most challenging tasks.

Of an extraordinary interest is the identification of concepts in the particular rule as well as its heading. For instance, one of these concepts is ”statute-barred” in *§ 438 I BGB*; shown in Figure 1. The concept identification uses statistical extraction methods as well as pattern-based methods. Especially latter methods are predestinated to identify cross references which are common in law texts. Because of the circumstance that some rules refer to another rule and some rules prohibit the applicability of another rule, the pattern based method has to distinguish between these two cases. Subsequent to the information extraction, the identified concepts are added as RDFS triples to the initial model.

Naturally, these methods will just help to identify entities but they will not be able to extract a very large amount of information, e.g. the relation between a number of entities. Therefore, additional tools have to be used. Meanwhile, there are various text engineering tools which are capable to extract information out of natural text; for instance Text2Onto [7] and Gate [8] with the OWLExporter plug-in [9] as well as Protégé [10] with its plug-in OntoLT [11].

Beside these tools, the Stanford Natural Language Processing Group (SNLPG) at the University of Stanford developed a broad range of computer linguistic tools including a part-of-speech (POS) tagger to break sentences down into their lemma and mark them with their part of speech [12]. SNLPG also provides a special Named Entity Recognizer to find and classify salient nouns, e.g., the noun “London” as a location [13]. Furthermore, a sentence parser, e.g., Stanford Parser [14], is provided which can be used to identify dependencies between words in a sentence.

The information extraction will be done as follows. Firstly, each sentence of the initial RDF ontology will be passed to the POS-tagger which will split each sentence into single words and figures out, which part of speech may be present, e.g., whether it is a noun, a verb or an adjective. Also the POS-tagger references from the words in a sentence to their lemmas. The lemma of nouns are added as isolated entities to the RDF model. After the sentence is tagged by the POS-tagger, the information about the part of speech is used by the Stanford Parser to generate a parsing tree. Dependency parsing is based on a parsing tree that represents a grammatical structure of a sentence, e.g., such as shown in Figure 4 for *§ 1 BGB* [6].

This parser allows it to detect references between verb and noun phrases. These references will be used as properties in the RDF model. Unfortunately, there is no German language support for the Stanford Dependency Parser [15]. Thus, an alternative is necessary which could be the Zurich Dependency Parser for the German language (ParZu) [16].

 ISBN: 978-1-61208-355-1

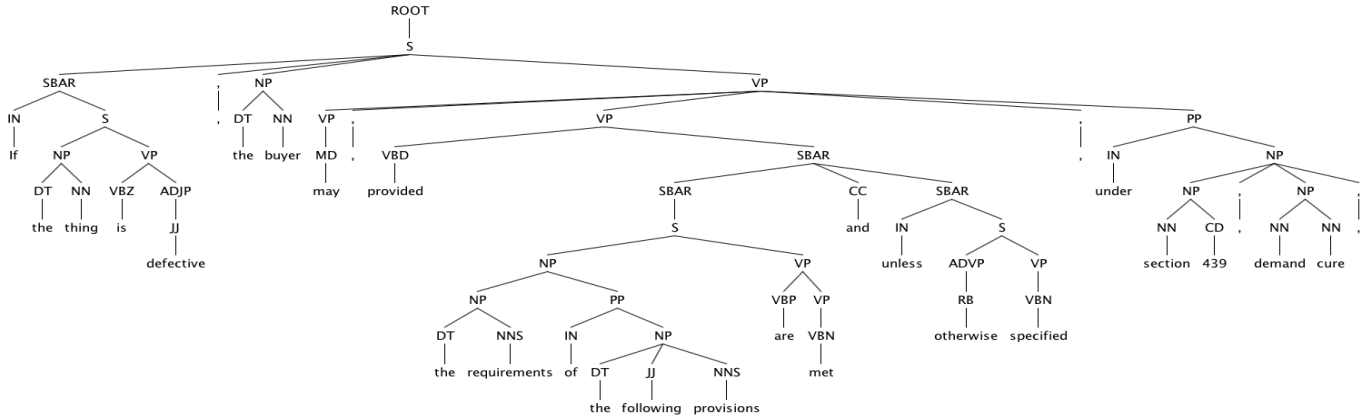


Figure 4: Output of the Stanford Parser

C. Reasoner

The reasoner is in an early stage; its purpose is to identify all rules which could be relevant for a given case. It queries the RDF ontology as well as the case ontology to identify connections between their concepts. Hereby, the reasoner is connected to several already existing knowledge bases which provide additional information like lexical-semantic information from GermaNet [17]. This information is necessary which is illustrated in the following example: A given case mentions the noun "bicycle", which cannot be found in the RDF ontology. For this noun, GermaNet returns a set of synonyms like "bike", "cycle" or "boneshaker" as well as its hyponym "two-wheeler". By following the resulting hyponym-chain, it leads to the words "vehicle" and "thing", which can be found in the RDF ontology. Therefore, each rule, which mentions a "thing", can also be applied for the concept "bicycle".

IV. CONCLUSION

To summarize, the extraction of the initial model out of XML-files was performed with German law texts. Furthermore, the development of an information extraction algorithm is advanced and common concepts were identified. There are also attempts to add information from the universal knowledgebase such as OpenCyc [18]. The advantages of this system design are obvious; it benefits from its high automation-degree enabling the fast adaption to a constantly changing law system. In addition, the RDF ontology is reusable, once generated. Also the system can be modified adapting to different countries and law systems. However, there are several unsolved problems. Firstly, there is no algorithm to transform a rule pattern-based into a logical statement. Secondly, the RDF ontology has to be validated by a test set, which does not exist yet. The third problem is, how inevitably emerging logical inconsistencies shall be handled by the reasoner. It has to be shown if RDFS is complex enough for this purpose, otherwise OWL could be an alternative. It will be the following task to answer this question and to develop the algorithms to create a logical net of statements, definitions and connections in order to solve a simple case automatically. The proposed system implementation will be done leveraging a software technology for Big Data application development, such as JUNIPER Project [19].

ACKNOWLEDGMENT

Authors would like to thank the consortium of the EU-ICT research project JUNIPER (Java platform for hIgh PErformance and Real-time large scale data management) for the support with the Java platform and first prototype of our software system development.

REFERENCES

[1] The RDF Specification: http://www.w3.org/TR/REC-rdf-syntax/ (retrieved: 07, 2014).

[2] L. Mehl: Automation in the Legal World from the Machine Processing of Legal Information to the “Law Machine”, Teddington Conference 1958.

[3] G. Sartor, P.Casanovas, M. A. Biasiotti, M. Fernández-Berrera: Approches to Legal Ontologies, Springer Press, ISBN 978-94-007-0119-9.

[4] P. Mazzega, D. Bourcier, P. Bourgine, N. Nadah, R. Boulet: Chapter 7, A Complex-System Approach: Legal Knowledge, Ontology, Information and Networks, published in: Approches to Legal Ontologies, Springer Press, ISBN 978-94-007-0119-9.

[5] Statistic of the Bundestag, 09.04.2014, p. 5: http://www.bundestag.de/blob/196202/860ee459a5e1d085fd796e376ef3bdd3/kapitel_10_01_statistik_zur_gesetzgebung-data.pdf (retrieved: 06, 2014).

[6] N. Mussett, Federal Ministry of Justice and consumer protection Germany: German Civil Code BGB, Date: 27.07.2011, published in: http://www.gesetze-im-internet.de/englisch_bgb/; Federal Ministry of Justice and consumer protection, Germany: http://www.gesetze-im-internet.de/bgb/xml.zip (retrieved: 06, 2014)

[7] P. Cimiano and J. Völker: Text2Onto A Framework for Ontology Learning and Data-driven Change Discovery, Institute AIFB, University of Karlsruhe.

[8] The Gate Project: https://gate.ac.uk/ (retrieved: 06, 2014).

[9] The OWLExporter Project: http://www.semanticsoftware.info/owlexporter (retrieved: 06, 2014).

[10] The Protégé Project: http://protege.stanford.edu/ (retrieved: 06, 2014).

[11] The OntoLT Project: http://olp.dfki.de/OntoLT/OntoLT.htm

[12] The Stanford POS Tagger: http://nlp.stanford.edu/software/tagger.shtml (retrieved: 06, 2014).

[13] The Stanford Named Entity Recognizer: http://nlp.stanford.edu/software/CRF-NER.shtml (retrieved: 06, 2014).

[14] The Stanford Parser: http://nlp.stanford.edu/software/lexparser.shtml (retrieved: 06, 2014).

[15] The Stanford Dependency Parser: http://nlp.stanford.edu/software/stanford-dependencies.shtml (retrieved: 06, 2014).

[16] The Zurich Dependency Parser for German: http://kitt.cl.uzh.ch/kitt/parzu/ (retrieved: 06, 2014).

[17] GermaNet Homepage: http://www.sfs.uni-tuebingen.de/GermaNet/ (retrieved: 06, 2014).

[18] Cycorp Homepage: http://www.cyc.com/ (retrieved: 06, 2014).

[19] JUNIPER Project: www.juniper-project.org/ (retrieved: 06, 2014).

 ISBN: 978-1-61208-355-1

Semantically Enriched Spreadsheet Tables in Science and Engineering

Jan Top[1,2], Mari Wigham[1] and Hajo Rijgersberg[1]
[1] Wageningen UR, Food and Biobased Research
Wageningen, The Netherlands
{firstname.secondname@wur.nl}

[2] VU University Amsterdam
Amsterdam, The Netherlands

***Abstract*—Tabular data are common in science and engineering. Datasets found in practice are often not very well specified, and are therefore hard to understand and use. Semantic standards are available to express the meaning and context of the data. However, present standards have their limitations in expressing heterogeneous datasets with several types of measurements. Such datasets are abundant in science and engineering. We propose the RDF Record Table vocabulary for semantically modelling tabular data. It complements the existing RDF Data Cube standard. RDF Record Table has a nested structure of records that contain self-describing observations. A first implementation of the model shows that it facilitates finding and integrating data from multiple spreadsheets. This support helps scientists to get the most out of available quantitative data with a minimum of effort.**

Keywords-semantics; table; spreadsheet; e-science.

I. INTRODUCTION

In science and engineering, datasets can be very complex, in particular, if they combine different experiments and observations. We propose a format that has observations and records, rather than traditional tables, as its basic building blocks.

Tabular data are common in science and engineering. Tools to handle such data, such as spreadsheets, are extremely popular because of their flexibility and ease of use. However, this flexibility often leads to data being ambiguous or even incomprehensible, and their provenance being unknown [1][2]. The possibility to immediately proceed to the analysis and visualization of the data, often has a negative effect on the quality of the actual registration in terms of complete and systematic recording. This makes finding, understanding and reusing the data very difficult [3]. As the amount of available data is exploding, it is essential to be able to efficiently locate and reuse existing datasets.

The traditional way to present tabular data is in tables on paper or on a screen. Rows and columns of cells make up their structure. In such a table, an individual recording shows up as a single value in one of the table cells. The associated header cell along the same column or row explains the meaning of this value, for example 'm (kg)' for mass measured in kilograms. In datasets found in practice, this header information is often ambiguous and incomplete. In fact, much of the information about the actual observation is frequently left out. This may even be done on purpose, in order to clean the data for presentation or processing. Tables also become more compact, if all records contain the same quantities, the same unit of measure and have the same interpretation. In this way, the 'bare' numerical or string value in the table cells is separated from the metadata, and directly visible for comparison and available for numerical computation. Researchers are trained in reading such tables and can interpret them immediately.

However, to further exploit datasets in science and engineering, we are not bound to the traditional two-dimensional table format. We can use richer representations to express more contextual information. Many methods have been developed over the last decades to express tabular datasets in a more flexible and rich manner. The W3C RDF (Resource Description Framework) standard provides a more general, graph-based language to do so [4]. RDF Data Cube is a prominent example of such an RDF-based standard [5].

Representing datasets semantically has major advantages. Firstly, the meaning of the measurements is independent of for example the precise text in a spreadsheet, so that data can be found and understood regardless of typos, abbreviations, local terminology and even different languages. Secondly, the use of semantic concepts makes tables machine readable, meaning that they can be (semi-) automatically processed, from simple unit conversion up to complex computations. Finally, allowable numerical values and units can be defined, making it possible to check or clean the data. Moreover, semantic tables can be used as templates for future observations and experiments.

Which requirements should a semantic standard meet to facilitate and stimulate structured annotation of tabular data? First, it should be able to annotate the individual data elements. For example, it should be possible to state that 'the mass of this sample is 2.95 grams', 'the city considered is Amsterdam', or 'this event has occurred 5 minutes and 6.3 seconds later'. Good scientific recordings contain extensive information about each observation, for example on which object it has been measured, by which method and by whom. The annotation (metadata) of the individual data elements explains them and describes their provenance and relations. A standard has to build on existing (domain)

 ISBN: 978-1-61208-355-1

ontologies in order to facilitate shared understanding of the individual observations.

Secondly, a semantic standard for tabular data should make explicit the grouping together of scientific observations that collectively form a 'snapshot' of the world. The observations are combined since they are generated in one experiment, using the same experimental protocol or by a single apparatus. A collection of snapshots, or *records*, is used to detect patterns, similarities or correlations. This grouping is essential for correct interpretation of the data. Within one experiment, the structure of the records is often quite similar. However, when comprehensive recording of all possibly relevant effects is required, datasets can be less homogeneous and well-formed. This, in particular, holds for datasets that combine observations from different origins. Moreover, exact science typically deals with quantities having diverse scales, units and other specifications; values may be missing or occasionally additional measurements are available. Consider for example research that combines input from a number of labs around the world. Some of them have recorded the environmental temperature in degrees Fahrenheit and others in degrees Celsius. One lab has not measured temperature at all. Semantic standards should allow these variations and at the same time provide enough structure.

In this paper, we intend to find a format that is sufficiently rich and flexible to handle complex datasets in science and engineering. In Section II, we first briefly describe existing approaches, in particular the RDF Data Cube vocabulary. This is a recommended W3C standard for multidimensional tables. To be able to handle more heterogeneous datasets, we propose RDF Record Table in Section III, as a supplement to RDF Data Cube. RDF Record Table uses self-contained observations and recursive records. In Section IV, we describe which steps can be taken to cope with the verbosity that is a consequence of the very explicit character of RDF Record Table datasets. This is followed by a description of a first implementation in Microsoft Excel in Section V. Finally, we conclude in Section VI, also listing a number of open issues.

II. RELATED WORK

Many methods take the relational database approach when they convert tables or databases into an RDF-based representation [6]-[8]. They assume that a table consists of a header row defining variables and other rows that contain strings or numbers representing the value of the variable in the same column. In general, they do not support more complex structures. All columns are translated into RDF properties of a single object. At this point, no other metadata is available than what is given in the header and data cells.

A richer format is defined by the RDF Data Cube vocabulary [5], a recommended W3C standard. This vocabulary has been developed in the context of statistical data in social sciences and policy studies, but is also being applied in other areas. Information about the meaning of the data and its provenance is expressed by linking to concepts from other ontologies, most typically the SDMX vocabulary [9]. Data Cube organizes observations as multidimensional datasets. Each observation is a point in n-dimensional space, defined by the associated values of the *dimensions*. Typical dimensions in RDF Data Cube are 'time', 'area' and 'gender'. Each observation contains one or more *measures*, for example 'life expectancy = 83.5 years'. Observations can have *attributes* that provide additional information about them, for example the unit of measure used. A separate section of an RDF Data Cube defines its *structure*; this section can be used as a template for future observations. Another section gives information for external reference to the entire dataset.

In its normalized form, each observation in a data cube contains all its dimensional values. One way to reduce redundancy is by moving shared attributes to the structure definition section. Further reduction can be obtained by introducing 'slices'. A slice is a lower-dimensional representation, which also serves as a proposed interpretation of the dataset. Moreover, one can refer to a slice as an independent entity. Table I shows the example table that RDF Data Cube definition uses to explain the vocabulary [5]. This reference shows the full model of Table I.

TABLE I. LIFE-EXPECTANCY DATA IN DIFFERENT REGIONS OVER TIME

	2004-2006		2005-2007		2006-2008	
	Male	Female	Male	Female	Male	Female
Newport	76.7	80.7	77.1	80.9	77.0	81.5
Cardiff	78.7	83.3	78.6	83.7	78.7	83.4
Monmouthshire	76.6	81.3	76.5	81.5	76.6	81.7
Merthyr Tydfil	75.5	79.1	75.5	79.4	74.9	79.6

The RDF Data Cube vocabulary is very well suited for modelling well-formed, complete datasets such as are produced by statistics offices. Software tools are available to provide useful views of the data. However, these advantages are the result of some restrictions on the data. We submit that these restrictions make the RDF Data Cube less suitable for heterogeneous, multi-scale data such as exist in science and engineering. The requirement to choose *a-priori* between dimensions and measures is problematic in those fields. Rather than assuming some causal order between quantities, we can only state that they have been observed together. For example, for Table I, RDF Data Cube assumes 'sex' (male or female) to be a dimension and 'life expectancy' (values in the table) to be a measure. This assumption is not needed and limits data analysis; it is sufficient to say that 'sex' and 'life expectancy' have been measured simultaneously.

One striking consequence of the hypercube approach is that multiple measures in a single observation are difficult to handle. This is, however, a common experimental setting in science and engineering. For example, imagine that in the above example in addition to 'life expectancy', also the quantities 'weight', 'waste size' and 'length' have been observed. RDF Data Cube has two alternative ways to

 ISBN: 978-1-61208-355-1

handle such a dataset, which cannot be used simultaneously. In the *multiple measures* approach one observation can contain more than one measured quantity. However, all quantities must have the same attributes, for example, the same type and unit of measure. This rules out this approach for most exact science applications. The second approach restricts observations to having a single measured value. It allows a dataset to carry multiple measures by adding an extra dimension, a measure dimension. This turns a measured value into a kind of semi-dimension. We submit that this construction complicates the model unnecessarily and may influence the interpretation of the data.

Another characteristic of RDF DataCube is that it makes extensive use of properties (rather than classes) as its main organizing mechanism. The design introduces many different types of properties. It is questionable whether these different properties are needed to express the meaning of the data. They make the design of a model rather complex.

RDF Data Cube is intended for describing 'well-formed' datasets. As a result, several constraints are placed on the data, for example that each observation must have a value for every measure. For example, if for one measurement in the example it is not known whether this person is a man or a woman, this data point cannot be included in the model. Another assumption is that the multidimensional structure is a regular (hyper)cube, not permitting rows with varying length for a single dimension. If we know the standard deviation of the life expectancy value for Cardiff and a few other regions, we cannot add this to the above in Table I. Another complication would arise if some life expectancy values were expressed in years-with-decimal (as in the table), and others in years-and-months.

Whereas RDF Data Cube and other standards define the structure and context of tabular data, they are not intended for expressing provenance of data on the web. For that purpose, additional vocabularies have been developed. The W3C-standard PROV is becoming increasingly popular for this purpose [10]. It describes the origins of any type of data, helping the user to evaluate how appropriate and trustworthy the data is for a particular use. PROV basically says that a `prov:Agent` performs a `prov:Activity`, in which he uses or generates a `prov:Entity`. Tables, records, slices and individual measurements can all be seen as subclasses of `prov:Entity`. The previously defined Dublin Core Terms [13] vocabulary complements the PROV model with detailed concepts about publications and authorship.

III. RDF RECORD TABLE

Experience with researchers over the past ten years has confronted us with many different datasets. Many of them are contained in spreadsheets and data analysis tools such as Matlab [11] and SPSS [12]. Our work on introducing electronic lab notebooks in the multidisciplinary domain of food science has revealed many issues in data recording in the lab. Annotation of the data is often scarce and ambiguous due to the focus of researchers on the research itself rather than its bookkeeping. In addition, large amounts of data are produced by automated measurement equipment in the lab. These devices tend to produce more systematic metadata, but linking data from different sources is as yet difficult and labor intensive. Initially, we proposed templates to stimulate systematic annotation of research data, but experience has shown that this restricts the creative and essentially unstructured character of scientific research. Moreover, researchers are typically reluctant to spend a lot of time on data bookkeeping. Inspired by other initiatives to annotate datasets using RDF, we have devised an approach that can work in the tools commonly used by researchers and at the same time support rich annotation. This approach has developed into a model for tabular data called RDF Record Table.

The RDF Record Table vocabulary is intended for recording original and processed data in science and engineering. It models datasets in terms of observations and records (see Fig. 1, using `rec:` as a prefix for the RDF Record Table namespace). An *observation* is a statement about an entity or the property of an entity, such as 'the temperature of this object measured by a pt-sensor is 36.5C' or 'this milk sample is from batch 20140612YTU'. A *record* combines observations to form a snapshot, thus conveying the assumption that in some way the observations are related - in time, location, subject, conditions, or in another way.

To express composite structures, in RDF Record Table any record can recursively contain sub-records, which again are of the type RDF Record Table. For example, an experiment may observe multiple samples at one fixed temperature. For each sample its viscosity, composition and mass are measured over time. This means that the entire dataset consists of a RecordTable that at its highest level contains (i) the observed temperature and (ii) a sub-record for each sample. Each sub-record in turn contains the sample identifier and sub-records that describe viscosity, composition and mass for that sample measured at a point in time. In the most explicit form, all sub-records are expanded into non-nested records. In this example, the top level RecordTable only contains sub-records, each of them stating the observed temperature, time point, sample id and the other measured properties.

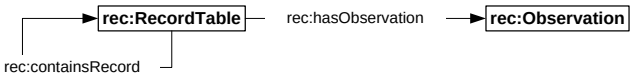


Figure 1. Basic RDF Record Table schema

In Turtle format RDF Record Table is defined as follows.

```
rec:RecordTable
      a  rdfs:Class ;
      rdfs:subClassOf prov:Entity .

rec:hasObserved
      a   owl:ObjectProperty ;
      rdfs:domain rec:RecordTable ;
      rdfs:range rec:Observation .
```

 ISBN: 978-1-61208-355-1

```
rec:containsRecord
      a  owl:ObjectProperty ;
      rdfs:domain rec:RecordTable ;
      rdfs:range rec:RecordTable .

rec:Observation
      a  owl:Class ;
      rdfs:subClassOf prov:Entity .
```

In practice, we see that two types of observations frequently occur, i.e., *identified entities* and *properties measured on a scale*. Examples of *identified entities* are 'sample XY876b', 'Newport' and 'Peter'. Quantities such as 'length', 'mass', and 'temperature' are examples of properties measured on a scale. These two types extend the basic schema by subclassing `rec:Observation`, as shown in Fig. 2.

In traditional tables, identified entities are typically represented by a unique, human readable identifier as a value, and a type indication in the associated header cell. RDF Record Table uses externally available domain ontologies to express all that is needed to know about such an entity by pointing to the relevant instance. In Table I, besides 'life expectancy' also 'periods', such as 2004-2006, can be considered as identified entities since they are not supposed to be read as numerical values.

For the other type of observation, a *property measured on a scale*, RDF Record Table uses ontologies that define quantitative or qualitative values defined on a scale, possibly with units of measure. In Table I, 'sex' and 'life expectancy' are typical measured properties, one on a nominal scale and the other on a rational scale, with unit 'Year'. In our work we use OM (Ontology of units of Measure and related concepts) [14] for expressing quantitative measurements. OM contains a large number of quantities and units of measure suited to scientific and engineering datasets. It also provides the necessary properties for linking the quantities, domain concepts and units. However, other ontologies such as QUDT [15] and SDMX [9] can be used equally well. The measured quantities can be properties of the observed entities, but do not need to be related to anything specific. For example, in Table I, the life expectancy measured is that of the associated geographical region. On the other hand, 'time' is usually not connected to a specific entity (except for example to a 'time zone').

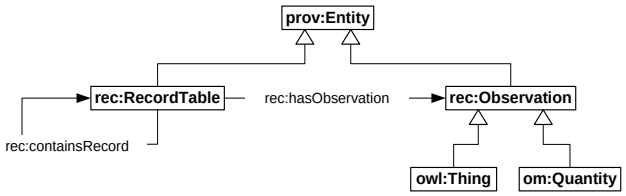


Figure 2. RDF Record Table expressing domain and provenance information

Finally, by making `rec:RecordTable` and `rec:Observation` subclasses of `prov:Entity` we ensure that all provenance information can be expressed for individual measurements and for records.

To illustrate the use of the RDF Record Table format, we show how the cells with values 76.7 and 83.3 in Table I are modelled. We see that the first level of nesting defines four records (:o1, :o2, :o3, :o4), one for each region. We use the ontology for geographic areas (as *identified entities*) that was also used in the RDF Data Cube example [5]. The next level specifies the three time periods, again using instances that were also used in the data cube example. At the third level of sub-records, we register two properties measured on a scale, viz. 'sex' and 'life expectancy'. For indicating the variable 'sex', we use an sdmx-code, as in that data cube; to illustrate the use of OM [14], we use the concept `om:Duration` from that ontology to describe 'life expectancy'. The value of a quantity in OM is of the type `om:Measure`, which is a combination of a numerical value and a unit.

```
:dataset1  a  rec:RecordTable ;
  rec:containsRecord :o1 , :o2 , :o3 , :o4 .

:o1  a   rec:RecordTable ;
   rec:hasObserved ex-geo:newport_00pr ;
   rec:containsRecord :o11 , :o12 , :o13 .

:o11  a   rec:RecordTable ;
   rec:hasObserved
<http://reference.data.gov.uk/id/gregorian-
interval/2004-01-01T00:00:00/P3Y> ;
   rec:containsRecord :o111 , :o112 .

:o111  a  rec:RecordTable ;
   rec:hasObserved sdmx-code:sex-M ,
        :lifeExpectancy_76_7YR .

:lifeExpectancy_76_7YR a om:Duration ;
   om:value :_76_7YR .

:_76_7YR  a  om:Measure ;
   om:numerical_value "76.7"^^xsd:string ;
   om:unit_of_measure_or_measurement_scale om:year
.

...

:o2  a   rec:RecordTable ;
   rec:hasObserved ex-geo:cardiff_00pt  ;
   rec:containsRecord :o21 , :o22 , :o23 .

:o21  a   rec:RecordTable ;
   rec:hasObserved
<http://reference.data.gov.uk/id/gregorian-
interval/2004-01-01T00:00:00/P3Y> ;
   rec:containsRecord :o211 , :o212 .

...

:o212  a  rec:RecordTable ;
   rec:hasObserved sdmx-code:sex-F ,
   :lifeExpectancy_83_3YR .
```

 ISBN: 978-1-61208-355-1

```
:lifeExpectancy_83_3YR a om:Duration ;
   om:value :_ 83_3YR .

:_83_3YR  a  om:Measure ;
   om:numerical_value "83.3"^^xsd:string ;
   om:unit_of_measure_or_measurement_scale  om:year
.
```

We now discuss a number of differences between RDF Record Table and RDF Data Cube. The most salient difference between RDF Data Cube and OQR Record Table is the fact that RDF Data Cube sees complex datasets as n-dimensional hypercubes, whereas RDF Record Tables are defined recursively via nesting. The second major distinction between the two approaches is that RDF Data Cube distinguishes between dimensions and measures, whereas OQR Record Table does not make a priori assumptions about the roles of individual observations. We consider making such decisions to be the task of the data analyst. Moreover, RDF Record Table has no centralized section describing the structure of the table. If it is necessary to prescribe an observation protocol or template, it suffices to list the identified entities and properties measured as the items to register in each record. Finally, where the RDF Data Cube definition makes intensive use of properties, RDF Record Table only has a few simple properties and further builds on concepts from dedicated, external ontologies.

RDF Data Cube does not allow missing variable-values or an occasional extra measurement. In contrast, in RDF Record Table any record can contain an arbitrary set of measurements, with different types and sub-records. Missing values or varying units of measure or other attributes within a single dataset are no problem. We do not demand completeness or regularity of the data, in the sense that a record can contain any set of entities and properties. This better reflects the reality of datasets in science and engineering, in particular, when datasets from different sources are combined. It can be argued that such datasets can be modelled in RDF DataCube simply by violating the integrity constraints. This is, however, a bad approach to using a standard, and can lead to interoperability problems between tools developed for the standard.

For example, in Table I we can add 'the measured average weight of the inhabitants of this region' to an existing observation using the OM quantity `om:Mass`. We can also switch to 'life expectancy' measured in months rather than years for this single observation. This is shown here:

```
:o431  a  rec:RecordTable ;
   rec:hasObserved sdmx-code:sex-M ,
   :lifeExpectancy_74_9MONTH ,
   averageWeight_71kg ;

:lifeExpectancy_74_9MONTH a om:Duration ;
   om:value :_ 74_9MONTH .

:_74_9MONTH  a  om:Measure ;
   om:numerical_value "74.9"^^xsd:string ;
   om:unit_of_measure_or_measurement_scale
om:month .

:averageWeight_71kg a om:Mass ;
   om:value :_71kg.

:_71kg  a  om:Measure ;
   om:numerical_value "71"^^xsd:string ;
   om:unit_of_measure_or_measurement_scale
   om:kilogram .
```

We conclude that RDF Record Table can be viewed as a generalized RDF Data Cube, making fewer assumptions about the regularity and completeness of the data. It can act as a precursor in the data cleaning, analysis and integration process. If a dataset that was originally drafted as an RDF Record Table meets certain requirements, it is in principle possible to automatically transform it into an RDF Data Cube. Any dataset expressed in RDF Data Cube, on the other hand, can be modeled as RDF Record Table.

IV. REDUCING REDUNDANCY IN RDF RECORD TABLE

In RDF Record Tables, the individual observations are in principle self-contained, allowing an extremely flexible approach. However, making all metadata available for each observation in practice leads to very large data files. In a single experiment, records are often very similar and much information is redundant. This means that many details can be referred to rather than repeated. In the traditional table, metadata is typically condensed in the header row, assuming that the reader knows that it holds for all rows. In an RDF-based graph model, we can be more flexible. We can use any completely specified value as a template for other observations. It is then possible, using for example SPARQL [16], to generate the full, extensive description from the reduced version when needed. This is in particular effective if the expansion to the fully explicit (normalized) form can be done locally, i.e., only for the interesting parts of a table.

Fig. 3 shows how RDF Record Table supports compression of datasets by giving metadata information by referring to a similar measurement. Each `rec:Observation` can hold a literal value (the string or number ending up in a table cell) and emulate another observation, which has identical attributes other than the value. These referencing observations are collected in records, just like normal observations.

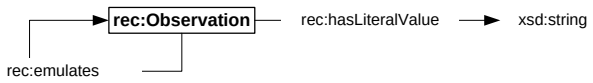


Figure 3. Describing an observation by reference.

In Turtle format, the definition is as follows.

```
rec:emulates
      a   owl:ObjectProperty ;
      rdfs:domain rec:Observation ;
      rdfs:range rec:Observation .

rec:hasLiteralValue
      a    owl:DatatypeProperty ;
```

 ISBN: 978-1-61208-355-1

```
      rdfs:domain rec:Observation ;
      rdfs:range xsd:string .
```

For example, the observation from Table I that in Monmouthshire the life expectancy of women in the period 2006-2008 was 81.7 years, is originally expressed in

```
:o332  a  rec:RecordTable ;
   rec:hasObserved sdmx-code:sex-F ,
   :lifeExpectancy_81_7YR .
```

as

```
:lifeExpectancy_81_7YR a om:Duration ;
   om:value:_81_7YR .

:_81_7YR  a  om:Measure ;
   om:numerical_value "81.7"^^xsd:string ;
   om:unit_of_measure_or_measurement_scale  om:year
.
```

Using the fact that all details for `:lifeExpectancy_81_7YR` are the same as for `:lifeExpectancy_76_7YR` from observation `:0111`, except for the actual value, we can summarize this as

```
:lifeExpectancy_81_7YR a rec:Observation ;
   rec:emulates :lifeExpectancy_76_7YR ;
   rec:hasLiteralValue "81.7"^^xsd:string.
```

For this example, this may not seem an impressive compression. However, if more metadata is included, such as descriptions, the devices used, methods applied and other background information, the reduction of the size will be substantial. This, in particular, holds for datasets with large numbers of similar measurements. Finally, further reduction of datasets is possible by applying general compression algorithms [17].

V. IMPLEMENTATION EXAMPLE

A good model of tabular data is useless if the data can't easily be input. Given the popularity of the classic table format in tools such as spreadsheets, it should be possible to use these for data entry and then construct semantic datasets from there. In order to make this process as easy as possible, it should fit into existing work procedures and tools and minimize additional effort by the user. Since Microsoft Excel is extremely popular, we have implemented the RDF Record Table model as an add-in for this tool, called Rosanne [14]. Rosanne supports engineers and scientists in creating semantic tables (as yet simple tables, i.e., rectangular with one header row or column). Similar functionality for the RDF Data Cube has been implemented in TabLinker [18]; however this is a standalone tool which cannot be accessed from within Excel. Rosanne allows users to enter their data in a simple table format. Rosanne then uses OM (Ontology of units of Measure and related concepts) [14] to assist users in adding relevant quantities and units of measure to the table. In addition, other domain-specific ontologies are available for annotating identified entities in the table, such as samples, objects, locations, etc.

The user is not confronted with the Record Table model nor do they have to have any knowledge of ontologies. The user selects the concepts they want from dropdown lists showing the user-friendly labels from the ontologies. The URIs (Uniform Resource Identifiers) for the ontology concepts are stored in the Record Table model by the add-in. The add-in can also automatically annotate existing data with units and quantities from OM, based on heuristics [19]. This does not always produce accurate results, but saves time for the user by creating an initial annotation which can be corrected where necessary. Finally, Rosanne allows users to search for annotated tables and integrate them.

Fig. 4 shows an example from food science. In this experiment, the researcher wishes to combine rheological measurements on protein samples with sample composition data. Without semantic support, this task would require her to find the relevant files somehow, then to copy and paste different data by hand, with plenty of scope for error. With Rosanne, she can find the files easily via the search function. The table has been annotated using OM and a domain ontology. She then selects 'Protein' as the identifier, and 'Storage Modulus' and 'Composition' as the variables of interest. Rosanne creates a query to find the relevant data, and generates the integrated table.

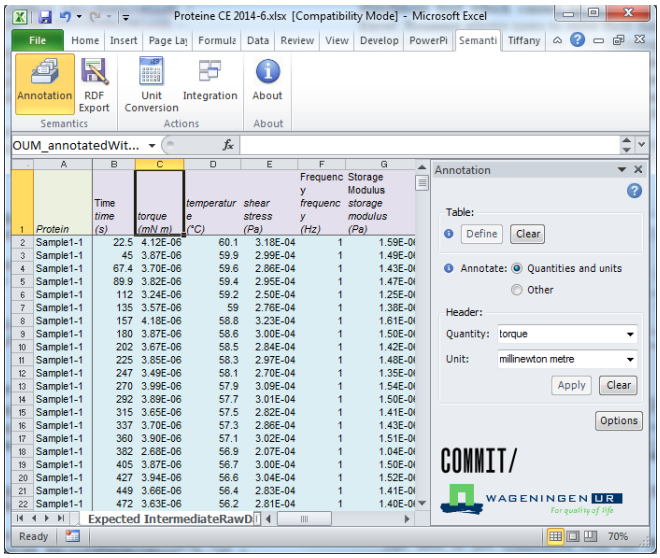


Figure 4. Rosanne using RDF Record Table.

VI. CONCLUSION AND FUTURE WORK

Looking to the future, semantic datasets are a step towards advanced quantitative e-science. The data can be documented and linked to the scientific process, assisting the researcher and ultimately leading to full transparency and reusability of quantitative scientific knowledge.

In practice, this means that data entry tools can be developed which use ontologies to support the user in

 ISBN: 978-1-61208-355-1

adding contextual information. Describing the content and structure of tabular data semantically makes it possible to easily find data even in disparate sources, to understand and clean the data and to combine it semi-automatically. This way, much richer datasets will be published in the future, so that others can fully understand them and build further on them.

We have proposed RDF Record Table as a way to organize observational data semantically. The model complements the RDF Data Cube vocabulary. RDF Data Cube offers the benefits of semantic modelling to domains such as statistics, with regular, standardized datasets. RDF Record Table offers more flexibility in storing heterogeneous data, and therefore extends those same benefits to the more complex world of science and engineering. A first implementation of the RDF Record Table model in Microsoft Excel, called Rosanne, demonstrates the benefits of semantic tables. This includes semi-automatic integration of datasets. This functionality is presently being evaluated by a number of R&D organizations of multinationals in food production, cooperating in TI Food and Nutrition [20]. In another area, we are using RDF Record Table for statistical analysis with the popular language R.

For full implementation of this model, several issues must still be solved. We mentioned the automatic (local) expansion and compression of datasets, mapping to and from RDF Data Cube, and the translation to and from two-dimensional representations. In addition to these, the recovery of legacy data needs attention. There is a wealth of data stored in existing spreadsheets, which have, in general, an informal structure and no annotations. Current results for fully automatic annotation are still of insufficient quality [19], so more research is needed to find how to unlock this legacy data. We plan to submit RDF Record Table to the CSV on the Web Working Group [21] for consideration and inspiration in their work to provide better interoperability for tabular data.

ACKNOWLEDGMENT

This publication was supported by the Dutch national program COMMIT.

REFERENCES

[1] Y. L. Simmhan, B. Plale, and D. Gannon. ‘A survey of data provenance in e-science.’ ACM SIGMOD Record, 2005. doi:10.1145/1084805.1084812

[2] A. Garcia, O. Giraldo, and J. Garcia, ‘Annotating Experimental Records Using Ontologies.’ Int. Conference on Biomedical Ontology, Buffalo, NY, USA, 2011. Available from: http://ceur-ws.org/Vol-833/paper12.pdf. Retrieved June, 2014.

[3] J. Gray, ‘Jim Gray on eScience: a transformed scientific method.’ in T. Hey, S. Tansley, K. Tolle (Eds.), The Fourth Paradigm: Data-Intensive Scientific Discovery, Microsoft Research, 2009, pp. xvii–xxxi.

[4] Semantic Web, W3C. Available from: http://www.w3.org/standards/semanticweb/. Retrieved: June, 2014.

[5] R. Cyganiak, D. Reynolds, (eds). RDF Data Cube Vocabulary, W3C, 2012. Available from: http://www.w3.org/TR/vocab-data-cube/. Retrieved June, 2014.

[6] L. Han, T. Finin, C. Parr, J. Sachs, and A. Joshi, ‘RDF123: From spreadsheets to RDF.’ Lecture Notes in Computer Science, Vol. 5318 LNCS, 2008, pp. 451–466. doi:10.1007/978-3-540-88564-1-29

[7] J. Cunha, J. Saraiva, and J. Visser, ‘From spreadsheets to relational databases and back.’ In Proceedings of the 2009 ACM SIGPLAN workshop on Partial evaluation and program manipulation - PEPM ’09 (p.179), 2009.

[8] C. Bizer, and R. Cyganiak, ‘D2R Server – Publishing Relational Databases on the Semantic Web.’, World, p. 26, 2006.

[9] S. Capadisli, S. Auer and A.-C. Ngonga Ngomo, ‘Linked SDMX Data’. Semantic Web, 2013. doi:10.3233/SW-130123

[10] P. Groth, L. Moreau. (eds), PROV Overview, W3C, 2013. Available from: http://www.w3.org/TR/2013/NOTE-prov-overview-20130430/. Retrieved June, 2014.

[11] Matlab, The Language of Technical Computing. Available from: http://www.mathworks.nl/products/matlab/. Retrieved July, 2014.

[12] SPSS Statistics. Available from: http://en.wikipedia.org/wiki/SPSS. Retrieved July, 2014.

[13] M. Nilsson, A. Powell, P. Johnston, and A. Naeve. ‘Expressing Dublin Core metadata using the Resource Description Framework (RDF).’, 2008. Available from: http://dublincore.org/documents/dc-rdf . Retrieved June, 2014.

[14] H. Rijgersberg, M. Wigham, and J. L. Top, ‘How semantics can improve engineering processes: A case of units of measure and quantities.’ Advanced Engineering Informatics, *25*(2), 2010, pp.276–287. doi:http://dx.doi.org/10.1016/j.aei.2010.07.008

[15] R. Hodgson, P. J. Keller, J. Hodges, and J. Spivak, ‘QUDT - Quantities, Units, Dimensions and Data Types Ontologies’. Available from: http://qudt.org/. Retrieved June, 2014.

[16] W3C, SPARQL Query Language for RDF. Available from: http://www.w3.org/TR/rdf-sparql-query/. Retrieved July, 2014.

[17] J. Urbani, J. Maassen, N. Drost, F. Seinstra, F., and H. Bal, ‘Scalable RDF data compression with MapReduce.’ Concurrency Computation Practice and Experience. Vol. 25, pp. 24–39, 2013. doi:10.1002/cpe.2840

[18] TabLinker, 2012. Available from: http://www.data2semantics.org/2012/02/19/tablinker/. Retrieved June, 2014.

[19] M. van Assem, H. Rijgersberg, M. Wigham, and J.L Top, ‘Converting and annotating quantitative data tables’. The Semantic Web - ISWC 2010, vol. 6496/2010, 2010, pp. 16–31. doi:10.1007/978-3-642-17746-0_2.

[20] TI Food and Nutrition. Available from: http://www.tifn.nl. Retrieved July, 2014.

[21] CSV on the Web Working Group Charter, 2013. Available from: http://www.w3.org/2013/05/lcsv-charter.html. Retrieved June, 2014.

 ISBN: 978-1-61208-355-1

An Ontology-Based Framework for Semantic Data Preprocessing Aimed at Human Activity Recognition

Rosario Culmone, Marco Falcioni, Michela Quadrini
Computer Science Division, School of Sciences and Technologies, University of Camerino
Email: {firstname.lastname}@unicam.it

***Abstract*—Over the last few years, complex systems which collect data from a considerable number of sources are increasing. However, it is not always possible to have a clear overall view of the information contained within data, due to both their granularity and to their wide amount. Since an analysis procedure able to take into account the semantics of records is often needed, ontologies are becoming widely used to describe the domain and to enrich the acquired data with its significance. In this paper, we propose an ontology-based methodology aiming to perform semantic queries on a data repository, whose records originate from a network of heterogeneous sources. The main goal of such queries is the pattern matching process, i.e., recognition of specific temporal sequences in fine-grained data. In our framework, benefits deriving from the implementation of a domain ontology are exploited in different levels of abstraction. Thereafter, reasoning techniques represent a preprocessing method to prepare data for the final temporal analysis. Our proposed approach will be applied to the ongoing AALISABETH, an Ambient Assisted Living project aimed to discover and manage the behaviour of monitored users.**

Keywords - Ontology; Semantic Reasoning; Complex Event Processing.

I. INTRODUCTION

In complex data-acquisition environments, the storing of data as well as the information carried by such records become more and more important. When data are generated by many heterogeneous sources, it turns out to be important both the integration of information and the interoperability of applications that process the data. Usually, these records are collected in a data repository and it sometimes results difficult to have a clear view of the whole acquired information. Therefore, it could be even more hardly to proceed with an analysis which do take into account the semantics of data. For this reason, ontologies are becoming more and more utilized to address this issue, because they are able to describe instances of a real-world system.

An example of the described situation could be represented by an Ambient Assisted Living (AAL) context. In such domestic environment a wide network of smart objects is installed, whose task is to provide the possibility to monitor the user lifestyle. In order to reach this aim, the Smart Home (SH) relies on many different types of objects: from clinical devices for the user health to indicators of presence, from temperature and humidity measurements to fridge and door opening sensors. Considering that the storing data repository, usually a Database (DB), often shows a lack of semantic information and relationships among the smart home components, acquired data from smart objects need to be treated according to their significance. Hence, data processing cannot prescind from the implementation of a domain ontology, whose primary scope is to entirely describe actors belonging to the smart home, i.e., user, smart objects and their relationships. Thereby, data can be treated according to their semantic, which is formalized in the domain ontology. Subsequently, the same ontology can be enriched by rules for a further analysis phase of the system. In fact, it can happen that several concepts are known, but they are not yet present in the data repository nor in the ontology. If such knowledge is needed for the successive phase of analysis, it can be introduced in the ontology.

In this paper, a framework capable to address the illustrated context is presented. The described methodology, in addition to pattern discovery techniques, has been developed to answer to the requirements of an Ambient Assisted Living project. The ongoing Ambient-Aware LIfeStyle tutoring for A BETter Health (AALISABETH) project aims to analyse the user's lifestyle by means of a non pervasive sensor network, which can monitor and detect well-specific daily activities. In particular, the main goal of this project is to detect a set of abnormal behaviours that could eventually bring to an onset of the most common diseases. In the present paper, we intend to discuss a novel methodology that consists of comparing the observed activities to those formalized in the ontology. Hence, the final task of the framework is to determine whether the prearranged patterns are matched, and thereafter communicate such results to caregivers.

This paper is structured as follows: Section II examines the related literature concerning the topics addressed in this work. Section III firstly explains the motivation of the proposed methodology, then provides a detailed description of the framework architecture and lists the tools used to implement each component. Finally, Section IV illustrates the work in progress and the nearly future development of the ongoing project.

II. RELATED WORK

The approach presented here includes different areas of research: ontology-based description of a domain, mapping a Database to an existing ontology and enrich external data with their significance, semantic data preprocessing, pattern matching and identification in a sequence of data.

Ontologies are commonly used to explicitly formalize and specify a domain of knowledge [1]. Furthermore, they improve the automation of integration of heterogeneous data sources, also providing a formal specification of the vocabularies of concepts and the relationships among them [2]. Many are

 ISBN: 978-1-61208-355-1

the publications in which ontologies are employed to achieve information integration over various domains. An example for Intelligent Environments are found in [3] [4] [5] , where an ontology is essentially implemented for both formally expressing the domotic environment (e.g., sensors, gateways and network) and providing reasoning mechanisms. This reasoning allows to support automatic recognition of device instances and to verify the formal correctness of the model. Further works presenting ontologies finalized to AAL activities are Mocholi et al. [6] and Fleury et al. [7].

Techniques of mapping an external Database to a local ontology are suggested by Sedighi and Javidan [8] and Barrasa et al. [9]. Also, tools that automatically generate OWL ontologies [10] from database schemas have been presented, for instance by Cullot et al. [11] and Rodriguez-Muro et al. [12].

Ontologies may also support a semantic approach to applications involving Business Process Management (BPM) techniques and analyses of processes based on a list of recorded events, i.e., Process Mining. In this case, a possible procedure is to enrich the event logs coming from external data sources by using ontology based data integration, as observed by Tran Thi Kim and Werthner [13]. Furthermore, a similar methodology used to integrate semantic annotation to the event log is illustrated in a BPM context by Ferreira and Thom [14], where semantic reasoning is used to automatically discover patterns from the recorded data.

Since in [14] only sequences of determined data are relevant, time constraints among events may not be strictly taken into account. Considering the temporal nature of activities as a succession of actions admits several feasible approaches, such as the probabilistic [15] and the statistical one [7].

Instead, in the field of activity recognition, time interval restrictions become essential. Cases of dealing with complex events are rapidly increasing. To address this issue, ontologies are used as a basis to preserve information and relationships among events. Thereafter, they are temporally managed by a Complex Event Processor (CEP), yielding to a semantic complex event processing technique [16].

III. METHODOLOGY

A. Motivation

Our proposed methodology originates from the necessity to deal with the significance of a wide amount of heterogeneous data, which are commonly stored in a data repository. Since the beginning of the entire procedure, the final goal of the analysis is well-known, as well as a detailed awareness of the whole system and records thereby acquired. Furthermore, one should focus not only on the single values of data, rather than on its meaning within the context. In order to take into account such relationships and formalize the knowledge of the whole context, the implementation of an ontology results to be actually mandatory. The general approach can be illustrated by Figure 1. On top, the real-world system is composed of both static knowledge and data generated by the considered system. As the former is fixed, the user is allowed to directly transfer his domain knowledge into the corresponding ontology. On the other hand, the latter produces a stream of data which is collected by the repository. Since in this step data are usually registered as a list of records, they show a fine-grained nature, carrying generally their value, originating device, data type, timestamp, and so on. In a similar context, the granularity features of acquired data are a stumbling block for the contained semantic information, which may be eventually lost. Also, a further verifiable aspect is data redundancy; that is, there can be several devices which apparently output different results, but they provide the same information. Hence, the ontology is introduced to somehow circumvent such technical aspects and to form a bridge from the real-world system and its formal representation. In fact, it is able to merge the static knowledge and the dynamic parts by means of classes and their instances, rebuilding the whole context. Therefore, the advantages of a semantic technique are exploited twice. Once the ontology-based method has provided a conceptualization and specific description of the real-world system, such formalization drives the analysis phase. In our specific case, it is needed to look for well-determined set of data. It is worth noting that such research has to be performed according to the own semantics of the desired set. This requirement represent the main reason why an ontology-based technique is introduced.

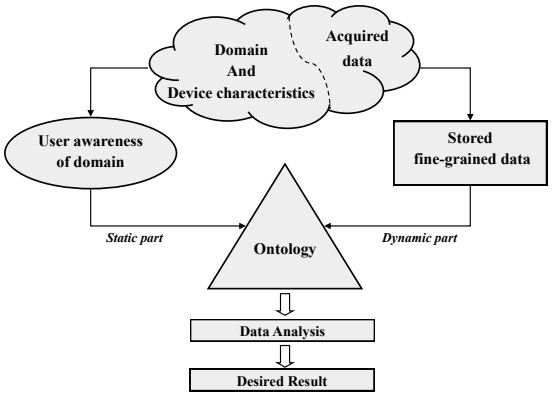


Figure 1. General approach: from real world to ontology

B. Architecture of the framework

In order to address the presented situation, we propose the framework depicted in Figure 2. One of the most common methods to collect data from a network of heterogeneous sources is to store them in a DB. Therefore, our first **necessity** is the possibility to somehow find a correspondence between the elements of a DB and the ones of the previously implemented ontology. Such a semantic model is built following a precise structure, as described in detail later. Once data are reorganized according to their meaning, the ontology plays a preprocessing role. In fact, the user can express semantic queries in order to extract from the ontology a well-specific aspect of the entire environment. It is worth noting that **some** particular views could not be previously retrieved from the fine-grained nature of the data stored in the DB. These different views may be considered as the output of sensors which are not physically present in the system, and we can label them as *virtual* sensors.

As far as time constraints are not taken into account, an ontology is sufficient to classify and organize data produced

 ISBN: 978-1-61208-355-1

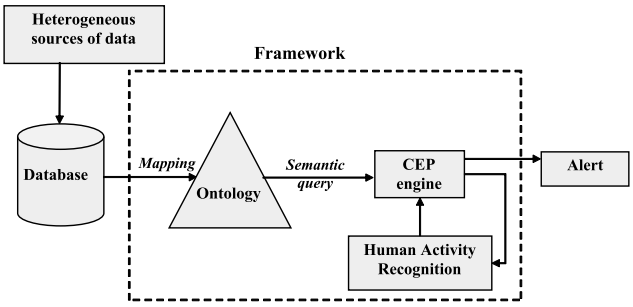


Figure 2. Multi step methodology of the framework

from both physical and virtual sensors. However, since our final aim is to obtain a specific time-dependent output, we need to introduce in our framework a component able to manage these time restrictions. This issue is solved by the use of a Complex Event Processing (CEP) engine, that is, a technique concerned with timely detection of compound events within streams of simple events [17].

C. Ontology structure

In our proposed framework, the main element is represented by the ontology that clearly defines the semantics of the considered domain and is used as a shared knowledge base for all the related components.

This ontology, called OntoAALISABETH, has a particular approach, as illustrated in Figure 3. Four main domain ontology systems - User, Environment, Activity and Device - represent the knowledge base in AAL context. User describes the concepts related to user's profile, while Activity describes several domestic activities that are necessary to detect abnormal behaviour. These two parts play the central role. Consequently, the appliances within the AAL environment should adapt to the user, and not vice versa. Then, Environment and Device describe user's house and the sensors network installed.
Furthermore, this ontology shows different abstraction layers

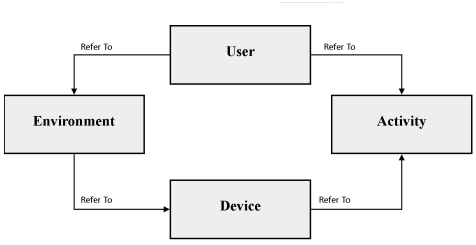


Figure 3. Context ontology overview.

that composed together form a pyramid-like structure, where each lower level specialises the one on the next upper layer.

The architecture, as reported in Figure 4, is realized by the following main components:

- A static layer (domain and domain-specific ontology);
- A dynamic layer (data and view ontology).

Each part of our ontology plays a specific role in order to respond to different requirements of the project, as described below.

1) Domain ontology: Initially, an upper domain ontology is built. One should note that this higher level of abstraction can be considered as a ready-to-use ontology for any other analogue domain. In other words, it consists of an ontology which generally formalizes concepts present in some context, and is thought to be commonly valid. In fact, concepts are described as much generally as possible, carrying *static* information. Since our instance is an AAL context, as the literature suggests, we implemented a domain ontology extending and reusing an existing one. In our case, the starting ontology has been chosen to be DogOnt [3]. It has been built in a smart home context, but does not take into account several elements of an AAL environment. Therefore, we have formalized classes and relationships about the SH, its architecture and furniture, the presence and activities of one or more users, the introduction of smart objects with a communication network, sensors and clinical devices, and so on.

2) Domain-specific ontology: This first middle layer places below the previous upper ontology, extends several static properties and focuses on the structure of the considered domain. In our domain-specific ontology, we formalize the various components belonging to the home environment: the real structure of the ambient and disposition of rooms, the personal information about who lives in the house, which sensors are installed in the network and how they communicate. Also, the complete knowledge of the domain allows the developer to add new elements and relationships in the ontology, which cannot be described in the technology of data storing.

3) Data ontology: The data ontology extends the previous domain-specific layer introducing the concept that each device generates fine-grained data. In this level, the described classes are instantiated with individuals that present a one-to-one correspondence with each record stored in the DB. This procedure is allowed by a technique known as Ontology-Based Data Access (OBDA) approach [12]. It consists of a *mapping* that associate data from the data sources with concepts in the ontology. In particular, by means of suitable SQL queries over the DB one extracts records and propagates them into concepts. Hence, the whole data ontology is implemented taking into account the sensor network, formalized in the previous layer, and is continuously updated. In this step, the semantic information about the fine-grained data is partially recovered, but the following layer permits to have custom specific views of the system.

4) View ontology: In our system, data are generated by the non pervasive network which is installed to monitor user lifestyle. In particular, such records may assume different meanings depending on the specific context. For instance, if a presence in the bedroom is followed by one in the kitchen, it has a different meaning from the same followed by one in the bathroom. Since a particular record deserves different semantic treatments, the view ontology takes into account such various circumstances. More frequently, one must evaluate the presence in the bedroom from different points of view. In terms of an ontology, this necessity converts to the implementation of

 ISBN: 978-1-61208-355-1

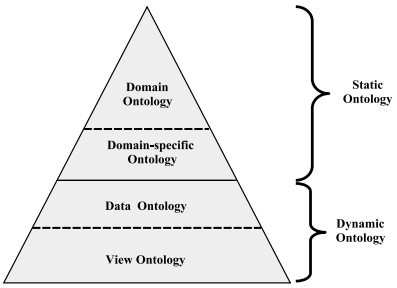


Figure 4. Pyramid-like structure of the ontology

new view classes where individuals are inferred. So, alternative views provided by this lower layer are needed in order to reorganize instances of data ontology. These views are defined by the expression of several equivalent classes. They are driven by the main scope to classify instances having well-determined properties and relationships; that is, these classes are populated by the desired individuals and carry the same knowledge replicated several times. The whole process of reorganization is allowed by the use of the reasoning tools, which represents the formal basis for the expressive strength of OWL. In fact, through this instrument, one can obtain additional statements that are inferred from the facts and axioms previously asserted. This reviewing step is the grounding of the *preprocessing* procedure. Thereafter, the reasoning tool allows to perform semantic queries on the ontology and extract the desired information for the following effective analysis, as reported in Figure 2. One should note that querying the ontology in this final step of the proposed methodology corresponds to select an amount of data generated by virtual sensors, i.e., a group of data following the user *interpretation* of the system. Moreover, this approach developed by means of inference classes has the important advantage to be extensible and additive.

In order to better explain the advantages deriving from the classification of the view ontology, let us consider the following cases. One of the most relevant aspects for our project is monitoring if the user gets up during the night for eating or toileting. In order to recognize these activities, we proceed creating two views, i.e., macro ontology classes. Each class contains all inferred individuals that allow the eventual recognition of the considered activity. In this particular case, the information about getting up and exiting from the bedroom are common. Instead, presence and utilization of the toilet is found in the first case, while presence in the kitchen and opening a sideboard or refrigerator belong to the second view. Furthermore, in both cases we require that the person comes back to the bedroom after some time and continues to sleep. Hence, these sets of individuals populating the view classes are selected as input for the following step of analysis. It is worth noting that processing data with the described technique allows to preserve relationships and constraints introduced by the previous domain-specific layers of the ontology. Contents of each layer of the pyramid-like structure are shown in Figure 5.

D. Process analysis

The first component of the framework previously described employs traditional Semantic Web (SW) techniques, e.g., query languages and automated reasoning. However, for a

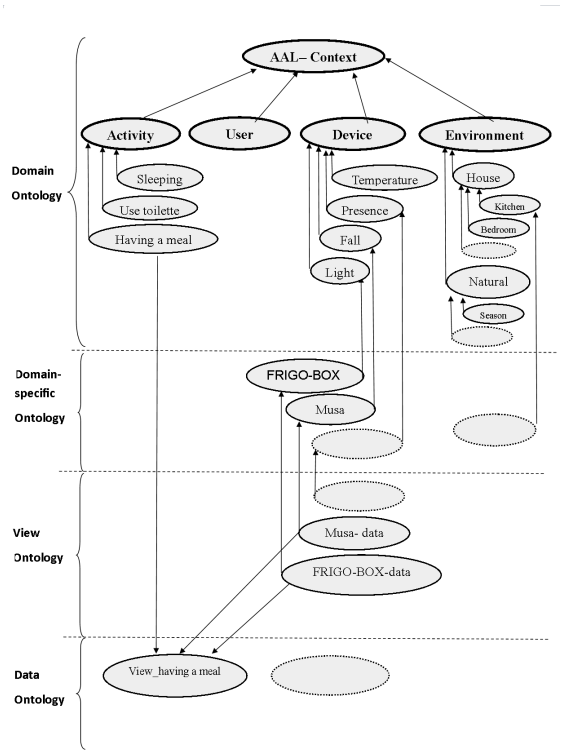


Figure 5. Class hierarchy diagram of OntoAALISABETH

dynamically changing dataset such traditional methods do not allow to perform reasoning over time and space, which is necessary to capture some of the important characteristics of streaming data and events. Since our goal is to monitor certain specific human activities in a domestic environment, we introduce a CEP engine in order to perform the temporal analysis procedure. This engine allows to combine data from multiple sources to infer events or patterns that suggest more complicated circumstances. In fact, the main objective is to recognize significant events. These identifications could be eventually reused to discover further more complex events, through additional uses of CEP engine.

E. Implementation of the framework

The OWL ontology is developed and tested in Protégé 4.3 [18], together with the Pellet Reasoner Plug-in [19], which permits the creation and population of equivalent classes. Through the Protégé Plug-in OBDA [12], we write down the statements that map the Database to the ontology, in order to enable the possibility of extracting data from the DB, which was written in MySQL. To implement the framework, we use Java as a coding language to combine several techniques. Thereby, we call functionalities of the OBDA Plug-in to establish a connection to the DB and effectively load the records in the ontology. Then, the ontology is managed by means of the OWL API. Thereafter, the Pellet reasoner is invoked through Jena [20] to perform reasoning over the ontology together with the individuals. The SPARQL query is also executed through Jena. Basically, using Jena we load the ontology file created with Protégé into an ontology model (a Java object implementing

 ISBN: 978-1-61208-355-1

the `OntModel` interface). We then choose to utilize Esper as CEP tool for several reasons: its open-source Java library for complex event processing, it can be used in different data stream and CEP applications, it has adapters that allow the user to provide different input formats for the representation of events. The whole Java framework is developed using Eclipse IDE [21].

IV. Conclusion and Future Work

In this paper, we have illustrated an ontology-based framework to retrieve semantic information from a data repository lacking of the original significance. The ontology represents the central element of the presented methodology, and is basically composed of four layers: a top-level ontology followed by a domain-specific one, and data layer which establishes over a final basis-view layer. This last part is thought as a data preprocessing step. It plays the role to organize data according to the desired context views, in order to allow a proper analysis. In the near future work, we intend to focus on the last part of the framework and carry out temporal pattern identifications. A further development of the CEP analysis method is needed to effectively perform recognition of pre-determined human activities. Once detected such behavioural events, they will be evaluated by means of the CEP engine, and compared with the existing recognition techniques, e.g., Bayesian networks, Hidden Markov Models, Learning Machine. Also, a feasible refinement to classify data will be the definition of custom SWRL rules, and their integration with the existing inference classes.

The ensemble of certain specific actions or behaviours can be considered as markers of some of the most common diseases affecting old people. Hence, discovering such behavioural sequences which commonly characterize diagnostic suspects represents the main motivation of the ongoing AAL-ISABETH project.

However, the AAL represents just one of the many possible domains of application for the introduced approach. Finally, another eventual domain of use could be a Smart City. Such modern urban system of devices connected in a common network has the intent to improve the quality of life and a sustainable economic development. A Smart City is an example of real-time monitoring system in a larger scale, and presents similarities to our dynamic and heterogeneous features. Hence, the proposed approach prescind from the size of the domain of application and can be proposed to manage the fine-grained data generated by heterogeneous networks.

Acknowledgements

The authors would like to thank every partner of AALISABETH project for the great working collaboration done until now. They also acknowledge the financial contribution of the Marche Region administration for supporting the research on the AALISABETH project.

References

[1] T. R. Gruber. (retrieved: March, 2014) What is an Ontology? [Online]. Available: http://www-ksl.stanford.edu/kst/what-is-an-ontology.html

[2] M. Gagnon, "Ontology-Based Integration of Data Sources," in *10th International Conference on Information Fusion*, Quebec, Canada, July 2007, pp. 1–8.

[3] D. Bonino, E. Castellina, and F. Corno, "The DOG gateway: Enabling Ontology-based Intelligent Domotic Environments," *IEEE Trans. Consumer Electronics*, vol. 54, pp. 1656–1664, November 2008.

[4] T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang, "An Ontology-based Context Model in Intelligent Environments," in *Proceedings of Communication Networks and Distributed System Modeling and Simulation Conference*, San Diego, California, USA, January 2004, pp. 270–275.

[5] D. Preuveneers, J. V. den Bergh, D. Wagelaar, A. Georges, P. Rigole, T. Clerckx, Y. Berbers, K. Coninx, V. Jonckers, and K. D. Bosschere, "Towards an Extensible Context Ontology for Ambient Intelligence," in *Second European Symposium on Ambient Intelligence*, ser. LNCS. Eindhoven, Netherlands: Springer, November 2004, pp. 148 – 159.

[6] F.-L. C. J. B. Mocholí, Sala Pidd and N. J. C., "Ontology for Modeling Interaction in Ambient Assisted Living Environments," in *Proc. IFMBE XII Mediterranean Conference on Medical and Biological Engineering and Computing 2010*, Chalkidiki, Greece, May 2010, pp. 566–658.

[7] A. Fleury, M. Vacher, N. Noury, and S. Member, "SVM-Based Multimodal Classification of Activities of Daily Living in Health Smart Homes: Sensors, Algorithms, and First Experimental Results," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, pp. 274–283, March 2010.

[8] S. M. Sedighi and R. Javidan, "Semantic Query in a Relational Database using a Local Ontology Construction," *South African Journal of Science*, vol. 108,, pp. 97–107, Oct 2012.

[9] J. Barrasa, Óscar Corcho, and A. Gómez-Pérez, "R2O, an Extensible and Semantically based Database-to-Ontology Mapping Language," in *in In Proceedings of the 2nd Workshop on Semantic Web and Databases(SWDB2004)*. Toronto, Canada: Springer, August 2004, pp. 1069–1070.

[10] (retrieved: May, 2014) OWL Web Ontology Language Document Overview (Second Edition). [Online]. Available: http://www.w3.org/TR/owl2-overview/

[11] N. Cullot, R. Ghawi, and K. Ytongnon, "DB2OWL:A Tool for Automatic Database-to-Ontology Mapping." in *SEBD'07*, Brindisi, Italy, June 2007, pp. 491–494.

[12] M. Rodriguez-Muro, L. Lubyte, and D. Calvanese, "Realizing ontology based data access: A plug-in for protg," in *In Proc. of the Workshop on Information Integration Methods, Architectures, and Systems (IIMAS 2008)*. Cancun, Mexico: IEEE Computer Society, April 2008, pp. 286–289.

[13] T. Tran Thi Kim and H. Werthner, "An Ontology Based Framework for Enriching Event Log Data," in *SEMAPRO 2011, The Fifth International Conference on Advances in Semantic Processing*, Lisbon, Portugal, November 2011, pp. 110–115.

[14] D. R. Ferreira and L. H. Thom, "A Semantic Approach to the Discovery of Workflow Activity Patterns in Event Logs," *International Journal of Business Process Integration and Management*, vol. 6, pp. 4–17, July 2012.

[15] D. J. Patterson, D. Fox, H. Kautz, and M. Philipose, "Fine-Grained Activity Recognition by Aggregating Abstract Object Usage," in *Proceedings of the Ninth IEEE International Symposium on Wearable Computers*, ser. ISWC '05, 2005, pp. 44–51.

[16] K. Taylor and L. Leidinger, "Ontology-driven complex event processing in heterogeneous sensor networks," in *Proceedings of the 8th Extended Semantic Web Conference on The Semantic Web: Research and Applications*, Heraklion, Crete, Greece, June 2011, pp. 285–299.

[17] D. Anicic, P. Fodor, S. Rudolph, and N. Stojanovic, "Ep-sparql: A unified language for event processing and stream reasoning," in *Proceedings of the 20th International Conference on World Wide Web*. Hyderabad, India: ACM, April 2011, pp. 635–644.

[18] (retrieved: June, 2014) Protege. [Online]. Available: http://protege.stanford.edu/

[19] (retrieved: June, 2014) Pellet. [Online]. Available: http://clarkparsia.com/pellet/protege/

[20] (retrieved: June, 2014) Apache jena. [Online]. Available: http://jena.sourceforge.net/

[21] (retrieved: June, 2014) Eclipse. [Online]. Available: https://www.eclipse.org/

 ISBN: 978-1-61208-355-1

Word Sense Disambiguation Based on Semi-automatically Constructed Collocation Dictionary

Minoru Sasaki, Kanako Komiya, Hiroyuki Shinnou
Dept. of Computer and Information Sciences
Faculty of Engineering, Ibaraki University
Email: {msasaki, kkomiya, shinnou}@mx.ibaraki.ac.jp

***Abstract*—In this paper, we propose a novel Word Sense Disambiguation (WSD) method based on collocation that has a particular meaning. This proposed method is to identify the sense of idiom or common phrase containing a target word before the existing statistical WSD method is applied by capturing the context information. To evaluate the efficiency of the proposed method using collocation dictionary, we make some experiments to compare with the result of the Support Vector Machine (SVM) classification. The results of the experiments show that almost the sense of the extracted collocation has only one particular sense when we obtain the word pair of (the target word, noun word) and (noun word, the target word) with high pointwise mutual information value. Moreover, in the experiment of WSD task, the total average precision of our system is improved in comparison with the baseline system using SVM.**

Keywords-word sense disambiguation; one sense per collocation; sense-tagged collocation dictionary construction

I. INTRODUCTION

Word Sense Disambiguation (WSD) [1] is one of the major tasks in natural language processing. WSD is the process of identifying the most appropriate sense for a polysemous word in a sentence. If we have training data which has already been disambiguated manually, the WSD task reduces to a classification problem based on supervised learning. In this approach, we construct a classifier to assign a word sense to new example by analyzing co-occurrence statistics of a target word. When we assign a sense to a word automatically, we can construct a sense tagged corpus and a case frame dictionary. To construct large-sized training data, language dictionary and thesaurus, it is increasingly important to further improve to select the most appropriate meaning of the ambiguous word.

WSD methods based on supervised learning exploit two powerful constraints: "one sense per collocation" [10] and "one sense per discourse" [3]. In the "one sense per collocation", the nearby words provide clues to the sense of the target word. "One sense per discourse" represents the sense that a target word is consistent with a given document. In the WSD research literature, currently, these two assumptions are widely accepted by natural language processing community and allow a supervised classifier with features based on context information to achieve enhanced classification performance.

Recent work develops above these assumptions into statistical models based on local and topical features surrounding a target word to be disambiguated [4] [7]. However, even when we make use of these assumptions, it is difficult to identify the sense of common expressions or idioms containing a target word. For example, the word "place" means general location. But, the meaning of the idiom "take place" is quite different from the meaning of "take her place". The idiom "take place" means that something occurs or happens at a particular time or place. Thus, an idiom is a group in a fixed order and has a particular meaning that is different from the meanings of the individual words regardless of context of the word to be disambiguated. Although there are many researches to solve WSD problem using phrase in WordNet and idiom dictionary, when we take into consideration the overall occurrence in the target corpus, there still remains some cases where a dictionary may not cover some of the idioms that exist in the target corpus.

In this paper, to solve this problem, we propose a novel word sense disambiguation method that aims to identify the sense of idiom and common phrase. In this method, we first extract idioms containing a target word and assign an appropriate sense to each of the extracted idioms manually to construct a idiom/collocation dictionary. Then, we identify the sense of idiom and common phrase before the existing statistical WSD method is applied by capturing the context information. Thus, this method enables us to identify the sense of a phrase that has a particular meaning regardless of context of the word such as metaphor expressions and idioms. A series of experiments shows our idiom sense identification effectively contributes to WSD precision.

The rest of this paper is organized as follows. Section 2 is devoted to the introduction of the related work in the literature. Section 3 describes a collocation dictionary generation method. Section 4 illustrates the proposed WSD system. In Section 5, we describe an outline of experiments. Experimental results are presented in Section 6. Finally, Section 7 concludes the paper.

II. RELATED WORKS

In this section, some previous research using such information will be compared with our proposed method.

Most WSD research has been focused on automatically assigning an appropriate sense to each occurrence of a target

 ISBN: 978-1-61208-355-1

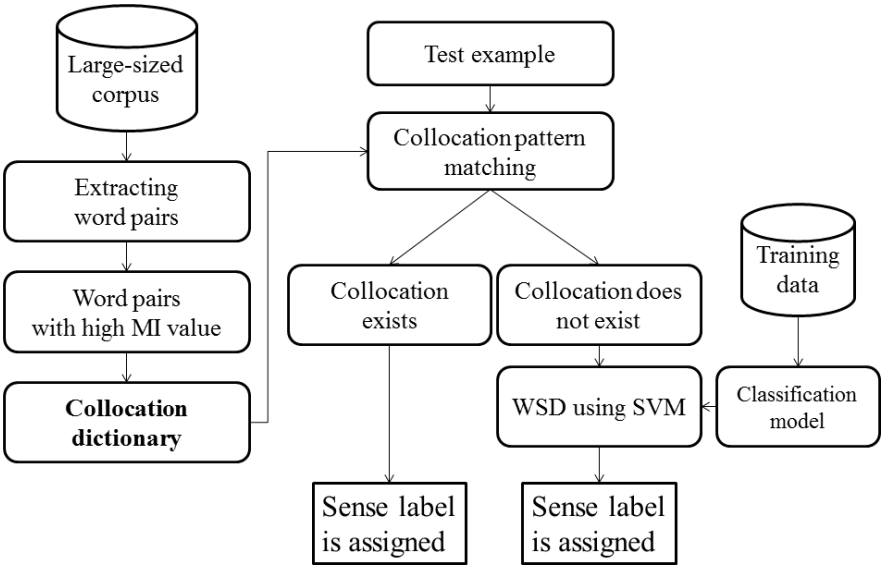


Figure 1. Overview of the proposed system

word in a text. In this research, many systems exploit two powerful constraints: "one sense per collocation" [10] and "one sense per discourse" [3]. Yarowsky's algorithm [10] employs an iterative bootstrapping approach. It starts from a small amount of seed collocation for the target word and assigning a sense using a decision list. The sense assignment process repeats until the whole corpus is consumed. Gale et al. [3] examines that there is a strong tendency for an ambiguous word to share the same sense in a well-written discourse.

In some previous research, collocation dictionary has been applied to the gloss disambiguation task. Yarowsky describes an unsupervised learning algorithm to perform WSD for unannotated English text. This method is to estimate the weighting using log-likelihood from the training set of data [11]. To identify an appropriate sense, it uses only nouns and considers only the two senses of a target word. However, in general, WSD task is a multi-class problem, as there can be more than two senses for a target word. Jimeno-Yepes et al. work on a knowledge-based WSD approach using collocation analysis [5]. This method extracts synonyms and collocations from meta-thesaurus to be added as alternative wordings of the target word. However, this system obtains related terms from the Unified Medical Language System meta-thesaurus [5], so that it does not take into consideration idioms and common expressions. There are some graph based approaches for knowledge based WSD, such as structural pattern recognition framework [8] and HyperLex [2].

III. Generating Collocation dictionary

In this section, we first describe the overview of generating collocation dictionary. From untagged corpora, we extract collocations of a given word in a semi-automatic manner. For more precise collocation data, the massive size of the untagged corpus is required. It is hard to get a large scale tagged corpus so that we use an untagged corpus for extracting collocations.

To extract collocations from large scale corpora, we explore the corpora to obtain the current and previous word pair, the current and next word pair, as well as the Part-Of-Speech tag of the previous and next words. We calculate the frequency of each word pair and use Pointwise Mutual Information (PMI) with each of the word pairs. The PMI is a popular measure of co-occurrence statistics of two words x and y in the data set as follows:

$$\mathrm{PMI}(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)}, \quad (1)$$

where $P(x,y)$ is the probability of the word pair occurring together, $P(x)$ is the probability of the word x occurring and $P(y)$ is the probability of the word y occurring.

Then, we take all word pairs that exceed a certain threshold value of mutual information and consider them as collocation. Finally, we assign a sense tag to each of the extracted collocations manually to construct a collocation dictionary. Collocation has a particular meaning that is different from the meanings of the individual words regardless of context of the word to be disambiguated.

IV. Word Sense Disambiguation method Using Sense-tagged Collocation

In this section, we describe the details of the WSD classifier construction using sense-tagged collocation dictionary as mentioned in the previous section. The proposed method is composed of two stages that are WSD using the collocation dictionary and WSD using supervised learning. The overall system of the proposed method is illustrated in the Figure 1.

A. Word Sense Disambiguation using Collocation

Using the constructed collocation dictionary, we first learn the decision list (a set of rules) from the collocation dictionary to disambiguate collocation sense. For all examples of the test data, we explore collocation patterns in the decision list and

apply the decision list classifier. When the collocation patterns are found in the example, the set of rules is used to assign its corresponding sense to the collocation. However, if word pairs are not found in the decision list, no sense label is assigned for the target word in this stage.

B. Supervised Learning Using Support Vector Machine

For the sentences in which sense of the target word is not assigned throughout the test data at the first stage, we next use an implementation of a Support Vector Machine algorithm to train the classifier using context information and assign a particular sense to the target word at the second stage.

At the first step, we extract a set of features (nouns and verbs) that have co-occurred with the target word from each sentence in the training and test data. Then, each feature set is represented as a vector by calculating co-occurrence frequencies of the words. For each target word, we can obtain a matrix derived from the set of word co-occurrence vectors.

For the obtained matrix, classification model is constructed by using Support Vector Machine (SVM). When the classification model is obtained by training data, we predict one sense for each test example using this model. When a new sentence including the target word is given, the sense of the target word is classified to the most plausible sense based on the obtained classification model. To employ the SVM for distinguishing more than two senses, we use one-versus-rest binary classification approach for each sense.

V. Experiments

To evaluate the efficiency of the proposed method using collocation dictionary, we make some experiments to compare with the result of the SVM classification. In this section, we describe an outline of the experiments.

A. Data

To construct a collocation dictionary, we used the white papers and best-selling books in the BCCWJ corpus which is a balanced corpus of one hundred million words of contemporary written Japanese [6]. The document sets of white papers and best-selling books consist of 1,500 documents (16.4MB) and 1,408 documents (13.4MB) respectively.

To evaluate our WSD method, we used the Semeval-2010 Japanese WSD task data set, which includes 50 target words comprising 22 nouns, 23 verbs, and 5 adjectives from the BCCWJ corpus [9]. In this data set, there are 50 training and 50 test instances for each target word. When we apply the SVM to identify the sense of a target word, this training data of the target word is used to construct a classification model. The test data is used for evaluating the performance of the proposed WSD system.

B. Experiment on collocation extraction

In order to investigate the quality of the constructed collocation dictionary, we make some experiments using our collocation extraction method. To extract collocations, some conditions are to be fulfilled in each of the experiments. These conditions are summarized as follows:

TABLE I. PRECISION RATIO OF THE EXTRACTED COLLOCATION

PMI	1	2	3	4	5
Noun Only	0.975	0.979	0.988	0.980	0.923
All POS	0.787	0.770	0.765	0.789	0.842

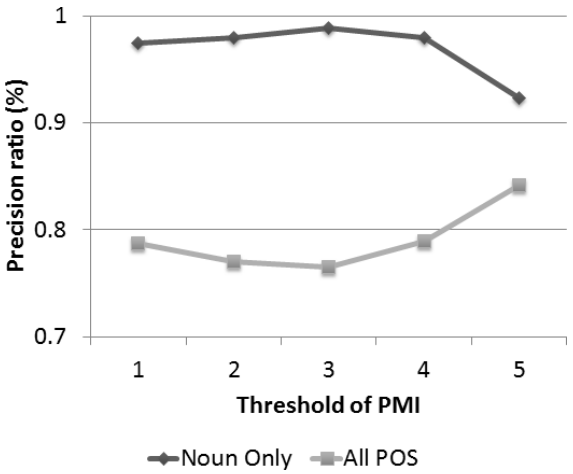


Figure 2. Precision ratio of the extracted collocation

- Part-Of-Speech(POS) of the previous and the next word (noun only or all POS)
- word pairs whose pointwise mutual information value is not less than the threshold k are considered as collocations ($k = 1,2,3,4,5$).

Under each of the above conditions, we construct a collocation dictionary and compare the quality of the extracted collocations. To evaluate the quality, we examine whether the sense of the extracted collocation has only one particular sense regardless of context of the target word. Then, we calculate the total number of correct collocations and the precision ratio of the number of collocations that have one particular sense to all extracted collocations for each target word. If the higher precision ratio is obtained, it turns out that the high quality collocation dictionary is constructed.

C. Experiment on WSD

To evaluate the results of the proposed method for the test data, we compare their performances with the results of simple SVM training. We obtain the precision value of each condition over all the examples to analyze the average performance of systems.

VI. Experimental Results

A. Quality of Collocation Extraction

Figure 2 and Table I show the result of the experiment of our collocation extraction method. In case that part-of-speech of both the previous and the next word is restricted to noun only, we obtain the high precision ratio. Therefore, almost the sense of the extracted collocation has only one particular sense, when we obtain the word pair of (the target word, noun

 ISBN: 978-1-61208-355-1

TABLE II. EXPERIMENTAL RESULTS OF WSD USING ADJACENCY NOUN

Accuracy	SVM	PMI=1	PMI=2	PMI=3	PMI=4	PMI=5
Ave.Prec.	0.690	0.704	0.696	0.694	0.693	0.691
Increase		17	13	9	6	2
Equal		31	35	41	44	48
Decrease		2	2	0	0	0

TABLE III. EXPERIMENTAL RESULTS OF WSD USING ALL ADJACENCY WORDS

Accuracy	SVM	PMI=1	PMI=2	PMI=3	PMI=4	PMI=5
Ave.Prec.	0.690	0.695	0.688	0.689	0.690	0.691
Increase		22	15	9	5	4
Equal		10	19	29	37	42
Decrease		18	16	12	8	4

word) and (noun word, the target word) with high PMI value. However, when the threshold value is 5, the precision value is decreased to 92.3%. The small number of the extracted collocation is obtained (197 collocations for $k = 1$ and 13 for $k = 5$) so that the precision ratio varies greatly.

In case that any part-of-speech is considered to the previous and the next word, the precision ratio is lower than the result using noun only. However, we obtain over 75% precision ration so that many word pairs have the potential to become collocation that has the particular sense.

B. Performance of WSD

Tables II and III show that the result of the experiment of WSD. In case that part-of-speech of both the previous and the next word is restricted to noun only, the total average precision of our system is improved in comparison with the baseline system using SVM. In the 50 target words, the precision of the only two words, "与える (ataeru; give, assign, ...)" and "経済 (keizai; economics, economy)", is decreased in comparison with the baseline system. These results are due to the failure to extract collocations that have a particular sense. However, if the threshold value k is larger than 3, the precision of our method has equal to the baseline system. In the data set used in these experiments, the number of training data is small so that many context words contained in the test data are not appeared in the training data. To improve the performance of the WSD system, we need to consider some additional information such as the glosses in WordNet and thesaurus.

In case that any part-of-speech is considered to the previous and the next word, the precision of our system is lower than that of the baseline. Using the threshold value $k = 1$, the precision of our system is higher. But, the precision of the 18 target words is decreased. Thus, the obtained collocation dictionary does not have good quality for disambiguating words, even though many collocations are extracted.

VII. Conclusion and Future Work

In this paper, we propose a novel word sense disambiguation method based on collocation that has a particular meaning. This proposed method is to identify the sense of idiom or common phrase containing a target word before the existing statistical WSD method is applied by capturing the context information. To evaluate the efficiency of the proposed method using collocation dictionary, we make some experiments to compare with the result of the SVM classification. The results of the experiments show that almost the sense of the extracted collocation has only one particular sense when we obtain the word pair of (the target word, noun word) and (noun word, the target word) with high PMI value. Moreover, in the experiment of WSD task, the total average precision of our system is improved in comparison with the baseline system using SVM. However, in case that any part-of-speech is considered to the previous and the next word, the precision of our system is lower than that of the baseline because the obtained collocation dictionary does not have good quality for disambiguating words.

Further work would be required to consider some additional information such as the glosses in WordNet, Wikipedia and other thesaurus to improve the performance of word sense disambiguation. Moreover, we need to consider a more syntactic information such as subject-verb-object relations and dependency structure to obtain more precise collocations.

References

[1] E. Agirre and P. Edmonds, *Word Sense Disambiguation: Algorithms and Applications*, 1st ed. Springer Publishing Company, Incorporated, 2007.

[2] E. Agirre, D. Martínez, O. L. de Lacalle, and A. Soroa, "Evaluating and optimizing the parameters of an unsupervised graph-based wsd algorithm," in *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, ser. TextGraphs-1. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 89–96.

[3] W. A. Gale, K. W. Church, and D. Yarowsky, "One sense per discourse," in *Proceedings of the Workshop on Speech and Natural Language*, ser. HLT '91. Stroudsburg, PA, USA: Association for Computational Linguistics, 1992, pp. 233–237.

[4] N. Ide and J. Véronis, "Word sense disambiguation: The state of the art," *Computational Linguistics*, vol. 24, pp. 1–40, 1998.

[5] A. Jimeno-Yepes, B. T. McInnes, and A. R. Aronson, "Collocation analysis for umls knowledge-based word sense disambiguation," *BMC Bioinformatics*, vol. 12, no. S-3, S4, 2011.

[6] K. Maekawa *et al.*, "Design, compilation, and preliminary analyses of balanced corpus of contemporary written japanese," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA), May 2010, pp. 1483–1486.

[7] R. Navigli, "Word sense disambiguation: A survey," *ACM Computing Surveys*, vol. 41, no. 2, pp. 10:1–10:69, Feb. 2009.

[8] R. Navigli and P. Velardi, "Structural semantic interconnections: A knowledge-based approach to word sense disambiguation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 7, pp. 1075–1086, Jul. 2005.

[9] M. Okumura, K. Shirai, K. Komiya, and H. Yokono, "Semeval-2010 task: Japanese wsd," in *Proceedings of the 5th International Workshop on Semantic Evaluation*, ser. SemEval '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 69–74.

[10] D. Yarowsky, "One sense per collocation," in *Proceedings of the Workshop on Human Language Technology*, ser. HLT '93. Stroudsburg, PA, USA: Association for Computational Linguistics, 1993, pp. 266–271.

[11] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, ser. ACL '95. Stroudsburg, PA, USA: Association for Computational Linguistics, 1995, pp. 189–196.

 ISBN: 978-1-61208-355-1

τOWL: A Framework for Managing Temporal Semantic Web Documents

Abir Zekri
University of Sfax
Sfax, Tunisia
abir.zekri@fsegs.rnu.tn

Zouhaier Brahmia
University of Sfax
Sfax, Tunisia
zouhaier.brahmia@fsegs.rnu.tn

Fabio Grandi
University of Bologna
Bologna, Italy
fabio.grandi@unibo.it

Rafik Bouaziz
University of Sfax
Sfax, Tunisia
raf.bouaziz@fsegs.rnu.tn

***Abstract*—The World Wide Web Consortium (W3C) OWL 2 Web Ontology Language (OWL 2) recommendation is an ontology language for the Semantic Web. It allows defining both schema (i.e., entities, axioms, and expressions) and instances (i.e., individuals) of ontologies. OWL 2 ontologies are stored as Semantic Web documents. However, OWL 2 lacks explicit support for time-varying schema or for time-varying instances. Hence, knowledge engineers or maintainers of semantics-based Web resources have to use ad hoc techniques in order to specify OWL 2 schema for time-varying instances. In this paper, for a disciplined and systematic approach to the temporal management of Semantic Web documents, we propose the adoption of a framework called Temporal OWL 2 (τOWL), which is inspired by the τXSchema framework defined for XML data. In a way similar to what happens in τXSchema, τOWL allows creating a temporal OWL 2 ontology from a conventional (i.e., non-temporal) OWL 2 ontology and a set of logical and physical annotations. Logical annotations identify which elements of a Semantic Web document can vary over time; physical annotations specify how the time-varying aspects are represented in the document. By using annotations to integrate temporal aspects in the traditional Semantic Web, our framework (i) guarantees logical and physical data independence for temporal schemas and (ii) provides a low-impact solution since it requires neither modifications of existing Semantic Web documents, nor extensions to the OWL 2 recommendation and Semantic Web standards.**

Keywords–Semantic Web; Ontology; OWL 2; τXSchema; Logical annotations; Physical annotations; Temporal database; XML Schema; XML

I. INTRODUCTION

Time is an omnipresent dimension in both classical and modern applications [1]; it is used to timestamp data values to keep track of changes in the real world and model their history. Hence, studying time has been, and continues to be, one of the main research interests in different scientific fields, such as databases and knowledge representation.

Since the second half of the 1980s, a great deal of work has been done in the field of temporal databases [2][3][4]. Several data models and query languages have been proposed for the management of time-varying data. Temporal databases usually adopt one or two time dimensions to timestamp data: (a) transaction-time, which indicates when an event is recorded in the database, and (b) valid-time, which represents the time when an event occurred, occurs or is expected to occur in the real world.

On the other hand, the World Wide Web (WWW or Web) [5] was shifted from the semi-structured internet to a more structured Web called the Semantic Web [6][7]. The new generation of Web aims to provide languages and tools that specify explicit semantics for data and enable knowledge sharing among knowledge-based applications. In this vision, ontologies [8] are used for defining and relating concepts that describe Web resources, in a formal way. The new emerging standard for describing ontologies, which has been recommended by the W3C since 2009, is OWL 2 [9][10][11]. It allows defining both schema (in terms of entities, axioms, and expressions) and instances (i.e., individuals) of ontologies; OWL 2 ontologies are stored as Semantic Web documents.

Due to the dynamic nature of the Web, ontologies that are used on the Web (like other Web application components such as Web databases, Web pages and Web scripts) evolve over time to reflect and model changes occurring in the real-world. Furthermore, several Semantic Web-based applications (like e-commerce, e-government and e-health applications) require keeping track of ontology evolution and versioning with respect to time, in order to represent, store and retrieve time-varying ontologies.

Unfortunately, while there is a sustained interest for temporal and evolution aspects in the research community [12], existing Semantic Web standards and state-of-the-art ontology editors and knowledge representation tools do not provide any built-in support for managing temporal ontologies. In particular, the W3C OWL 2 recommendation lacks explicit support for time-varying ontologies, at both schema and instance levels. Thus, knowledge engineers or maintainers of semantics-based Web resources must use ad hoc techniques when there is a need, for example, to specify an OWL 2 ontology schema for time-varying ontology instances. In the rest of the paper, we define as Knowledge Base Administrator (KBA) a knowledge engineer or, more in general, the person in charge of the maintenance of semantics-based Web resources.

According to what precedes, we think that if we would like to handle ontology evolution over time in an efficient manner and to allow historical queries to be executed on time-varying ontologies, a built-in temporal ontology

 ISBN: 978-1-61208-355-1

management system is needed. For that purpose, we propose in this paper a framework, called τOWL, for managing temporal Semantic Web documents, through the use of a temporal OWL 2 extension. In fact, we want to introduce with τOWL a principled and systematic approach to the temporal extension of OWL 2, similar to that Snodgrass and colleagues did to the eXtensible Markup Language (XML) with Temporal XML Schema (τXSchema) [13][14][15]. τXSchema is a framework (i.e., a data model equipped with a suite of tools) for managing temporal XML documents, well known in the database research community and, in particular, in the field of temporal XML [16]. Moreover, in our previous work [17][18][19], with the aim of completing the framework, we augmented τXSchema by defining necessary schema change operations acting on conventional schema, temporal schema, and logical and physical annotations (extensions which we plan to apply to τOWL too).

Being defined as a τXSchema-like framework, τOWL allows creating a temporal OWL 2 ontology from a conventional (i.e., non-temporal) OWL 2 ontology specification and a set of logical (or temporal) and physical annotations. Logical annotations identify which components of a Semantic Web document can vary over time; physical annotations specify how the time-varying aspects are represented in the document. By using temporal schema and annotations to introduce temporal aspects in the conventional (i.e., non temporal) Semantic Web, our framework (i) guarantees logical and physical data independence [20] for temporal schemas and (ii) provides a low-impact solution since it requires neither modifications of existing Semantic Web documents, nor extensions to the OWL 2 recommendation and Semantic Web standards.

The remainder of the paper is organized as follows. Section II motivates the need for an efficient management of time-varying Semantic Web documents. Section III describes the τOWL framework that we propose for extending the Semantic Web to temporal aspects: the architecture of τOWL is presented and details on all its components and support tools are given. Section IV discusses related work. Section V provides a summary of the paper and some remarks about our future work.

II. MOTIVATION

In this section, we present a motivating example that shows the limitation of the OWL 2 language for explicitly supporting time-varying instances. Then, we state the desiderata for an OWL 2 extension which could accommodate time-varying instances in a disciplined and systematic way.

A. Motivating Example

The Friend of a Friend (FOAF) project [21] is creating a Web of machine-readable pages describing people, the links between them and the things they create and do.

Suppose that the Web site "Web-S1" publishes the FOAF definition for his user "Nouredine". A fragment of the FOAF Resource Description Framework (RDF) document of "Nouredine" is presented in Fig. 1. It describes, according to the FOAF ontology, the personal information of "Nouredine" (i.e., name and nickname) and the information about his online accounts on diverse sites (i.e., the home page of the site, and the account name of the user). In this example, we limit to describe user's information concerning the account on the online Web site "Facebook".

Assume that information about the user "Nouredine" of the Web site "Web-S1" was added on 2014-01-15. On 2014-02-08, Nouredine modified his nickname from "Nor" to "Nouri" and his account name of Facebook from "Nor_Tunsi" to "Nouri_Tunsi". Thus, the corresponding fragment of the Nouredine FOAF RDF document was revised to that shown in Fig. 2.

```
…
<foaf:Person rdf:ID="#Person1">
  <foaf:name>Nouredine Tounsi</foaf:name>
  <foaf:nick>Nor</foaf:nick>
  <foaf:holdsAccount>
    <foaf:OnlineAccount
          rdf:about="https://www.facebook.com/
          Nouredine.Tounsi">
      <foaf:accountName>Nor_Tunsi
      </foaf:accountName>
    </foaf:OnlineAccount>
  </foaf:holdsAccount>
</foaf:Person>
…
```

Figure 1. A fragment of Nouredine FOAF RDF document on 2014-01-15.

```
…
<foaf:Person rdf:ID="#Person1">
  <foaf:name>Nouredine Tounsi</foaf:name>
  <foaf:nick>Nouri</foaf:nick>
  <foaf:holdsAccount>
    <foaf:OnlineAccount
          rdf:about="https://www.facebook.com/
          Nouredine.Tounsi">
      <foaf:accountName>Nouri_Tunsi
      </foaf:accountName>
    </foaf:OnlineAccount>
  </foaf:holdsAccount>
</foaf:Person>
...
```

Figure 2. A fragment of Nouredine FOAF RDF document on 2014-02-08.

In many Semantic Web-based applications, the history of ontology changes is a fundamental requirement, since such a history allows recovering past ontology versions, tracking changes over time, and evaluating temporal queries [22]. A τOWL time-varying Semantic Web document records the evolution of a Semantic Web document over time by storing all versions of the document in a way similar to that originally proposed for τXSchema [13].

Suppose that the webmaster of the Web site "Web-S1" would like to keep track of the changes performed on our FOAF RDF information by storing both versions of Fig. 1 and of Fig. 2 in a single (temporal) RDF document. As a result, Fig. 3 shows a fragment of a time-varying Semantic Web document that captures the history of the specified information of "Nouredine".

```
…
<foaf:Person rdf:ID="#Person1">
  <foaf:name>Nouredine Tounsi</foaf:name>
  <versionedNick>
    <NickVersion>
```

```
      <nickValidityStartTime>2014-01-15
      </nickValidityStartTime>
      <nickValidityEndTime>2014-02-07
      </nickValidityEndTime>
      <foaf:nick>Nor</foaf:nick>
    </NickVersion>
    <NickVersion>
      <nickValidityStartTime>2014-02-08
      </nickValidityStartTime>
      <nickValidityEndTime>now
      </nickValidityEndTime>
      <foaf:nick>Nouri</foaf:nick>
    </NickVersion>
  </versionedNick>
  <foaf:holdsAccount>
    <foaf:OnlineAccount
          rdf:about="https://www.facebook.com/
          Nouredine.Tounsi">
      <versionedAccountName>
        <AccountNameVersion>
          <accountNameValidityStartTime>
               2014-01-15
          </accountNameValidityStartTime>
          <accountNameValidityEndTime>
               2014-02-07
          </accountNameValidityEndTime>
          <foaf:accountName>Nor_Tunsi
          </foaf:accountName>
        </AccountNameVersion>
        <AccountNameVersion>
          <accountNameValidityStartTime>
               2014-02-08
          </accountNameValidityStartTime>
          <accountNameValidityEndTime>
               now
          </accountNameValidityEndTime>
          <foaf:accountName>Nouri_Tunsi
          </foaf:accountName>
        </AccountNameVersion>
      </versionedAccountName>
    </foaf:OnlineAccount>
  </foaf:holdsAccount>
</foaf:Person>
...
```

Figure 3. A fragment of the time-varying Nouredine FOAF RDF document.

In this example, we use valid-time to capture the history of Nouredine information. In order to timestamp the entities which can evolve over time, we use the following optional tags: **nickValidityStartTime** and **nickValidityEndTime**, for recording **nick** name evolution, and **accountNameValidityStartTime** and **accountNameValidityEndTime**, for keeping the **accountName** history. These are optional Data Properties which can be added to a temporal entity. The domain of nickValidityEndTime or accountNameValidityEndTime includes the value "now" [23]; the entity that has now as the value of its validity end time property represents the current entity until some change occurs.

Assume that the extract of the FOAF ontology presented in Fig. 4 contains the conventional (i.e., non-temporal) schema [13] for the FOAF RDF document presented in both Fig. 1 and Fig. 2. The conventional schema is the schema for an individual version, which allows updating and querying individual versions.

```
<rdf:RDF>
  <owl:Ontology
      rdf:about="http://purl.org/az/foaf#">
    <rdfs:Class rdf:about="#Person">
      <rdf:type
         rdf:resource="http://www.w3.org/2002/
         07/owl#Class"/>
    </rdfs:Class>
    <rdf:Property rdf:about="#holdsAccount">
      <rdf:type
          rdf:resource="http://www.w3.org/2002/
          07/owl#ObjectProperty"/>
      <rdfs:domain rdf:resource="#Person"/>
      <rdfs:range
          rdf:resource="#OnlineAccount"/>
    </rdf:Property>
    <rdf:Property rdf:about="#accountName">
      <rdf:type
          rdf:resource="http://www.w3.org/2002/
          07/owl#DatatypeProperty"/>
      <rdfs:domain
          rdf:resource="#OnlineAccount"/>
    </rdf:Property>
    …
</rdf:RDF>
```

Figure 4. An RDF/XML extract from the OWL 2 FOAF ontology.

The problem is that the time-varying ontology document (see Fig. 3) does not conform to the conventional ontology schema (see Fig. 4). Thus, to resolve this problem, we need a different ontology schema that can describe the structure of the time-varying ontology document. This new schema should specify, for example, timestamps associated to entities, time dimensions involved, and how the entities vary over time.

B. *Desiderata*

There are several goals which can be fulfilled when augmenting the OWL 2 language to support time-varying instances. Our approach aims to satisfy the following requirements.

- Facilitating the management of time for KBAs.
- Supporting both valid time and transaction time.
- Supporting (temporal) versioning of OWL 2 instances.
- Keeping compatibility with existing OWL 2 W3C recommendations, standards, and editors, and not requiring any changes to these recommendations, standards, and tools.
- Supporting existing applications that are already using OWL 2 ontologies.
- Providing OWL 2 data independence so that changes at the logical level are isolated from those performed at the physical level, and vice versa.
- Accommodating a variety of physical representations for time-varying OWL 2 instances.

III. THE τOWL FRAMEWORK

This section presents our framework τOWL for handling temporal Semantic Web documents and provides an illustrative example of its use. It describes the architecture of τOWL and the tools used for managing both τOWL schema and τOWL instances. Since τOWL is a τXSchema-like framework, we were inspired by the τXSchema architecture and tools while defining the architecture and tools of τOWL.

 ISBN: 978-1-61208-355-1

The τOWL framework allows a KBA to create a temporal OWL 2 schema for temporal OWL 2 instances from a conventional OWL 2 schema, logical annotations, and physical annotations. Since it is a τXSchema-like framework, τOWL use the following principles:

- separation between (i) the conventional (i.e., non-temporal) schema and the temporal schema, and (ii) the conventional instances and the temporal instances;
- use of temporal and physical annotations to specify temporal and physical aspects, respectively, at schema level.

Fig. 5 illustrates the architecture of τOWL. Notice that only the components which are shaded in the figure are specific to an individual time-varying OWL 2 document and need to be supplied by a KBA. The framework is based on the OWL 2 language [9], which is a W3C standard ontology language for the Semantic Web. It allows defining both schema (i.e., entities, axioms, and expressions) and instances (i.e., individuals) of ontologies. Thus, we consider that the signature of an OWL 2 ontology O can be defined as follows: O = {E, A, Exp} such that:

i) E = {C, DP, OP, AP} represents the set of the entities with:
- C: Class, represents the set of concepts;
- DP: Data Property, represents the set of properties of the concepts;
- OP: Object Property, represents the set of the semantic relations between the concepts;
- AP: Annotation Property, represents the set of annotations on the entities and those on the axioms.

ii) A = {EAx, KAx} represents the set of axioms with:
- EAx: Entity Axioms, represents the axioms which concern the entities;
- KAx: Key Axioms, represents all the identifiers associated to the various classes.

iii) Exp = {CE, OPE, DPE} represents the set of the used expressions (an expression is a complex description which results from combinations of entities by using constructors such as enumeration, restriction of cardinality and restriction of properties) with:
- CE: Class Expressions, represents the set of combinations of concepts by using constructors;
- OPE: Object Property Expressions, represents the set of combinations of relations;
- DPE: Data Property Expressions, represents the set of combinations of properties.

The KBA starts by creating the *conventional schema* (box 6), which is an OWL 2 ontology that models the concepts of a particular domain and the relations between these concepts, without any temporal aspect. To each conventional schema corresponds a set of conventional (i.e., non-temporal) OWL 2 instances (box 11). Any change to the conventional schema is propagated to its corresponding instances.

After that, the KBA augments the conventional schema with *logical* and *physical annotations*, which allow him/her to express in an explicit way all requirements dealing with the representation and the management of temporal aspects associated to the components of the conventional schema, as described in the following.

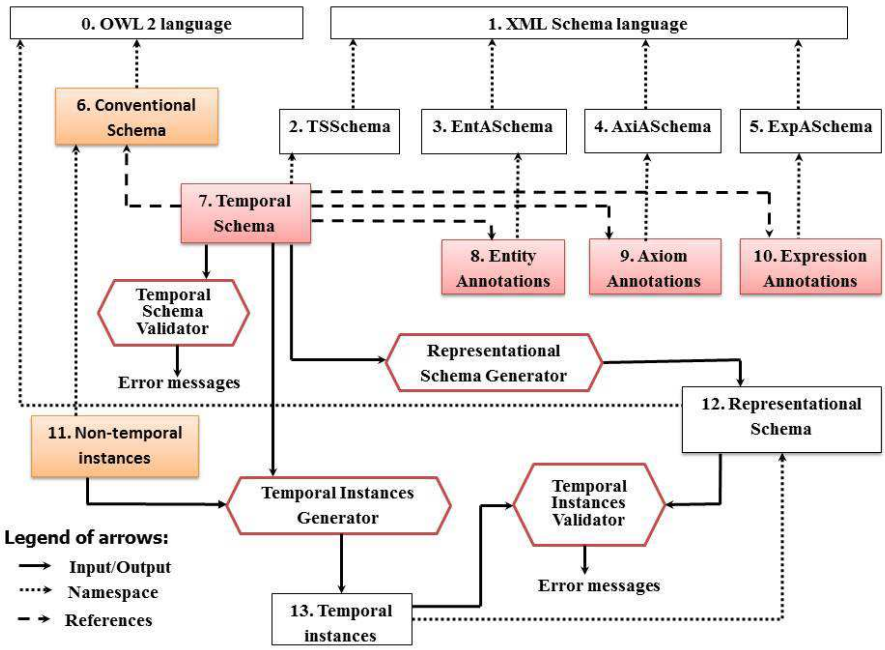


Figure 5. τOWL overall architecture.

Logical annotations [15] allow the KBA to specify (i) whether a conventional schema component varies over valid time and/or transaction time, (ii) whether its lifetime is described as a continuous state or a single event, (iii) whether the component may appear at certain times (and not at others), and (iv) whether its content changes. If no logical annotations are provided, the default logical annotation is that anything can change. However, once the conventional schema is annotated, components that are not described as time-varying are static and, thus, they must have the same value across every instance document (box 11).

Physical annotations [15] allow the KBA to specify the timestamp representation options chosen, such as where the timestamps are placed and their kind (i.e., valid time or transaction time) and the kind of representation adopted. The location of timestamps is largely independent of which components vary over time. Timestamps can be located either on time-varying components (as specified by the logical annotations) or somewhere above such components. Two OWL 2 documents with the same logical information will look very different if we change the location of their physical timestamps. Changing an aspect of even one timestamp can make a big difference in the representation. τOWL supplies a default set of physical annotations, which is to timestamp the root element with valid and transaction times. However, explicitly defining them can lead to more compact representations [15].

In order to improve conceptual clarity and also to enable a more efficient implementation, we adopt a "separation of concerns" principle in our approach: since the entities, the axioms and the expressions of an OWL 2 ontology evolve over time independently, we distinguish between three separate types of annotations to be defined and to be associated to a conventional schema: the *entity annotations* (box 8), the *axiom annotations* (box 9) and the *expression annotations* (box 10).

Entity annotations describe the logical and physical characteristics associated to the components of an OWL 2 ontology: classes, relations and properties. They indicate for example the temporal formats of these components which could be valid-time, transaction-time, bi-temporal or snapshot (by default). The schema for the logical and physical entity annotations is given by EntASchema (box 3). Axiom annotations and expression annotations describe the logical and physical aspects of axioms and expressions defined on classes or on properties. The schema for the logical and physical axiom annotations is given by AxiASchema (box 4) and the schema for the logical and physical expression annotations is given by ExpASchema (box 5).

Notice that AntASchema, AxiASchema, and ExpASchema, which all contain both logical and physical annotations, are XML Schemas [24]. The annotations associated to the same conventional schema can evolve independently. Any change to one of the three sets of annotations does not affect the two other sets.

Finally, the KBA creates the *temporal schema* (box 7) in order to provide the linking information between the conventional schema and its corresponding logical and physical annotations. The temporal schema is a standard XML document which ties the conventional schema, the entity annotations, the axiom annotations, and the expression annotations together. In the τOWL framework, the temporal schema is the logical equivalent of the conventional OWL 2 schema in a non-temporal context. This document contains sub-elements that associate a series of conventional schema definitions with entity annotations, axiom annotations, and expression annotations, along with the time span during which the association was in effect. The schema for the temporal schema document is the XML Schema Definition document *TSSchema* (box 2).

Notice that, whereas *TSSchema* (box 2), *AntASchema* (box 3), *AxiASchema* (box 4), and *ExpASchema* (box 5) have been developed by us, OWL 2 (box 0) and XML Schema (box 1) correspond to the standards endorsed by the W3C.

In a way similar to what happens in the τXSchema framework, the temporal schema document (box 7) is processed by the ***temporal schema validator*** tool in order to ensure that the logical and physical entity annotations, axiom annotations and expression annotations are (i) valid with respect to their corresponding schemas (i.e., *AntASchema*, *AxiASchema*, and *ExpASchema*, respectively), and (ii) consistent with the conventional schema. The temporal schema validator tool reports whether the temporal schema document is valid or invalid.

Once all the annotations are found to be consistent, the ***representational schema generator*** tool generates the *representational schema* (box 12) from the temporal schema (i.e., from the conventional schema and the logical and physical annotations); it is the result of transforming the conventional schema according to the requirements expressed through the different annotations. The representational schema becomes the schema for temporal instances (box 13). Temporal instances could be automatically created from the *non-temporal instances* (box 11) and the temporal schema (box 7), using the ***temporal instances generator*** tool (such an operation is called "squash" in the original τXSchema approach). Moreover, temporal instances are validated against the representational schema through the ***temporal instances validator*** tool which reports whether the temporal instances document (box 13) is valid or invalid.

Notice that the four mentioned tools (i.e., Temporal Schema Validator, Temporal Instances Validator, Representational Schema Generator, and Temporal Instances Generator) are under development. For example, the temporal instances validator tool is being implemented as a temporal extension of an existing conventional ontology instance validator.

Illustrative example. In order to show the functioning of the proposed approach, we provide in the following an example that shows how management of temporal ontology document versions is dealt with in our τOWL approach.

Let us resume the example of Sec. II.A. On 2014-01-15, the KBA creates a conventional ontology schema, named "PersonSchema_V1.owl" (as in Fig. 4), and a conventional ontology document, named "Persons_V1.rdf" (as in Fig. 1), which is valid with respect to this schema. Suppose that the

 ISBN: 978-1-61208-355-1

KBA defines also a set of logical and physical annotations, associated to that conventional schema; they are stored in an ontology annotation document titled "PersonAnnotations_V1.xml" as shown in Fig. 6.

```
<?xml version="1.0" encoding="UTF-8"?>
<ontologyAnnotationSet>
  <logicalAnnotations>
    <item target="/Person/nick">
      <validTime kind="state"
                 content="varying"
                 existence="constant"/>
    </item>
  </logicalAnnotations>
  <physicalAnnotations>
    <stamp target="Person/nick"
           dataInclusion="expandedVersion">
      <stampkind timeDimension="validTime"
                 stampBounds="extent"/>
    </stamp>
  </physicalAnnotations>
</ontologyAnnotationSet>
```

Figure 6. The annotation document on 2014-01-15.

After that, the KBA creates the temporal ontology schema in Fig. 7, that ties "PersonSchema_V1.owl" and "PersonAnnotations_V1.xml" together; this temporal schema is saved in an XML file titled "PersonTemporalSchema.xml". Consequently, the Temporal Instances Generator tool uses the temporal ontology schema of Fig. 7 and the conventional ontology document in Fig. 1 to create a temporal document as in Fig. 8, that lists both versions (i.e., temporal "slices") of the conventional ontology documents with their associated timestamps. The squashed version of this temporal document, which could be generated by the Temporal Instances Generator, is provided in Fig. 9.

On 2014-02-08, the KBA updates the conventional ontology document "Persons_V1.rdf" as presented in Sec. II.A to produce a new conventional ontology document named "Persons_V2.rdf" (as in Fig. 2). Since the conventional ontology schema (i.e., PersonSchema_V1.owl) and the ontology annotation document (i.e., PersonAnnotations_V1.xml) are not changed, the temporal ontology schema (i.e., PersonTemporalSchema.xml) is consequently not updated. However, the Temporal Instances Generator tool updates the temporal document, in order to include the new slice of the conventional ontology document, as shown in Fig. 10. The squashed version of the updated temporal document is provided in Fig. 11.

```
<?xml version="1.0" encoding="UTF-8"?>
<temporalOntologySchema>
  <conventionalOntologySchema>
    <sliceSequence>
      <slice location="PersonSchema_V1.owl"
             begin="2014-01-15" />
    </sliceSequence>
  </conventionalOntologySchema>
  <ontologyAnnotationSet>
    <sliceSequence>
      <slice
        location="PersonAnnotations_V1.xml"
        begin="2014-01-15" />
    </sliceSequence>
  </ontologyAnnotationSet>
</temporalOntologySchema>
```

Figure 7. The temporal schema on 2014-01-15.

```
<?xml version="1.0" encoding="UTF-8"?>
  <td:temporalRoot
    temporalSchemaLocation="PersonTemporalSchema.xml
" />
    <td:sliceSequence>
      <td:slice location="Persons_V1.rdf"
                begin="2014-01-15" />
    </td:sliceSequence>
  </td:temporalRoot>
```

Figure 8. The temporal document on 2014-01-15.

```
<foaf:Person rdf:ID="#Person1">
  <foaf:name>Nouredine Tounsi</foaf:name>
  <nick_RepItem>
    <nick_Version>
      <timestamp_ValidExtent
              begin="2014-01-15" end="now" />
      <foaf:nick>Nor</foaf:nick>
    </nick_Version>
  </nick_RepItem>
  <foaf:holdsAccount>
    <foaf:OnlineAccount
          rdf:about="https://www.facebook.com/
          Nouredine.Tounsi">
      <accountName_RepItem>
        <accountName_Version>
          <timestamp_ValidExtent
                 begin="2014-01-15" end="now" />
          <foaf:accountName>Nor_Tunsi
          </foaf:accountName>
        </accountName_Version>
      </accountName_RepItem>
    </foaf:OnlineAccount>
  </foaf:holdsAccount>
</foaf:Person>
```

Figure 9. The squashed document correponding to the temporal document on 2014-01-15.

```
<?xml version="1.0" encoding="UTF-8"?>
  <td:temporalRoot
    temporalSchemaLocation="PersonTemporalSchema.xml
" />
    <td:sliceSequence>
      <td:slice location="Persons_V1.rdf"
                begin="2014-01-15" />
      <td:slice location="Persons_V2.rdf"
                begin="2014-02-08" />
    </td:sliceSequence>
  </td:temporalRoot>
```

Figure 10. The temporal document on 2014-02-08.

```
<foaf:Person rdf:ID="#Person1">
  <foaf:name>Nouredine Tounsi</foaf:name>
  <nick_RepItem>
    <nick_Version>
      <timestamp_ValidExtent begin="2014-01-15"
                             end="2014-02-07" />
      <foaf:nick>Nor</foaf:nick>
    </nick_Version>
    <nick_Version>
      <timestamp_ValidExtent begin="2014-02-08"
                             end="now" />
      <foaf:nick>Nouri</foaf:nick>
    </nick_Version>
  </nick_RepItem>
  <foaf:holdsAccount>
    <foaf:OnlineAccount
          rdf:about="https://www.facebook.com/
          Nouredine.Tounsi">
```

 ISBN: 978-1-61208-355-1

```
      <accountName_RepItem>
        <accountName_Version>
          <timestamp_ValidExtent
                      begin="2014-01-15"
                      end="2014-02-07"/>
          <foaf:accountName>Nor_Tunsi
          </foaf:accountName>
        </accountName_Version>
        <accountName_Version>
          <timestamp_ValidExtent
                      begin="2014-02-08"
                      end="now" />
          <foaf:accountName>Nouri_Tunsi
          </foaf:accountName>
        </accountName_Version>
      </accountName_RepItem>
    </foaf:OnlineAccount>
  </foaf:holdsAccount>
</foaf:Person>
```

Figure 11. The squashed document correponding to the temporal document on 2014-02-08.

Obviously, each one of the squashed documents (see Fig. 9 and Fig. 11) should conform to a particular schema, i.e., the representational schema, which is generated from the temporal schema shown in Fig. 7.

IV. Related Work Discussion

OWL-Time (formerly DAML-Time) [25] is a temporal ontology that has been developed for describing the temporal content of Web pages and the temporal properties of Web services. Excepting language constructs for representing time in ontologies, mechanisms for representing evolution of concepts (e.g., events) over time are absent. Furthermore, temporal relations cannot be expressed directly in OWL, since they are ternary (i.e., properties of objects that change in time involve also a temporal value in addition to the object and the subject); representing such temporal relations in OWL requires appropriate methods (e.g., 4D-fluents [26]). Our approach allows KBA representing (i) evolution of concepts over time, and (ii) temporal relations.

In [27], the authors present the annotation features of OWL 2 by showing that this latter allows for annotations on ontologies, entities, anonymous individuals, axioms (e.g., giving information about who asserted an axiom or when), and annotations themselves. In our work, we took another direction from using OWL 2 annotation features because we rather wanted to exploit the power of the τXSchema approach (e.g. including the exploitation of a τXSchema-like underlying infrastructure).

Time dimension(s) are explicitly added to Semantic Web languages and formalisms (e.g., RDF, OWL, and SPARQL Protocol and RDF Query Language (SPARQL)) in order to represent time in semantic annotations, to build temporal ontologies and to support temporal querying and reasoning. An annotated bibliography of previous work in this area is presented in [12], and a survey on the models and query languages for temporally annotated RDF is provided in [37]. In particular, in the literature, there are various contributions that propose to represent temporal data in the Semantic Web.

Gutiérrez et al. [28] presented a comprehensive framework to incorporate temporal reasoning into RDF, yielding temporal RDF graphs. They define a syntactic notion of temporal RDF graphs. A powerful system, called CHRONOS, for reasoning over temporal information in OWL ontologies is presented in [38]. Since qualitative representations are very common in natural language expressions such as in free text or speech and can be proven to be valuable in the Semantic Web, the authors choose to represent both qualitative temporal (i.e., information whose temporal extents are unknown such as “before”, “after” for temporal relations) and quantitative information (i.e., where temporal information is defined precisely, e.g., using dates). The CHRONOS reasoner can be applied to temporal relations in order to infer implied relations and to detect inconsistencies while retaining soundness, completeness and tractability over the supported relations set. As opposed to Gutiérrez et al. [28] and Anagnostopoulos et al. [38], in our present approach, we are not interested in temporal reasoning (and, thus, in spatio-temporal reasoning).

A model of a multi-temporal RDF Schema (RDFS) database is proposed in [29] where the author considered that this database is a set of RDF triples timestamped along the valid and/or transaction time axes. To enable querying such a database, an extension of SPARQL language [30], called T-SPARQL, has been defined in [22]. The paper [31] proposes a logic-based approach to introduce valid-time into RDFS and OWL 2 languages. An extension of SPARQL that can be used to query temporal RDF(S) and OWL 2 is also presented. Moreover, the author describes a general query evaluation algorithm that can be used with all entailment relations used in the Semantic Web. Finally, he presents two optimizations of the algorithm that are applicable to entailment relations characterized by a set of deterministic rules, such RDF(S) and OWL 2 RL/RDF Entailment. In [32], the authors introduce “The Valid Ontology” approach as a temporal extension of OWL. Indeed, they propose to use a single temporal XML document to represent and store a multi-version ontology and use a temporal XML query processor to efficiently extract valid OWL ontologies from the XML document as temporal snapshots. The result is an efficient ontology temporal versioning solution, relying on standard XML technology. Two complementary and alternative proposals for modeling temporally changing information in OWL are proposed in [33]. They are based on the perdurantist theory and benefit from results coming from the discipline of Formal Ontology, in order to restrict the appropriate use of the proposed frameworks. In the first proposal, the authors combine the perdurantist worm view with the notion of individual concepts for formulating a conceptual structure that allows one to separate from the information that define all the individuals the information concerning those that can possibly change. In the second proposal, they extend the first proposal with the distinction between objects and moments and the notion of qua individuals, where a qua individual is the way an object participates in a certain relation. With regard to Grandi [29], Motik [31], Grandi et al. [32], and Zamborlini et al. [33], our approach does not deal with modeling of time inside the ontology. It just supports temporal versioning.

 ISBN: 978-1-61208-355-1

O'Connor et al. [34] present a methodology and a set of tools for representing and querying temporal information in OWL ontologies. Their approach uses a lightweight temporal model to encode the temporal dimension of data. It also uses the OWL-based Semantic Web Rule Language (SWRL) and the SWRL-based OWL query language (SQWRL) to reason with and query the temporal information represented using the proposed model. By now, our approach does not support temporally-aware semantic rules.

The authors of [35] propose a new language, called temporal OWL (tOWL), which is an extension of the Ontology Web Language Description Logics (OWL-DL) to the temporal aspect. It enables the representation of time and change in dynamic domains. Through a layered approach, they introduce three extensions: (i) Concrete Domains, which allow the representation of restrictions using concrete domain binary predicates, (ii) Temporal Representation, which introduces timepoints, relations between timepoints, intervals, and Allen's 13 interval relations [36] into the language, and (iii) TimeSlices/Fluents, which implement a perdurantist view on individuals and enable the representation of complex temporal aspects such as process state transitions. The main purpose of our approach is to support past ontology versions, to be accessed via time-slice queries. We think that supporting temporal ontology versions is very interesting for several purposes and in different areas. The problem of not having temporal versions is that, e.g., if we have now to investigate on someone having put some illegal material on Facebook last week, we want to be able to individuate the account details even if they have been changed thereafter.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed τOWL, a τXSchema-like framework, which allows creating a temporal OWL 2 ontology from a conventional OWL 2 ontology and a set of logical and physical annotations. Our framework ensures logical and physical data independence, since it (i) separates conventional schema, logical annotations, and physical annotations, and (ii) allows each one of these three components to be changed independently and safely. Furthermore, adoption of τOWL provides for a low-impact solution, since it requires neither modifications of existing Semantic Web documents, nor extensions to the OWL 2 recommendation and Semantic Web standards. The extension of OWL 2 to temporal and versioning aspects is performed without having to depend on approval of proposed extensions by standardization committees (and on upgrade of existing tools conforming to standards to comply with approved extensions). In the next future, we intend to (i) study querying and updating instances of τOWL ontologies, and (ii) develop a prototype tool that shows the feasibility of our approach.

Our future work aims at extending τOWL to also support schema versioning [19][39] which is the most powerful technique for managing the history of schema changes, since (i) ontology schemata are also evolving over time to reflect changes in real-world applications [40], and (ii) keeping a fully fledged history of ontology changes, i.e. involving both the ontology instances and the ontology schema, is a required feature for many Semantic Web-based applications.

REFERENCES

[1] C. S. Jensen and R. T. Snodgrass, "Temporal Data Management," IEEE Transactions on Knowledge and Data Engineering, vol. 11, January/February 1999, pp. 36-44.

[2] O. Etzion, S. Jajodia, and S. Sripada (eds.), "Temporal Databases: Research and Practice," LNCS 1399, Springer-Verlag, 1998.

[3] C. S. Jensen and R. T. Snodgrass, "Temporal Database," in Liu L., Özsu M.T., (Eds.), Encyclopedia of Database Systems, Springer US, 2009, pp. 2957-2960.

[4] F. Grandi, "Temporal Databases," in M. Koshrow-Pour, (Ed.), Encyclopedia of Information Science and Technology (3rd Ed.), IGI Global, Hershey, in press.

[5] T. Berners-Lee, R. Cailliau, A. Luotonen, H. F. Nielsen, and A. Secret, "The World Wide Web," Communications of the ACM, vol. 37, August 1994, pp. 76-82.

[6] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," Scientific American, vol. 284, May 2001, pp. 34-43.

[7] Semantic Web project. <http://www.w3.org/2001/sw/> [retrieved: July, 2014]

[8] N. Guarino (Ed.), Formal Ontology in Information Systems, IOS Press, Amsterdam, 1998.

[9] W3C, OWL 2 Web Ontology Language – Primer (Second Edition), W3C Recommendation, 11 December 2012. <http://www.w3.org/TR/owl2-primer/> [retrieved: July, 2014]

[10] W3C, OWL 2 Web Ontology Language – Document Overview (Second Edition), W3C Recommendation, 11 December 2012. <http://www.w3.org/TR/owl2-overview/> [retrieved: July, 2014]

[11] W3C, OWL 2 Web Ontology Language – Profiles (Second Edition), W3C Recommendation, 11 December 2012. <http://www.w3.org/TR/owl2-profiles/> [retrieved: July, 2014]

[12] F. Grandi, "An Annotated Bibliography on Temporal and Evolution Aspects in the Semantic Web," SIGMOD Record, vol. 41, December 2012, pp. 18-21.

[13] F. Currim, S. Currim, C. E. Dyreson, and R. T. Snodgrass, "A Tale of Two Schemas: Creating a Temporal XML Schema from a Snapshot Schema with tXSchema," Proceedings of the 9th International Conference on Extending Database Technology (EDBT 2004), Heraklion, Crete, Greece, 14-18 March 2004, pp. 348-365.

[14] R. T. Snodgrass, C. E. Dyreson, F. Currim, S. Currim, and S. Joshi, "Validating Quicksand: Schema Versioning in τXSchema," Data Knowledge and Engineering, vol. 65, May 2008, pp. 223-242.

[15] F. Currim et al., "τXSchema: Support for Data- and Schema-Versioned XML Documents," TimeCenter Technical Report TR-91, 279 pages, September 2009. <http://timecenter.cs.aau.dk/TimeCenterPublications/TR-91.pdf> [retrieved: July, 2014]

[16] C. E. Dyreson and F. Grandi, "Temporal XML," in L. Liu and M. T. Özsu (Eds.), Encyclopedia of Database Systems, Springer US, 2009, pp. 3032-3035.

[17] Z. Brahmia, R. Bouaziz, F. Grandi, and B. Oliboni, "Schema Versioning in τXSchema-Based Multitemporal XML Repositories," Proceedings of the 5th IEEE International Conference on Research Challenges in Information Science (RCIS 2011), Guadeloupe - French West Indies, France, 19-21 May 2011, pp. 1-12.

[18] Z. Brahmia, F. Grandi, B. Oliboni, and R. Bouaziz, "Versioning of Conventional Schema in the τXSchema

 ISBN: 978-1-61208-355-1

Framework," Proceedings of the 8th International Conference on Signal Image Technology & Internet Systems (SITIS'2012), Sorrento – Naples, Italy, 25-29 November 2012, pp. 510-518.

[19] Z. Brahmia, F. Grandi, B. Oliboni, and R. Bouaziz, "Schema Change Operations for Full Support of Schema Versioning in the τXSchema Framework," International Journal of Information Technology and Web Engineering, in press, 2014. IGI Global.

[20] T. Burns et al., "Reference Model for DBMS Standardization, Database Architecture Framework Task Group (DAFTG) of the ANSI/X3/SPARC Database System Study Group," SIGMOD Record, vol. 15, March 1986, pp. 19-58.

[21] The Friend of a Friend (FOAF) project. <http://www.foaf-project.org/> [retrieved: July, 2014]

[22] F. Grandi, "T-SPARQL: a TSQL2-like temporal query language for RDF," Proceedings of the 1st International Workshop on Querying Graph Structured Data (GraphQ 2010), Novi Sad, Serbia, 20 September 2010, pp. 21-30.

[23] J. Clifford, C. Dyreson, T. Isakowitz, C. S. Jensen, and R. T. Snodgrass, "On the Semantics of "Now" in Databases," ACM Transactions on Database Systems, vol. 22, June 1997, pp. 171–214.

[24] XML Schema Part 0: Primer Second Edition, W3C Recommendation, 28 October 2004. <http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/> [retrieved: July, 2014]

[25] W3C, Time Ontology in OWL, W3C Working Draft, 27 september 2006. < http://www.w3.org/TR/owl-time/ > [retrieved: July, 2014]

[26] C. A. Welty and R. Fikes, "A Reusable Ontology for Fluents in OWL," Proceedings of the 4th International Conference on Formal Ontology in Information Systems (FOIS 2006), Baltimore, Maryland, USA, 9-11 November 2006, pp. 226-236.

[27] W3C, OWL 2 Web Ontology Language – New Features and Rationale (Second Edition), W3C Recommendation, 11 December 2012. <http://www.w3.org/TR/owl2-new-features/> [retrieved: July, 2014]

[28] C. Gutiérrez, C. A. Hurtado, and A. A. Vaisman, "Introducing time into RDF," IEEE Transactions on Knowledge and Data Engineering, vol. 19, February 2007, pp. 207-218.

[29] F. Grandi, "Multi-temporal RDF ontology versioning," Proceedings of the 3rd International Workshop on Ontology Dynamics (IWOD 2009), Washington DC, USA, 26 October 2009. CEUR Workshop Proceedings (CEUR-WS.org), Vol-519. <http://ceur-ws.org/Vol-519/grandi.pdf> [retrieved: July, 2014]

[30] W3C, SPARQL Query Language for RDF, W3C Recommendation, 15 January 2008, <http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/> [retrieved: July, 2014]

[31] B. Motik, "Representing and Querying Validity Time in RDF and OWL: A Logic-based Approach," Proceedings of the 9th International Semantic Web Conference (ISWC 2010), Shanghai, China, 7-11 November 2010, pp. 550-565.

[32] F. Grandi and M. R. Scalas, "The valid ontology: A simple OWL temporal versioning framework," Proceedings of the 3rd International Conference on Advances in Semantic Processing (SEMAPRO 2009), Sliema, Malta, 11-16 October 2009, pp. 98-102.

[33] V. Zamborlini and G. Guizzardi, "On the representation of temporally changing information in OWL," Workshops Proceedings of the 14th IEEE International Enterprise Distributed Object Computing Conference (EDOCW 2010), Vitória, Brazil, 25-29 October 2010, pp. 283-292.

[34] M. J. O'Connor and A. K. Das, "A method for representing and querying temporal information inOWL," In Biomedical Engineering Systems and Technologies, volume 127 of Communications in Computer and Information Science, pp. 97-110. Springer-Verlag, Heidelberg, Germany, 2011.

[35] V. Milea, F. Frasincar, and U. Kaymak, "tOWL: A Temporal Web Ontology Language," IEEE Transactions on Systems, Man, and Cybernetics, Part B, vol. 42, February 2012, pp. 268-281.

[36] J. F. Allen, "Maintaining Knowledge About Temporal Intervals," Communications of the ACM, vol. 26, November 1983, pp. 832-843.

[37] A. Analyti and I. Pachoulakis, "A survey on models and query languages for temporally annotated RDF," International Journal of Advanced Computer Science and Applications, vol. 3, September 2012, pp. 28-35.

[38] E. Anagnostopoulos, S. Batsakis, and E. G. M. Petrakis, "CHRONOS: A Reasoning Engine for Qualitative Temporal Information in OWL," Proceedings of the 17th International Conference in Knowledge-Based and Intelligent Information & Engineering Systems (KES 2013), Kitakyushu, Japan, 9-11 September 2013, pp. 70-77.

[39] J. F. Roddick, "Schema Versioning," in Liu L., Özsu M.T., (Eds.), Encyclopedia of Database Systems, Springer US, 2009, pp. 2499-2502.

[40] D. Rogozan and G. Paquette, "Managing ontology changes on the semantic web," Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2005), Compiegne, France, 19-22 September 2005, pp. 430-433.

 ISBN: 978-1-61208-355-1