

Developing DCO: The DebugIT Core Ontology for Antibiotics Resistance Modelling

Daniel Schober

Institute for Medical Biometry and
Medical Informatics, Freiburg Univer-
sity Medical Center, Germany
schober@imbi.uni-
freiburg.de

Martin Boeker

Institute for Medical Biometry and
Medical Informatics, Freiburg Univer-
sity Medical Center, Germany
beakmachine@gmail.com

Ilinca Tudose

Institute for Medical Biometry and
Medical Informatics, Freiburg Univer-
sity Medical Center, Germany
ilina.tudose@gmail.com

Stefan Schulz

Institute for Medical Biometry and
Medical Informatics, Freiburg Univer-
sity Medical Center, Germany
stschulz@imbi.uni-
freiburg.de

Abstract

Antibiotics resistance development in European hospitals has increased alarmingly in recent years. To counteract this danger, a semantic web based IT solution is proposed which intends to integrate the access to relevant clinical data repositories from different European hospitals. This endeavor relies on formalized and shared models of the clinical domain. We describe the development process of the DebugIT Core Ontology, which is a key-mediator for semantic as well as syntactic clinical data integration in the mentioned endeavor. We show how UML diagrams can be used to illustrate the ontology engineering phase and the ontologies use case. Some domain statements are given which are then converted into more human friendly representations to be verified by medical experts.

1 Introduction

Antibiotics resistance development poses a significant problem in today's hospital care. Massive amounts of clinical data relevant for this domain are being collected and stored in proprietary but unconnected systems in heterogeneous format, preventing re-use and exploitation of potentially valuable data. The **DebugIT** project (**D**etecting and **E**liminating **B**acteria **U**sin**G** **I**nformation **T**echnology, <http://www.debugit.eu/>), a large scale EU funded data integration project, intends to ana-

lyze antibiotics prescription practices and their outcomes across Europe and intends to exploit this knowledge to detect patient safety related patterns in distributed hospital data, i.e. to discover indicators for better treatments and ultimately antibiotics resistance prevention [1].

The challenge here is to establish a coherent and systematic exchange of rich data, harmonised across the different DebugIT sites and their Clinical Data Repositories (CDR), including information about patients, their illness situations, pathogens and antibiotics therapies. The semantic glue towards integrating such data is the DebugIT Core Ontology (DCO), an application ontology that enables data miners to query distributed CDRs in a semantically rich and content driven manner.

Here we outline basic DCO engineering methods, illustrate some example statements expressed in DCO and show how these are exploited by logics reasoners and visualization tools providing views readily understandable by biologists not acquainted with logics formalisms.

2 Methods

DCO is developed in the description logics RL flavor¹, using the Protégé 4.1 ontology editor [2]. The `dco.owl` file leverages on the domain upper level ontology BioTop [3] by direct `owl:import`. Detailed design principle docu-

¹ http://www.w3.org/TR/owl2-profiles/#OWL_2_RL

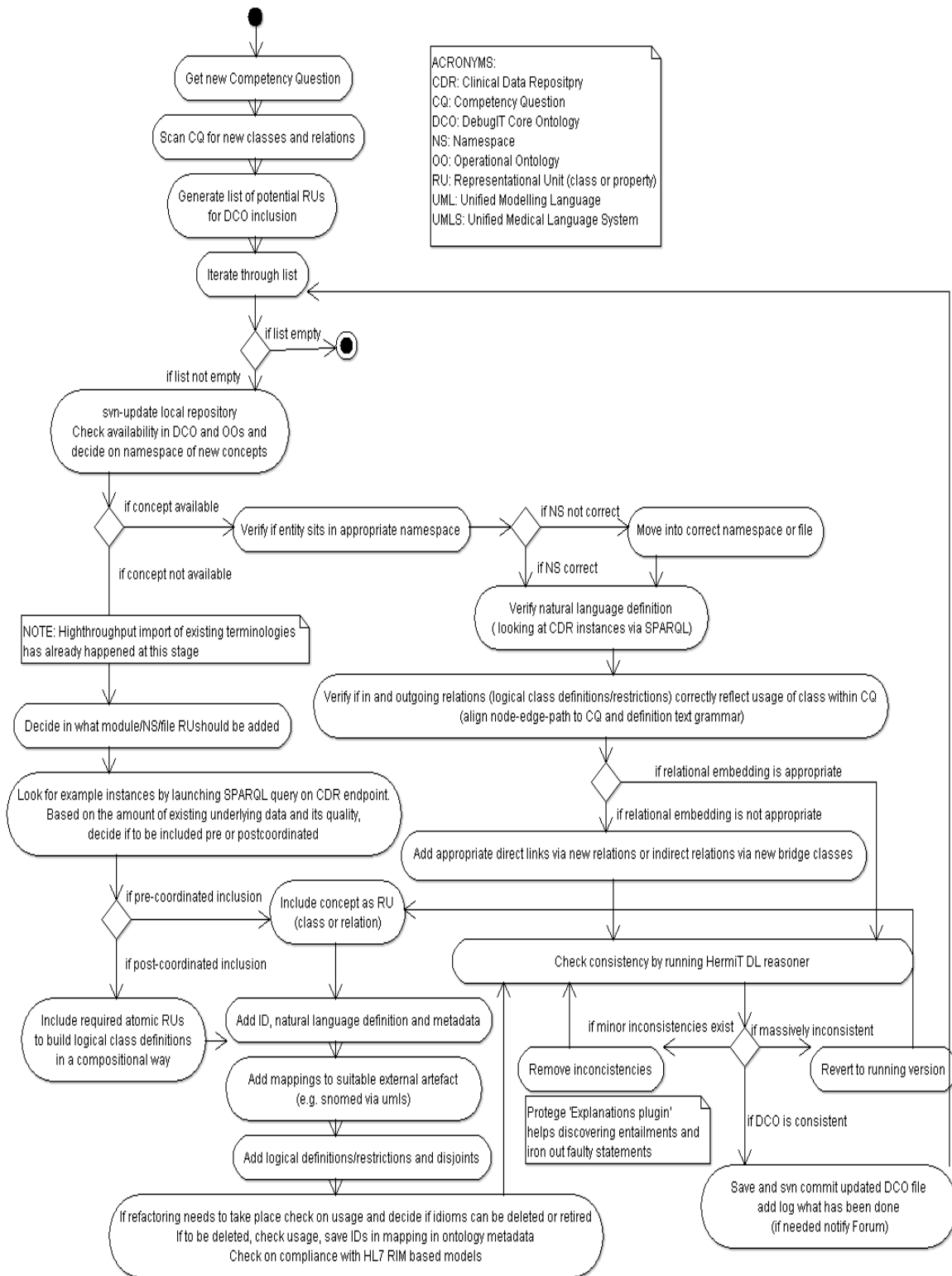


Figure 1: UML activity diagram illustrating DCO engineering upon receiving a new clinical question

mentation is available on the supplementary material website:
<http://www.imbi.uni-freiburg.de/~schober/DCO/>

2.1 Input sources for DCO enrichment

The main input sources for ontology population in the kickoff-phase have been the hospitals CDR schemata, ensuring a data-driven bot-

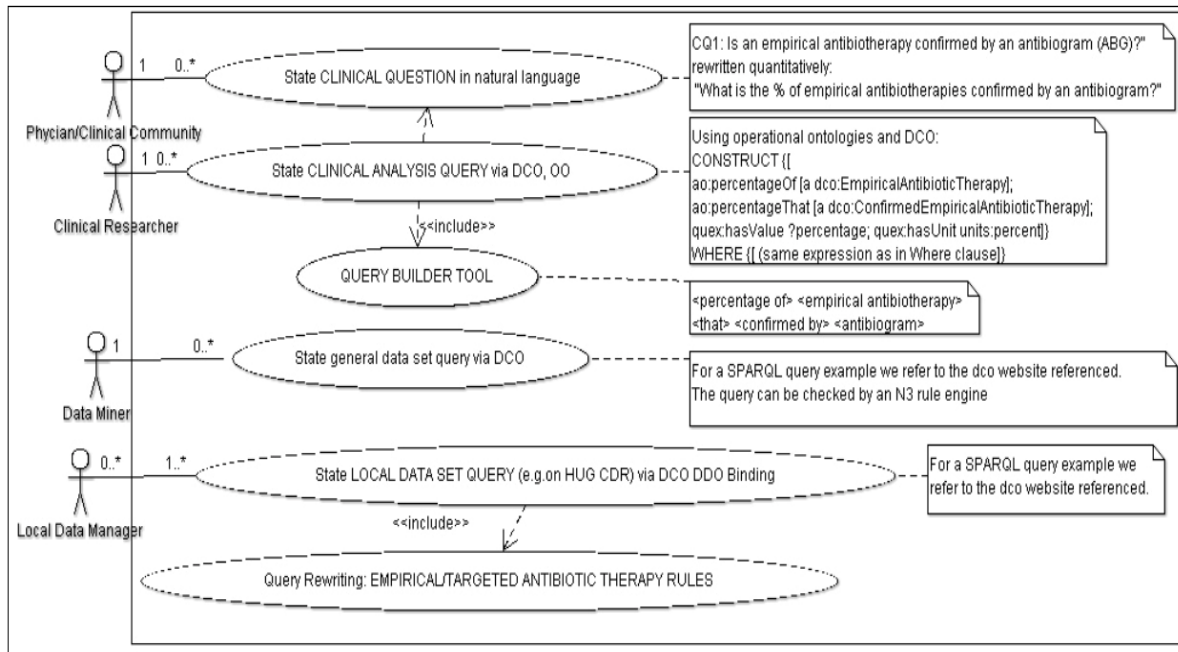


Figure 2: UML Use Case diagram illustrating Ontology usage in different formalisation steps of an example competency question. For SPARQL code examples we refer to the supplementary material.

tom-up enrichment approach. Further sources were competency questions (CQ) and later specific term requests stated by collaborators in a web-forum.

2.2 Competency Questions

To be able to verify whether DCO is sufficiently complete to represent our use case, we have gathered competency questions [4] from clinicians (see Supplementary material). The ontology needs to contain a necessary and sufficient set of axioms to represent these questions, which will serve as benchmark for DCO content coverage evaluation.

2.3 DCO maintenance and evolution

DCO is maintained using a Subversion (SVN) repository² allowing easy detection of work progress exploiting log files and allows for file revision history tracking. Progress monitoring between the ontology work package (WP1a) and the other involved work packages is realized via weekly teleconferences along the SCRUM³ project management methodology. To access the ontology conveniently in a web

browser, we have set up an HTML serialisation⁴.

To illustrate the DCO ontology engineering process in detail, we here present a UML activity diagram illustrating ontology engineering activities upon acceptance of a new competency question (Fig. 1). Additional graphics illustrating the DCO engineering method in the kick-off phase can be found in the supplementary material.

2.4 Information integration via SPARQL

The gap between the different hospitals CDR is bridged by linking RDF models of the various local CDR to DCO concepts in a mapping SPARQL query. In the query process two kinds of ontologies are applied: DCO is used for formulating a hospital independent clinical SPARQL query. In another query formalization step DCO is then mapped to the local CDR via an RDF converted database schema⁵ called data definition ontology (DDO), acting as a query mediator to the proprietary hospital data. This approach is outlined in more detail in [5]. Within the DebugIT interoperability

²<http://www.greeninghealthcare.org/repository/debut/trunk>

³<http://www.scrum.org/scrumguides/>

⁴ http://www.imbi.uni-freiburg.de/~schober/dco_owlDoc/

⁵ E.g. a DDO with a PREFIX insert: <http://debugit1.spim.jussieu.fr/resource/vocab/> as in example query

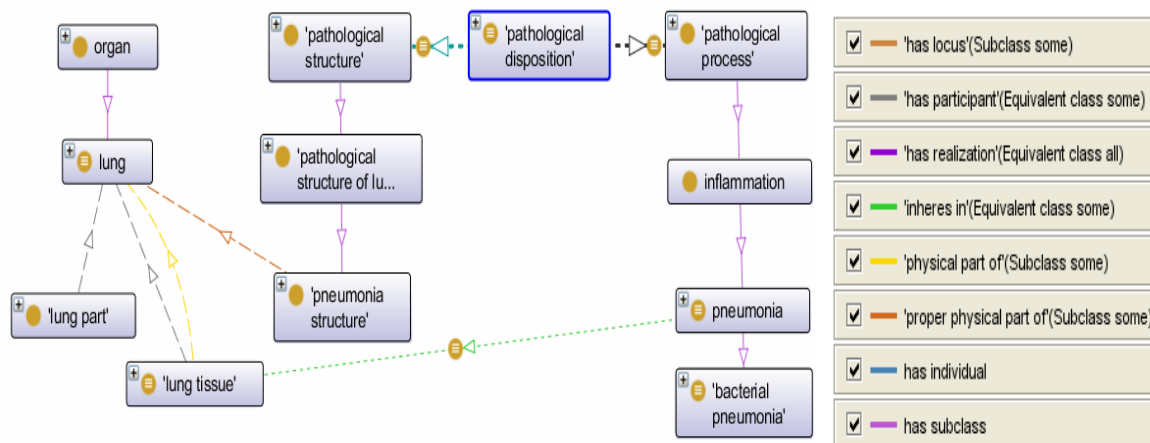


Figure 3: An OntoGraf view on the tripartite ontological dco disease model. A disposition is realized by a disease (Process), which is manifested as a disorder (PathologicalStructure).

platform a clinical query is successively formalized from natural language over semi-formal intermediate query steps towards a formal site dependent local data set query. During this process it is passed from the clinician over to a data miner and further on to the different local data managers. To illustrate how DCO concepts are used within these different query formalisation steps we here include a UML use case diagram (Fig. 2).

3 Results

3.1 DCO current metrics

The current description logic expressivity is SRIF(D). We are using the Hermit DL reasoner⁶, which takes ~2 minutes to classify DCO including BioTop on an average PC. Table 1 illustrates the statistics of DCO and BioTop.

Ontology elements and axioms	Count (all)	DCO	BioTop
Classes	1281	965	375
Object Properties (relations)	78	3	74
Datatype Properties	11	10	0
Subclass Axioms	1494	1050	444
Equivalent Class Axioms	197	98	99
Disjoint Axioms	76	1	75

Table 1: Content and size of DCO and its Biotop upper level ontology

3.2 An ontological model of infectious diseases

A knowledge domain of great importance for the DebugIT project, but also for the wider

healthcare domain is a granular and expressive disease model, i.e. distinguishing pathological processes and agents from pathological structures and dispositions (Fig. 3). We started modeling a prototypical infectious disease, Pneumonia together with some of its key-aspects in a simple pre-coordinated way:

$Pneumonia \equiv Inflammation \sqcap \exists \text{has-participant. LungTissue}$
 $AcutePneumonia \equiv Pneumonia \sqcap \exists \text{bearer-of. AcuteQuality}$

$BacterialPneumonia \equiv Pneumonia \sqcap \exists \text{has-agent. BacteriaPopulation}$

$ViralPneumonia \equiv Pneumonia \sqcap \exists \text{has-agent. VirusPopulation}$

The above however misses some aspects, e.g. it allows for a Pneumonia in the kidney, because *LungTissue* and *KidneyTissue* have not been made disjoint. We also can't specify whether an *AcutePneumonia* can have, besides *Acute*, further qualities, e.g. *Chronic*, or whether the **has-agent** in *BacterialPneumonia* can also be filled by, e.g. *VirusPopulation*. We therefore needed to provide SubclassOf definitions for inferring e.g.

$BacterialPneumonia \sqsubseteq BacterialInflammation$

Mereotopological axioms were needed in order to infer from

$Pneumonia \equiv Inflammation \sqcap \exists \text{has-participant. LungTissue}$

and

$LungTissue \sqsubseteq \exists \text{part-of. Lung}$

that

$Pneumonia \sqsubseteq \exists \text{has-locus. Lung}$

Disjoints like $Process \sqsubseteq \neg Structure$ were added to be able to infer that

$PathologicalProcess \sqsubseteq \neg PathologicalStructure$

We amended DCO successively, providing restrictions for post-coordinations, e.g. con-

⁶ <http://hermit-reasoner.com/>

straining a user to enter new Pneumonias only in Loci where lung tissue exists:

$Pneumonia \sqsubseteq \forall \text{has-locus}. (\exists \text{locus-of}. LungTissue)$

By this and exploiting the following restrictions

$LungTissue \sqsubseteq \exists \text{has-locus}. Lung$

$Lung \sqsubseteq \exists \text{has-locus}. Thorax$

$\exists \text{has-locus}. Thorax \sqsubseteq \neg \exists \text{has-locus}. (Abdomen \sqcup Extremity)$

an ontology-based annotation interface can now provide and guide a user with correct localisations possible for a certain infectious disease [6].

3.3 Maintaining multiple-parenthood by a logics reasoner

From an engineering standpoint, we apply the normalization approach of [7] and use single-asserted parenthood throughout the taxonomy. This facilitates the orientation in the taxonomy and its maintenance. The description logics reasoner Hermit infers multiple parenthood from the formal restrictions. E.g. it enriches the *Sample* class hierarchy by auto-classifying *BodyLiquidSamples*, e.g. from the given facts

$Blood \sqsubseteq BodyLiquid$

$BodyLiquidSample \equiv Sample \sqcap \exists \text{derivesFrom}. BodyLiquid$

$BloodSample \equiv Sample \sqcap \exists \text{derivesFrom}. Blood$

the reasoner infers that

$BloodSample \sqsubseteq BodyLiquidSample$

enriching the taxonomy. This is also done for all other liquid samples.

3.4 Graphical visualisations

To allow biomedical experts not acquainted with ontology editors to view, understand and check parts of the ontology, we apply ontology visualizations as generated by the OwlPropViz⁷ and OntoGraf⁸ Protégé plugins (Fig.3) which enable visual, parallel and hence faster perception of the term networks.

3.5 Constraint natural languages

To allow biomedical experts not acquainted with description logics to view, understand and check parts of the ontology, we investigate highly enduser compliant ways to present ontological statements via Constrained Natural Languages (CNL) like Attempto Controlled

English (ACE)⁹. This creates natural language text that can be used to present ontology fragments to domain experts and makes enduser verification of complex DL statements possible by the non-ontology expert. E.g. it renders the Manchester OWL Syntax

```
InfectiousDisease
EquivalentTo
  biotop:AcquiredPathologicalState
  and (biotop:hasAgent some
    (biotop:Organism
      and (biotop:bearerOf some InfectorRole)))
```

into the following ACE sentence:

“Every InfectiousDisease is an AcquiredPathologicalState that hasAgent an Organism that bearerOf an InfectorRole. Every AcquiredPathologicalState that hasAgent an Organism that bearerOf an InfectorRole is an InfectiousDisease.”

4 Discussion

We have reported on the development of a clinical ontology for data integration and annotation. Although a certain level of semantic integration has been reached, many steps must be performed manually and hence are error-prone as well as time and resource intensive. Regarding the issue to what an extend the ontology should contain pre-coordinated expressions, creating restrictions for guiding users in post-coordinative class generation enforces a transition from OWL 2 EL towards RL expressivity, because disjoints and universal restriction constructors need to be applied. Reasoners used to prevent redundant post-coordinations need to be fast, which is still rarely the case. Traditional large scale RL ontology reasoning is slow and might not be feasible for post-coordination at annotation time when a large set of constraints needs to be verified timely. Here, fast local, incremental reasoning methods need to be investigated.

Some ontologically difficult areas were the modeling of time, e.g. introducing TimeQuality classes versus using simple xsd:dateTime datatype properties; how to model intervals such as episode of care or patient stay; process modifications like adapted or merely planned therapies also depend on a rigid time model. We tried to find a pragmatic compromise between needed complexity and ease of use of time related expressions. Time constructs exploitable by a reasoner were only included

⁷<http://protegewiki.stanford.edu/index.php/OwlPropViz>

⁸<http://protegewiki.stanford.edu/wiki/OntoGraf>

⁹<http://attempto.ifi.uzh.ch/aceview/>

when not making expressions overly difficult to read and create for a human user.

Regarding ontology evaluation, CNLs are not yet in a stage where they can contribute to a better understanding of more complex and especially nested DL expressions. Some expressions, annoyingly the more interesting ‘hub-node’ ones, could not be transcribed and, e.g. the above example should have generated the text

“Every InfectiousDisease is an AcquiredPathological-State that **has as an agent** an Organism **that is** the bearerOf an InfectorRole”

to be intuitive. Further it needs to be investigated how large ontologies can be sub-structured into small digestible parts or modules that can be timely managed by domain specialists.

5 Conclusion

We believe to have created a robust and scalable disease model that can serve the wider biomedical domain. Hence, a next step will be the submission of the above disease definitions as a content ontology design pattern, e.g. towards the OntologyDesignPattern.org repository [8]. Further such micro-models will follow, e.g. for modeling drugs and their prescriptions.

Whereas earlier attempts integrating CDRs via purely syntactical means fail to exploit computer interpretable formal semantics [9], projects begin to appear that show the usefulness and even feasibility of applying owl-DL semantics in healthcare data integration settings. The [LinkedLifeData](http://LinkedLifeData.org)¹⁰, a platform for semantic data integration through RDF warehousing demonstrates how efficient reasoning can help to resolve conflicts within the data. However, such goal, and this is also an important lesson learned in the DebugIT endeavor, can only be achieved if particular care is taken on reasoning performance. Logics-based reasoning will only be feasible in realistically large ontologies when computationally expensive owl-RL constructs are applied consciously. Ultimately the fast-paced progress in semantic web technologies leads to frequent changes in even the most basic tools, such as APIs, reasoners and SPARQL endpoint software. Due to this inherent dynamics one should constantly check where one can restrict one-self to a more robust subset of cutting-edge techniques.

References

- [1] Lovis C et al. **DebugIT for patient safety - improving the treatment with antibiotics through multimedia data mining of heterogeneous clinical data.** *Stud Health Technol Inform.* 136 (2008), 641-6
- [2] Noy NF, Crubezy M, Fergerson RW, Knublauch H, Tu SW, Vendetti J, Musen MA: **Protege-2000: An Open-source Ontology-development and Knowledge-acquisition Environment.** *Proc AMIA Symp* 2003:953.
- [3] Elena Beisswanger, Stefan Schulz, Holger Stenzhorn, and Udo Hahn. **BioTop: An upper domain ontology for the life sciences – a description of its current structure, contents, and interfaces to OBO ontologies.** *Applied Ontology*, 3(4):202–212, 2008.
- [4] Grueninger, M and Fox, M (1994). **The role of competency questions in enterprise engineering.** In *IFIP WG 5.7, Workshop Benchmarking. Theory and Practice*, Trondheim/Norway.
- [5] Daniel Schober, Martin Boeker, Jessica Bullenkamp, Csaba Huszka, Kristof Depraetere, Douglas Teodoro, Nadia Nadah, Remy Choquet, Christel Daniel, Stefan Schulz, **The DebugIT Core Ontology: semantic integration of antibiotics resistance patterns,** *Proceedings MEDINFO 2010*
- [6] Stefan Schulz, Daniel Schober, Djamila Raufie, Martin Boeker, **Pre- and Postcoordination in Biomedical Ontologies,** OBML 2010 Workshop Proceedings, IMISE-Report Nr 2/2010, ISSN 1610-7233, Universität Leipzig, 2010
- [7] Rector AL, Rogers JE, Zanstra PE, Van Der Haring E: **OpenGALEN: open source medical terminology and tools.** *AMIA Annu Symp Proc* 2003:982.
- [8] V. Presutti and A. Gangemi. **Content ontology design patterns as practical building blocks for web ontologies.** In *Proceedings of ER2008.* Barcelona, Spain, 2008.
- [9] Brailer, D. (2005). **Interoperability: The Key to the Future Health Care System,** *Health Affairs (The Policy J. of the Health Sphere)*, Vol. 10 (January), 19-21.

¹⁰ <http://linkedlifedata.com/>