An Integrated Method using Conceptual Modelling to generate an Ontology-based Querying Mechanism

Ronéll Alberts[1] and Enrico Franconi[2]

[1] Universidad Politécnica de Madrid and CSIR Meraka Institute
ralberts@csir.co.za
[2] Free University of Bozen-Bolzano
franconi@inf.unibz.it

Abstract. The application of ontologies in information systems is becoming increasingly prevalent. In particular, the use of ontologies as a conceptual view for the purpose of accessing information stored in underlying data repositories can increase the effectiveness of mechanisms for information querying and browsing. However, the process of ontology construction requires expertise in DLs, which is a scarce skill. An opportunity has been identified to capitalise on the advantages of ontology-based querying while reducing the challenges encountered in ontology construction, by utilising conceptual modelling techniques more prevalent in information systems development teams. This can be achieved by using traditional conceptual modelling techniques to develop a conceptual model of the domain of interest and utilising this to generate an ontology-based query mechanism. In this paper, an overview of an integrated method that uses conceptual modelling to generate an ontology-based query mechanism is presented. The described method utilises existing tools, technologies and approaches. The challenges regarding integration are addressed resulting in an end-to-end method. The developed method was evaluated and refined by applying it in a real-world domain of Indigenous Knowledge, in particular to the themes of African Traditional Medicine and Food Security. The resulting method can be utilised by any organisation interested in building an ontology-based query mechanism over information in a specific domain.

1 Introduction

Ontologies are being applied in information systems at an increasing rate. In particular, recent work towards using ontologies as conceptual views over underlying data stores for the purposes of data integration and access has created opportunities for increased effectiveness of information querying and browsing. By employing ontologies during the process of query formulation the user can be guided to formulate precise, satisfiable queries without an in-depth knowledge of the structure of the ontology or the underlying information repository [1]. During the query answering process, the application of ontologies provide the ability to utilise reasoning services to increase the accuracy of the results by compensating for missing data and returning results based on inference over the knowledge base rather than merely pattern matching and lexical services [2].

Effective ontology construction requires a multi-disciplinary team with expertise in various disciplines. Therefore, effective communication and collaboration between the different role players are crucial. This can be supported through visualisation and verbalisation of techniques. Although visualisation tools to graphically represent OWL ontologies are available (e.g. OWLviz and OntoGraf), the graphical support of the most popular tools for ontology construction, e.g. Protégé and SWOOP, is limited and rely strongly on proficiency in DLs. Expertise in DL syntax and semantics is a specialised skill not widely available in standard information system development teams.

The use of conceptual modelling have been around since the 1970s and is widely utilised in information systems projects for various purposes such as domain modelling, database design, technical architecture, technical design and business process analysis [3]. Different conceptual modelling techniques have been developed for specific purposes such as Entity Relationship (ER) models for data modelling, Object Role Modelling (ORM) for domain modelling and Unified Modelling Language (UML) for object-oriented modelling. The skills of constructing conceptual models in ER, UML and ORM are currently more prevalent in information systems development teams than that of knowledge engineers with the required skills for ontology construction.

An opportunity has been identified to take advantage of the effectiveness of ontology-based querying while eliminating the associated challenges of ontology construction by using traditional conceptual modelling techniques that are often used in information systems projects. The aim of this paper is to provide an overview of a usable, repeatable, integrated method using conceptual modelling to generate an ontology-based query mechanism. The method identifies and utilises appropriate existing tools and techniques while all integration issues are addressed in order to produce an integrated, end-to-end method. This paper is based on the work performed for the degree of European Master in Computational Logics and the method is described in detail in the thesis [4].

The paper is structured as follows: Existing work related to the aspects of the method is shortly described in Section 2. The method and how it was applied in a real-world domain is presented in Section 3 and each aspect of the method is described in Sections 3.1 to 3.6. The method is discusses in Section 4 and the paper is concluded in Section 5.

2 Related work

The method entails the definition of a conceptual model of the domain of interest using traditional conceptual modelling techniques. This conceptual model is translated into an ontology that will formally reflect the knowledge modelled in the conceptual model. The resulting ontology is linked to underlying data repositories and utilised by an ontology-based querying mechanism enabling querying of information from the data repositories through the ontology. Since the construction of the method depended on existing work in various fields of research a short overview of each applicable field is presented in the following paragraphs.

Conceptual modelling: The discipline of conceptual modelling has been applied in computer science since the 1970s. The purpose of a conceptual model is to enable clear communication between all the role players involved in the definition and implementation of an information system [5]. Since this study focuses on the modelling of the information in a specific domain, the approaches towards information modelling are of particular interest. Information modelling techniques include Entity-Relationship (ER) modelling [6], the Unified Modelling Language (UML) [7] and Object Role Modelling (ORM). ORM is a fact oriented approach to conceptual modelling and models the domain of interest in terms of facts that are true in the domain. The facts are described in terms of objects (entities or values) that play roles (parts in relationships) in the domain. All facts in the domain, including attributes associated with an object, are expressed as n-ary fact types [8]. Due to its role-based notation, ORM can explicitly specify a wide variety of constraints. Unlike ORM, UML and ER notations do not support the explicit definition of all possible constraints. Also, UML and ER do not support verbalisation into natural language and their representations are far removed from the original natural language the domain was originally described in [9].

Conceptual model to OWL translation: Ongoing research is conducted to enable reasoning over conceptual models. The main objectives of enabling reasoning over conceptual models are (1) to verify the model in terms of consistency and satisfiability; and (2) to explicitly derive constraints that were implicitly modelled. Existing approaches and technologies developed to add reasoning services over conceptual models can be grouped into three classes: (1) Building specialised reasoners to reason over traditional conceptual models; (2) Building conceptual modelling tools that have built-in reasoning services; and; (3) Translating the conceptual models built with traditional conceptual modelling tools into DLs and OWL, utilising existing reasoning related tools available for DLs and OWL. Approaches to translate ER models into DLs include: ERONTO [10] and mapping rules that capture the semantics of ER and translate it into OWL Lite [11]. Examples of translating UML to OWL include approaches utilising eXtensible Stylesheet Language Transformations (XSLT) to generate an OWL representation from a UML model [12]; a translation algorithm to extract OWL ontologies from UML class diagrams [13]; and the encoding of UML class diagrams in the DL $\mathcal{ALCQI}$ [14]. Recent research in the field of translating *ORM* into OWL includes the definition of a formal syntax and semantics for ORM2 conceptual models including a formal mapping from the formal ORM2 syntax, into $\mathcal{FOL}$, $\mathcal{DLR}$ and $\mathcal{ALCQI}$ DLs [15] [16].

Linking data repositories to OWL ontologies: Ontologies describe knowledge about a particular domain of interest. Since information regarding the domain of interest is often captured and stored in data repositories, linking the two creates opportunities to exploit the advantages of ontologies when accessing the information. Two approaches were identified in the field of linking ontologies to data repositories. They are: (1) Creating a dynamic mapping between the ontology and the data repository e.g. D2RQ [17], VisAVis [18], the work of Xu et al. [19], DB2OWL [20] and OBDA [21] [22]; and; (2) Loading the information from a data repository into the ABox of the ontology e.g. DataMaster [23] or using technical building blocks (OWL API [24] and the Jena API [25]) to develop custom made applications to load the data from a data repository into the ABox of an ontology.

Ontology-based querying: The potential of enhancing querying mechanisms with semantics has been recognised from the inception of the Semantic Web [26]. The approaches to ontology-based querying were classified into three categories [27]: (1) Augmenting keyword searching; (2) Ontology-based multifacet searching; and (3) Native ontology-based searching by utilising the reasoning services available through the use of ontologies during query formulation and query answering processes. Examples of native ontology-based querying include: GoPubMed [28], Textpresso [29], Wonder [30] and Quelo [1].

3 Method design and application

This section describes the design of the integrated method to use conceptual modelling for the generation of an ontology-based querying mechanism over data repositories and its application in a real-world domain. The method was applied in the application domain of Indigenous Knowledge (IK) for evaluation purposes, in particular in the themes of African Traditional Medicine (ATM) and Food Security (FS). The design of the method was adjusted according to challenges encountered and lessons learned during the application of the method in the domain.

The method comprises of six (6) distinct steps. Each step is described in more detail in the following subsections. The method is schematically depicted in Figure 1.

3.1 Conceptual modelling

The aim of the conceptual modelling step of the method is to systematic construct a conceptual model of the domain of interest in a vocabulary familiar to domain experts. The notation to be used is Object Role Modelling (ORM) [9].

The modelling tool used is the Natural ORM Architect (NORMA) tool [31]. NORMA is an open source plug-in to Visual Studio .NET that enables the construction of formal conceptual models using the rich graphical notation of ORM [32]. It also provides a feature where the captured facts in the conceptual model are verbalised in natural language for verification purposes.

For the application in the IK domain, a conceptual model was constructed in ORM using the NORMA modelling tool through collaboration with domain experts. The complete conceptual model for the IK domain is described in [4].

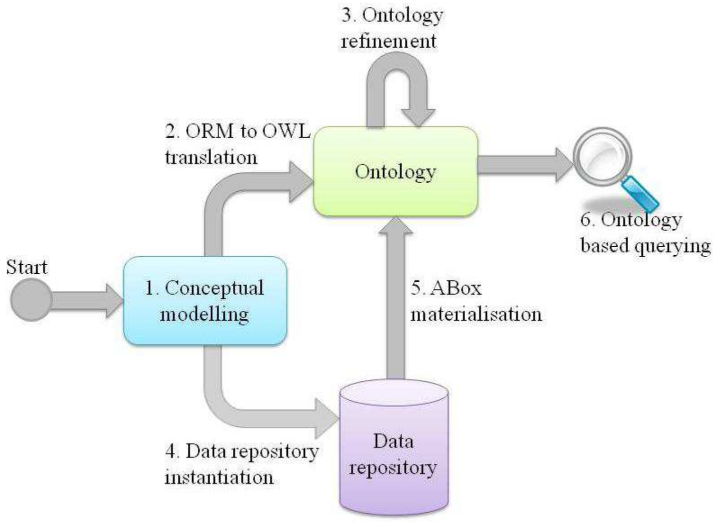

Fig. 1. Integrated method using conceptual modelling to generate an ontology-based querying mechanism over data repositories.

A challenge encountered in this step of the method was that in the querying step it was found that for the construction of useful queries over the resulting ontology, it is required to define the inverse for every defined role. Since it is only possible to define the inverse of binary roles, it was necessary to rewrite all n-ary fact types in the conceptual model, where $n > 2$, into binary fact types.

3.2 ORM to OWL translation

The aim of the translation step of the method is to translate the conceptual model developed in ORM using the NORMA tool into an OWL ontology that is recognisable and understandable to domain experts for the purpose of ontology-based querying. Therefore, the translation procedure must (1) be able to translate an ORM model from the internal XML representation of the NORMA tool to a usable OWL ontology; (2) preserve the intended semantics of the original ORM model; and; (3) preserve the structure of the original conceptual model in terms of concepts and properties to ensure the usability of the ontology for the purposes of ontology-based querying.

The translation approach selected for the method is that of Franconi et al. [15]. This approach provides a formal syntax and semantics for the ORM2 modelling language. It specifies a formal procedure to transform a conceptual model, defined in the formal ORM2 syntax into $\mathcal{FOL}$, $\mathcal{DLR}$ and $\mathcal{ALCQI}$ DLs. An application was developed that implements the defined transformations [33]. Detail on the defined transformations is available in [16].

However, the selected approach requires the conceptual model to be described in the formal ORM2 syntax as input. Since the original conceptual model is developed using the NORMA tool, which stores it in a proprietary XML format, this is not the case. Also, the defined translation initially did not preserve the structure of the original ORM model. This was due to the required reification of n-ary relations. Due to this, a two-staged translation procedure had to be devised. For the first stage a translation process (NormaToMath) had to be developed in XSLT to transform the conceptual model from the internal NORMA XML format into the formal ORM2 syntax. For the second stage, the ORM2OWL translation tool [33], based on the transformations defined by Franconi et al., was used to translate the ORM conceptual model from the formal ORM2 syntax into an OWL2 ontology.

During the application of the method in the IK domain a number challenges were encountered. They were: (1) The primitive entities in an ORM conceptual model are disjoint by definition. This could not be added to the translation procedure since it would produce a non-monotonic behaviour which is undesirable. To address this, the initial disjointness had to be added manually during refinement. (2) The reification of binary relations during the translation procedure initially produced an OWL ontology that was not usable for the purposes of ontology-based querying since the resulting ontology was not understandable to domain experts. This had to be addressed through the following: (a) The object properties and data properties that were defined in the original ORM conceptual model had to be renamed by hand to their original names. (b) The reification process added a number of new concepts and properties to the ontology that was not in the original ORM conceptual model. These should not be visible to the user in the querying mechanism. For this purpose, the translation application was adjusted to add two new concepts named `_ATOP1_` and `_ATOP2_`. All the concepts from the original ORM conceptual model are asserted as subclasses under `_ATOP1_`, while all the additional concepts added through reification are asserted as subclasses of `_ATOP2_`. `_ATOP1_` is also asserted as disjoint from `_ATOP2_`. The domain and range of the properties from the original ORM conceptual model are explicitly asserted. This prevented the properties introduced through reification to be visible when working with the concepts under `_ATOP1_`.

3.3 Ontology refinement

The aim of the ontology refinement step of the method is to refine the ontology resulting from the translation step. It consists of compulsory adjustments and optional refinements. The compulsory adjustments are changes to the ontology that are required for the correct functioning of final querying mechanism. The compulsory adjustments are: (1) Make the originally primitive concepts disjoint since the default disjointness is not implemented in the translation process. (2) Rename original object properties and data properties back to their original names since it is lost during reification in the translation process. The optional refinements are changes to the ontology that will enhance the operation of the resulting query mechanism, but are not required for the correct operation thereof. Examples of such refinements include adding finer classifications by adding levels to the subsumption trees in the ontology; creating new super classes by adding new concepts as parent concepts and moving existing concepts under it as subclasses; and; importing other related ontologies into the ontology. The refinement tasks can be performed using any ontology editing tool such as Protégé or SWOOP.

3.4 Data repository instantiation

The aim of the data repository instantiation step of the method is to create a data repository based on the original conceptual model defined in ORM. This step is optional in the case where there already exist one or more data stores for the domain of interest. The NORMA tool used in the initial conceptual modelling step of the method includes the functionality to generate other implementation artefacts from the defined ORM model such as data repository definition artefacts for popular Database Management Systems (RDBMS) such as DB2, MySQL, Oracle, PostgreSQL and Microsoft SQL Server. The data repository can be created using the creation script generated by the NORMA tool and populated with domain data.

For the application of the method in the IK domain, a data repository was created in PostgeSQL and populated with information extracted from a book on medicinal plants of South Africa. The populated database consists of 150 different species of plants with 190 recorded uses.

3.5 ABox materialisation

The aim of the ABox materialisation step in the method is to load the information from the domain data repositories into the ABox of the ontology. The procedure to follow is: (1) Create database views if the information to be used for materialising the ABox is sourced from multiple data repositories or if the structure of the data repository is vastly different from that of the ontology. (2) Develop an ABox materialisation application in Java using the applicable JDBC API for the RDBMS of the source data repository(ies) to connect to the database and the Jena or OWL API to connect to the ontology and populate the ABox of the ontology with the data from the data repository(ies). The ABox materialisation application will create a stable, populated ontology that can be used as a data warehouse for the purposes of ontology-based querying.

For the application of the method in the IK domain, an ABox materialisation application was developed in Java using the Jena API to load the information from the data repository in PostgreSQL into the ABox of the ontology.

3.6 Ontology-based querying

The aim of the ontology-based querying step of the method is to enable end-users to construct queries and receive results over the information from the data repositories by employing reasoning services over the ontology. This can be achieved by applying any ontology-based querying or exploring mechanism to the resulting ontology. This is feasible due to the fact that the resulting ontology is presented in OWL and the structure of the ontology closely reflects the domain of interest as defined in the original ORM conceptual model.

Any ontology-based querying mechanism can be used for this step. The ontology driven query interface, Quelo [1], was selected for the ontology-based querying step of the method. The selection of the Quelo tool was based on the following: (1) it runs on any well-formed, satisfiable ontology; (2) it supports the end-user to formulate a query without any specific knowledge of the vocabulary used in the ontology or the underlying organisation of the data; (3) it provides visual access to underlying reasoning services in order to build precise, satisfiable queries, logically consistent with the context and defined constraints; (4) it allows users to explore the domain of interest by providing suggestions of legal formulations based on the underlying ontology; and; (5) it provides the ability of intentional navigation in order to explore the domain of interest and the related information in the data repository(ies). The Quelo tool can be accessed on-line at the following URL: http://krdbapp.inf.unibz.it:8080/quelo/, or it can be implemented as a Web application on a local Tomcat server.

Figure 2 depicts a query constructed in Quelo over the IK ontology and underlying data repository to return all plants of which the aerial parts are used in medicine that treats ailments of the respiratory system. In the data repository, the parts of the plant are captured on the lowest level (e.g. "stem" or "root") and the ailments are also captured as specific diseases (e.g. "Asthma", "Influenza", etc.). Through the refinements on the ontology and the use of reasoning services, this query can be answered through the ontology-based query mechanism.

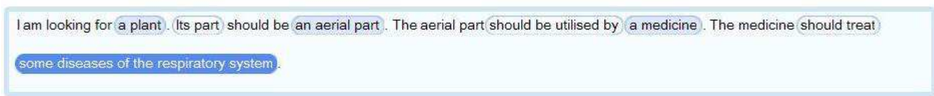

Fig. 2. Query example: Plants of which the aerial parts are used to treat diseases of the respiratory system

The following challenges were encountered when the IK ontology was exposed through the Quelo tool for the purposes of ontology-based querying: (1) At the inception of this study, the functionality of the Quelo tool was limited to a facility to explore the underlying ontology through interactive query formulation. The Quelo tool had to be extended to include the facility to submit the constructed query over the ABox of the underlying ontology and return the results. (2) The ontology resulting from the translation process had two top branches of concepts: `_ATOP1_`, containing all the concepts and properties that were defined in the original ORM conceptual model and `_ATOP2_`, containing all the additional concepts and properties introduced through the translation process. Both these concepts are available to the end-user in the querying tool. The user must deliberately select the `_ATOP1_` branch in order to navigate through the familiar concepts and properties of the domain of interest as defined in the original conceptual model. The concepts under the `_ATOP2_` are not understandable and should not be visible to the user. (3) The Quelo tool currently does not support data properties and is only able to return the URI's of the resulting concepts when a query is executed. The results are therefore not very readable. The ABox materialisation process had to be adjusted to include the description of the concept in the URI during population to improve the readability of the results returned by the Quelo tool. (4) The Quelo tool uses the inverse of properties in the query formulation process. For this reason it is required to define the inverse of all the object properties in the ontology.

4 Discussion

The method described in Section 3 provides a step-by-step approach to use conceptual modelling to generate an ontology-based querying mechanism. The requirements for the integration of the different techniques and tools utilised in the different steps of the method have been successfully addressed and provides an integrated, end-to-end method. The method has successfully been implemented in a real-world domain and can be repeated in the same manner on any domain of interest.

A number of limitations were identified in the current design of the method. They are: (1) The NormaToMath XSLT translation script developed to translate the ORM conceptual model from the internal NORMA XML format into the formal ORM2 syntax only transforms a subset of the full complement of constructs available in ORM. (2) The tool that translates the conceptual model from the formal ORM2 syntax to OWL2 currently reifies all n-ary relationships including binary relationships. While this is unavoidable for n-ary relationships with $n > 2$, this is not the case for binary relationships. If the translation tool is updated to preserve the binary relationships in the original conceptual model and only reify the n-ary relationships where $n > 2$, the challenges associated with the reification can be reduced significantly. (3) The current approach to load the information from the data repository into the ABox of the ontology creates a static link between the two. By creating a dynamic link between the ontology and the data repository, a more robust system will be produced where the information being queried is always current. (4) Due to the structure of the ontology after the translation process, the user is presented with the choice of the concepts `ATOP1` and `ATOP2` in the querying tool. The `ATOP2` concept and all the concepts and properties under it that was introduced into the ontology through the reification process should be hidden to the end-user. (5) Currently the Quelo tool only returns the URI's of concepts as a result when the constructed query is executed. This result still needs to be interpreted by the end user to be of use. In addition, knowledge in the ontology in terms of data properties and associated values is not currently available for exploration and querying though the Quelo tool. If the Quelo tool is extended to include data properties in the query construction procedure and to transform the results into natural language, the usefulness of the results and the coverage of the exploration facility will be greatly improved.

5 Conclusions

This paper described a usable, integrated method using conceptual modelling to generate an ontology-based query mechanism over information in a domain of interest. The method consists of 6 distinct steps. (1) Conceptual modelling; (2) ORM to OWL translation; (3) Ontology refinement; (4) Data repository instantiation; (5) ABox materialisation; and (6) Ontology-based querying.

The underlying technologies and approaches required have been investigated, selected and implemented. All challenges regarding the integration of the different technologies into an integrated method have been addressed. The resulting method was evaluated and refined by applying it in the real-world domain of Indigenous Knowledge. The design of the method was adjusted according to challenges encountered and lessons learned during the application of the method in the domain.

A number of limitations were identified in the resulting method. Future work will be to address the identified limitations by (1) extending the translation mechanism to translate the conceptual model from the internal format of the NORMA tool into the formal ORM2 syntax to support all possible ORM constructs; (2) investigating the update to the translation approach from the formal ORM2 syntax into OWL2 to preserve the binary relations in the original conceptual model and only apply reification to n-are relations where $n > 2$; (3) investigating to replace materialising the ABox of the ontology with information from the data repository with a dynamic link between the ontology and the data repository; (4) extending the Quelo tool to support value properties in the query formulation process and to return the results of the query as natural language; and (5) applying the method to more domains and refining it further. A more extensive evaluation of the method is also needed.

The method presented in this paper can be utilised by any organisation interested in building an ontology-based query mechanism over information in a specific domain.

References

1. Franconi, E., Guagliardo, P., Trevisan, M.: An intelligent query interface based on ontology navigation. In: Workshop on Visual Interfaces to the Social and Semantic Web (VISSW2010), IUI2010. (2010)
2. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rosati, R.: Linking data to ontologies: The description logic DL-LiteA. In: Proc. of the 2nd Int. Workshop on OWL: Experiences and Directions (OWLED 2006), volume 216 of CEUR Electronic Workshop Proceedings. (2006)
3. Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S.: How do practitioners use conceptual modeling in practice? Data and Knowledge Engineering **58** (2006) 358 – 380

4. Alberts, R.: An integrated method using conceptual modelling to generate an ontology-based query mechanism. Master's thesis, Facultad de Informática, Universidad Politécnica de Madrid (2012)
5. ter Hofstede, A., Proper, H., van der Weide, T.P.: Formal definition of a conceptual language for the description and manipulation of information models. Information Systems **18(7)** (1993)
6. Chen, P.P.S.: The entity-relationship model - toward a unified view of data. ACM Trans. Database Syst. **1** (1976) 9–36
7. Fowler, M.: UML distilled. Addison-Wesley (2004)
8. Halpin, T.: ORM 2. In Meersman, R., Tari, Z., Herrero, P., eds.: On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops. Volume 3762. Springer, Berlin, Heidelberg (2005) 676–687
9. Halpin, T., Morgan, A.J., Morgan, T.: Information modeling and relational databases. Second. Morgan Kaufmann (2008)
10. Upadhyaya, S.: Eronto: a tool for extracting ontologies from extended ER diagrams. In: Proceedings of the 2005 ACM symposium on Applied computing. (2005)
11. Myroshnichenko, I., Murphy, M.: Mapping ER schemas to OWL ontologies. In: IEEE International Conference on Semantic Computing, 2009. ICSC '09. (2009)
12. Na, H.S., Cho, O.H., Lim, J.E.: A method for building domain ontologies based on the transformation of UML models. In: Fourth International Conference on Software Engineering Research, Management and Applications. (2006)
13. Xu, Z., Ni, Y., He, W., Lin, L., Yan, Q.: Automatic extraction of OWL ontologies from UML class diagrams: a semantics-preserving approach. World Wide Web (2011) 1–29 10.1007/s11280-011-0147-z.
14. Berardi, D., Calvanese, D., Giacomo, G.D.: Reasoning on UML class diagrams. Artificial Intelligence **168** (2005) 70–118
15. Franconi, E., Mosca, A., Solomakhin, D.: ORM2: Syntax and semantics. Internal report, KRDB Research Centre for Knowledge and Data, Free University of Bozen-Bolzano (2011)
16. Franconi, E., Mosca, A., Solomakhin, D.: ORM2 encoding into Description Logics. In: 2012 International Description Logics workshop (DL-2012), Rome, Italy, June 2012. (2012)
17. Bizer, C., Seaborne, A.: D2RQ: Treating non-RDF databases as virtual RDF graphs. In: Proceedings of the 3rd International Semantic Web Conference (ISWC2004). (2004)
18. Konstantinou, N., Spanos, D., Chalas, M., Solidakis, E., Mitrou, N.: VisAVis: An approach to an intermediate layer between ontologies and relational database contents. In: Proceedings of the International Workshop on Web Information Systems Modeling. (2006)
19. Xu, Z., Zhang, S., Dong, Y.: Mapping between relational database schema and OWL ontology for deep annotation. In: Proceedings of the 2006 IEEE/WIC/ACM International Conferenceon Web Intelligence. (2006)
20. Ghawi, R., Cullot, N.: Database-to-ontology mapping generation for semantic interoperability. In: VLDB 2007 International Conference on Very Large Data Bases. (2007)
21. Rodriguez-Muro, M., Lubyte, L., Calvanese, D.: Realizing ontology based data access: A plug-in for protege. In: Proceedings of the Workshop on Information Integration Methods, Architectures, and Systems (IIMAS 2008), IEEE Computer Society Press, pp. 286289. (2008)

22. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., Rosati, R.: Ontologies and databases: The DL-lite approach. Lecture Notes in Computer Science, Semantic Technologies for Informations Systems **5689** (2009) 255–356
23. Nyulas, C., O'Connor, M.: DataMaster: a plug-in for importing schemas and data from relational databases into protege. In: Proceedings of the 10th International Prot ege Conference. (2007)
24. Horridge, M., Bechhofer, S.: The OWL API: A Java API for working with OWL 2 ontologies. In: Proceedings of the workshop on OWL: Experiences and Directions. (2009)
25. McBride, B.: Jena: a semantic web toolkit. Internet Computing, IEEE **6 Issue: 6** (2002) 55–59
26. Berners-Lee, T.: Semantic web roadmap. Internal note, World Wide Web Consortium. http://www.w3.org/DesignIssues/Semantic.html (1998)
27. Hoang, H.H., Tjoa, A.M.: The state of the art of ontology-based query systems: A comparison of existing approaches. In: Proc. of ICOCI06. (2006)
28. Doms, A., Schroeder, M.: GoPubMed: exploring PubMed with the gene ontology. Nucleic Acids Research **33** (2005) W783 – W786
29. Mller, H., Kenny, E., Sternberg, P.: Textpresso: An ontology-based information retrieval and extraction system for biological literature. PLoS Biology **2(11)** (2004) e309
30. Calvanese, D., Keet, C.M., Nutt, W., Rodríguez-Muro, M., Stefanoni, G.: Web-based graphical querying of databases through an ontology: the wonder system. In: Proceedings of the 2010 ACM Symposium on Applied Computing. SAC '10, New York, NY, USA, ACM (2010) 1388–1395
31. Curland, M., Halpin, T.: Model driven development with NORMA. In: 40th Annual Hawaii International Conference on System Sciences, 2007. HICSS 2007, IEEE (2007) 286a–286a
32. Halpin, T. In: Object-Role Modeling version 2 (ORM2). Springer (2009)
33. Solomakhin, D.: Logical formalization of semantic business vocabulary and rules. Master's thesis, Faculty of Infromatics, Vienna University of Technology (2011)