Knowledge Change Management and Analysis for Multi-Disciplinary Engineering Environments

Fajar J. Ekaputra[1] Estefanía Serral[2] Marta Sabou[1] Stefan Biffl[1]

[1)] Vienna University of Technology,
Christian Doppler Laboratory CDL-Flex
Favoritenstrasse 9-11/188, AT 1040 Vienna, Austria
{firstname.lastname}@tuwien.ac.at

[2)] KU Leuven, Dept. of Decision Sciences and Information Management
Naamsestraat 69, 3000 Leuven, Belgium.
Estefania.Serralasensio@kuleuven.be

ABSTRACT

Multi-Disciplinary Engineering (MDEng) environments involve a wide range of models, processes and tools that were not designed to cooperate together. The Ontology-Based Information Integration (OBII) approach has been proposed to address the integration issue within such environments. However, knowledge changes management and analysis (KCMA) process within the environment are not covered within the OBII approach. While the traditional ontology change management approach has been investigated to the general problem, it remains unclear how to use the available solutions within MDEng context. In this paper, we extend the OBII approach to enable the KCMA process. We have identified the main KCMA requirements within MDEng projects and studied the related work of Ontology Change Management to propose a suitable solution, as well as suggesting further works.

Categories and Subject Descriptors

I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods - *Semantic network*; H.5.3 [**Group & Organization Interfaces**] Collaborative computing

Keywords

Knowledge Change Management, Change Analysis, Ontology-Based Information Integration, Multi-Disciplinary Engineering

1. INTRODUCTION

The process of designing complex mechatronic objects such as power plants or steel mills often requires teams of engineers from diverse engineering domains (e.g., mechanical, electrical and software engineering) to work together. As a result, this design process takes place in a multi-disciplinary engineering (MDEng) environment in which experts from various engineering domains and organizations work together towards creating a complex engineering artifact [13]. This environment is highly heterogeneous as it involves a wide range of data models, processes, and tools that were originally not designed to cooperate seamlessly.

An illustrative MDEng setting is the engineering of a power plant. As any large-scale project, the development of the power plant requires the coordinated work of engineers from multiple disciplines, which needs to converge to a high-quality product. This heterogeneous team of experts should be coordinated in such a way that important project-level technical and management constraints are fulfilled (e.g., the mass and dimension constraints of the base plate are not exceeded by individual equipment). Such coordination requires aggregating relevant data across teams from various disciplines but it is hampered by the semantic heterogeneity of the data, with different disciplines using diverse terms to refer to the same entities.

The Ontology-Based Information Integration (OBII) approach has been previously proposed (e.g., in [1, 16]) to integrate data from the heterogeneous sources using Semantic Web technologies. OBII consists of three components: local ontologies (to represent data specific to one engineering discipline – i.e., local data), a common ontology (i.e., represent the aggregation of relevant and related concepts in organizational level, e.g., power plant) and the mapping between these common and local ontologies to enable linking and integration between these heterogeneous data.

However, to the best of our knowledge, OBII does not provide support for knowledge change management and analysis (KCMA), which is an essential and challenging requirement for MDEng environments. In MDEng environments, the used models and data change over time very often due to (1) **changes in the represented domains**, such as the introduction/removal of domain concepts; (2) **changes in the underlying data sources**, such as when new data elements become available and old data elements become obsolete; or (3) **changes in the intended use of the semantic models and data**, such as by changing requirements of the currently supported tools or the design of new tools.

Ontology change management has been investigated as a generic problem to address the dynamic of ontology data and its derivative challenges (e.g., [7, 8, 14, 17]. However, it is not clear how applicable the current solutions are to improve the KCMA support of OBII approach in the context of MDEng. Therefore, the proposed approaches for dealing with ontology change management in general settings must be adapted to fulfill the requirements of MDEng such as change propagation to the overlapped data in other disciplines.

In this paper, we extend the OBII approach to address the specific requirements of KCMA in MDEng environment. We have identified a set of requirements from the environment as and studied the related work on the ontology change management to derive our proposed approach for addressing the problem.

The rest of the paper will be structured as follows: Section 2 will identify key requirements for KCMA within OBII based MDEng solution and summarized the relevant related works. We explain our proposed solution in Section 3 and finally conclude the paper in Section 4 and identify the potential future work.

Table 1 Solution Alternatives for Knowledge Change within MDEng Environment

Requirements	**Related Approaches Details**	***MDEng KCMA***	*Kle04* [7]	*Sto04* [14]	*Noy06* [8]	*Pap09* [10]	*Grö10* [4]	*Zab11* [17]	*Van13* [12]	*Hor13* [6]	*Gra14* [3]
Multiple Linked Ontologies	*Single Ontology*				x	x	x	x			x
	Several, Loosely coupled ontologies		x	x					x	x	
	*Several Closely coupled ontologies **(1)***	**x**		(x)							
Scale	*> 1M triples **(2)***	**x**				<200K				x	100K
*Knowledge Change Focus **(3)***	*A-Box*	**x**		x		x			x	x	x
	T-Box	**x**	x	x	x	x	x	x		x	
*Change Validation **(4)***	*Manual (User)*	**x**		x						x	
	Automatic	**x**		(x)				x			
*Change Detection **(5)***	*Low-level*	**x**			x	x	x		x	x	x
	High-level	**x**				x	x				
*Ontology Change **(6)***	*Ontology Evolution*	**x**	x	x	x	x	x	x		x	
	Ontology Versioning	**x**		(x)					x	(x)	x

2. REQUIREMENTS & RELATED WORK

At the level of actual MDEng environment data, industrial partners need to keep data versions, move backwards to previous versions, and query **different versions** of **large data** coming from the heterogeneous local data sources. Furthermore, in multidisciplinary MDEng environments the effective and considerate propagation of changes is essential to ensure **a consistent view** of the project, to minimize defects and risks, and acceptance of new solutions with domain experts. To achieve this, the changes coming from one discipline need to be communicated and coordinated with the participants of other disciplines, where those changes are relevant (**closely linked ontologies**), while striving to provide the **high-level changes** definition (e.g., defined in terms of domain concepts such as "Motor X updated to new version") instead of low-level changes (i.e., at levels up individual change operations on the versioned files) to ease the analysis process of the data.

Next, we will identify the key requirements to support KCMA within OBII based MDEng solution based on our interviews with domain experts and our experience in handling knowledge in such environment. Furthermore, we summarized the relevant related works of knowledge / ontology change management from SW community (see summary in Table 1; number on the requirements explanation correlates with the number in the table).

(1) Closely Coupled Ontologies. In the OBII based MDEng, we are dealing with KCMA in a closely linked ontologies environment, where local ontology changes (both of axioms and facts) might affect and change other ontologies via change propagation. This is not the typical setting for KCMA in Semantic Web community, where they are dealing with open Web data. This difference reflected within most of traditional KCMA that focused on multiple ontologies [6, 7, 11, 12]. Stojanovic [14] provide an exception to this trend, where she provided an attempt to propagate changes to relevant ontologies. However, the work is not continued and not further developed.

(2) Large Amount of Data. An average power plant's engineering design data is ranging between several hundreds thousand and tens of millions of signals. Those numbers, combined with the hundreds of process iterations lead to a large number of data to process. Horridge et al. [6] has shown the answer to the large scale challenge of the changed data by introducing the binary formats of store ontology data and version differences, claimed to be working with more than one million triples. A different approach is adopted by Graube et al. [3], where they tried to use named graphs to store changes and ontology versions. Their approach did not scale well for change data analysis, since the query performance on the change data dropped significantly after several thousands of triples. Papavassiliou et al. [10], on the other hand, successfully experimented their approach on almost 200k triples.

(3) Axiom and Facts Changes. Heterogeneity of data sources within MDEng environment also means that additional tools could be added anytime, which may imply changes in the common and other local ontologies. The goal of the KCMA within MDEng environment is to address such changes in the data structure (Axioms) as well as data instances (Facts) to support the stakeholders in analyzing the changed data for their use. Several KCMA approaches are already able to address this requirement [6, 10, 14], and their approach could be used as the basis for the KCMA for MDEng environment.

(4) Automatic-to-Manual Validation Shift. Given the mission critical nature of the project in the MDEng, the domain experts and engineers do not want to totally rely on the automatic change validation mechanism based on the constraint definition, (e.g., to decide whether changes initiated by a local ontology will break the global data consistency). They wanted to involve the domain experts in the validation workflow, in order to make critical decision about changes and how to proceed with it.

Stojanovic et al. [15] have provided a mechanism to involve domain experts to check the semantic validity of ontology change over multiple ontologies. One interesting line of work came recently that could be applied in the change validation, which involved using crowdsourcing to better structure model coming from automatic ontology engineering [5].

(5) High-Level Change Definition and Detection. In the typical tools used within the power plant design, they are able to produce report data that consists of signal list that represent the atomic parts of a factory handled by specific tools. Difference between two versions of signal lists represents the changes between them. However, it is challenging for a project manager to grasp the meaning with such low-level changes data. They need the data to be presented in a more meaningful manner as high-level changes, in terms of domain level common concepts.

Papavassiliou et al. [10] shown an example on how to derive such high-level changes from low-level changes without compromising performance. Alternatively, Gröner et al. [4] had shown the usage of a subset of OWL-DL reasoning to recognize high-level changes pattern. The goal of this requirement is to provide stakeholders with a better decision support system w.r.t. KCMA in OBII based MDEng approach.

(6) Ontology Change. Flouris et al. [2] has provided an excellent definition of ontology evolution, defined as "*a process of modifying an ontology in response to a certain change in the domain or its conceptualization*" and ontology versioning, defined as "*an ability to handle an evolving ontology by creating and managing different variants/versions of this ontology*".

Most of the ontology change management approaches focus on one of them, e.g., ontology evolution [4, 7, 8, 17] or ontology versioning [3, 12], while the others are trying to address both of them [6, 14]. In the context of our work, these approaches would become a good basis for our solution approach.

3. PROPOSED SOLUTION

In order to address the challenge of providing support for Knowledge Change Management and Analysis (KCMA) in the Ontology Based Information Integration (OBII) approach within Multi-Disciplinary Engineering (MDEng) environment, we extend the OBII approach [1, 16] as shown in Figure 2. We have added four additional phases (shown as white boxes in the figure), which are derived from the related works and available standards in ontology change management and related fields from the Semantic Web community. We utilize IDEF-0[1] diagram to structure the proposed approach, in which processes shown as boxes, and resources are shown as directed arrows.

There are three domain experts involved in the framework: Knowledge Engineer (KE), Project Manager (PM) and Domain Expert (DE). The framework draws on several standards and technologies, (e.g., SPARQL for querying, PROV-O) which will be used for structuring and implementing our approach. Input and output of the system is shown in the left and right side of the diagram respectively.

The main stages of the proposed approach are:

(1) Local Ontologies Definition. This phase requires the Knowledge Engineer and Domain Experts to work together to translate the local tools data structure (e.g., MCAD model for mechanical engineer) to the local ontology axioms definition.

(2) Common Ontology & Mapping Definition. KE and DE will be working together in this phase to define the common ontology and its mappings to the local ontologies. To support this goal, Semantic Web vocabularies and standards are required to formalize the ontology and mapping. There are several approaches, e.g., SPARQL or SPIN[2], which could be used to define the mapping and transformation rules within our context.

(3) Local Ontologies ETL. With regards to the heterogeneous domain tools and their data formats within the MDEng environment, we need to provide the suitable extract, transform, and load (ETL) functions phase to produce the data in the required ontology formats. Several solutions could be re-used to address this problem, e.g., Apache Jena[3] and R2RDF[4].

(4) Change Definition and Detection. This phase focuses on the definition and detection of low-level (i.e., triples) and high-level (e.g., semantic and domain-specific) changes between two versions of engineering data. An important point to consider within this phase is to balance the expressivity of high-level changes definition and the computational complexity of the detection algorithm, as mentioned by Papavassiliou et al [10]. Generic open source Ontology APIs (e.g., Apache Jena, Sesame API) typically provides mechanisms for detection of low-level (triple) changes between two ontology versions. Additionally, research results, e.g., PROMPTDIFF [9] and the high-level changes definition approach from Papavassiliou et al [10], could be used to further enhance the detection algorithm. These approaches will be used as a basis for our work to address change definition and detection in MDEng environment.

(5) Change Validation. The phase of change validation requires definition of constraints for preserving the validity of data in the local (e.g., mechanical engineering) and global scope (e.g., power plant). Workflow definition is another important element, in order to configure involvement of validation components (e.g., constraint validation engine and domain experts) in the validation process. To formulate the constraints, recently, there is an initiative called Shapes Constraint Language (SHACL[5]) W3C working group, which aims to provide the constraint standard vocabulary for RDF graph data.

(6) Change Propagation. Changes in the MDEng environment need to be propagated to the relevant components (i.e., common ontologies and other relevant local ontologies). The phase requires the common ontology and mapping definitions, as well as the validated changes. Knowledge engineer will need to configure the propagation based on the mapping definitions (e.g., based on SPIN or SWRL[6] rules), to make sure that no corrupted or irrelevant data is included in the propagation process.

(7) Data Store and Analysis. The goal of this phase is to enable relevant stakeholders (e.g., project manager) to access and analyze the data and its changes within the projects. The changes data will be stored within RDF triple stores, e.g., Sesame[7]. We are planning to utilize the W3C standard PROV-O[8] vocabulary for storing the change provenance information. Examples of queries that would be made on this change data are: (1) Provenance information of the changes (e.g., committer, date, reasons of change), (2) Change overview on specific objects, and (3) Analysis of completeness and inconsistencies over changes.

4. CONCLUSION & FUTURE WORK

We have extended the OBII approach, mainly created for the purpose of data integration, to properly address the challenge of KCMA within MDEng environment. We have identified key requirements as well as studied the related state of the art from the ontology change management area. This work is meant to lay the foundation towards a solution for providing a fully functional KCMA solution for OBII-based MDEng domain.

[1] http://www.idef.com/idef0.htm
[2] http://goo.gl/TcTB8R
[3] http://jena.apache.org/
[4] http://www.w3.org/TR/r2rml/
[5] https://w3c.github.io/data-shapes/shacl/
[6] http://www.w3.org/Submission/SWRL/
[7] http://rdf4j.org/
[8] http://www.w3.org/TR/prov-o/

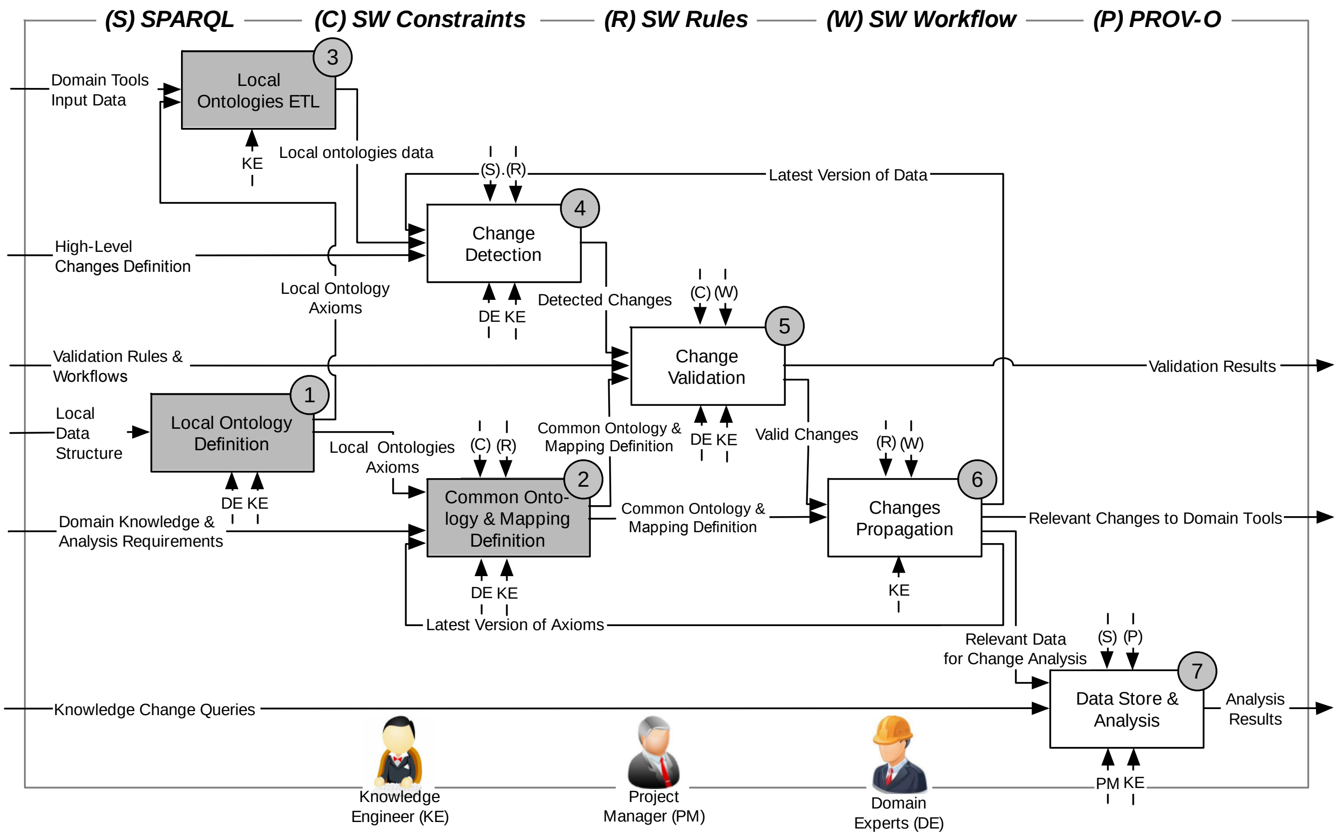

Figure 1 Extended OBII approach to address KCMA in the MDEng environment

In the process of investigating a suitable extension, we found out that there is a gap in the standardization of several aspects of Semantic Web, e.g., constraint and rules vocabulary, which could hinder further adoption of semantic web in the context of MDEng domains, e.g., Industrial Automation System, and make it difficult the use of the proposed extension. Fortunately, there are already initiatives towards standardization of these vocabularies, e.g., SHACL working group for RDF graph constraint and RML Mapping Language for semantic mapping.

As future work, we will develop the prototype implementation based on our proposed OBII extension framework, as well as conduct evaluations of the approach. We are also planning to generalize the approach to address similar problem settings, such as in scholarly data management.

5. ACKNOWLEDGMENTS

This work was supported by the Christian Doppler Forschungsgesellschaft, the Federal Ministry of Economy, Family and Youth and Österreichischer Austauschdienst (ÖAD).

6. REFERENCES

[1] Calvanese, D., De Giacomo, G. and Lenzerini, M. 2001. Ontology of Integration and Integration of Ontologies. *International Description Logics Workshop* (2001).

[2] Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D. and Antoniou, G. 2008. Ontology change: Classification and survey. *The Knowledge Engineering Review*. 23, 2 (2008).

[3] Graube, M., Hensel, S. and Urbas, L. 2014. R43ples: Revisions for Triples. *Proceedings of the 1st Workshop on Linked Data Quality* (2014).

[4] Gröner, G., Parreiras, F. and Staab, S. 2010. Semantic recognition of ontology refactoring. *The Semantic Web–ISWC 2010*. (2010).

[5] Hanika, F., Wohlgenannt, G. and Sabou, M. 2014. The uComp Protégé Plugin: Crowdsourcing Enabled Ontology Engineering. *Knowledge Engineering and Knowledge Management* (2014).

[6] Horridge, M., Redmond, T., Tudorache, T. and Musen, M. 2013. Binary OWL. *OWL Experiences and Directions Workshop (OWLED)* (2013).

[7] Klein, M. 2004. *Change Management for Distributed Ontologies*. PhD Thesis. Vrije University Amsterdam, Netherlands.

[8] Noy, N., Chugh, A., Liu, W. and Musen, M. 2006. A framework for ontology evolution in collaborative environments. *The Semantic Web-ISWC 2006*. (2006).

[9] Noy, N. and Musen, M. 2002. Promptdiff: A fixed-point algorithm for comparing ontology versions. *AAAI/IAAI*. (2002).

[10] Papavassiliou, V., Flouris, G. and Fundulaki, I. 2009. On Detecting High-Level Changes in RDF/S KBs. *8th International Semantic Web Conference, Chantilly, VA, USA*. (2009).

[11] Redmond, T., Smith, M., Drummond, N. and Tudorache, T. 2008. Managing Change: An Ontology Version Control System. *OWLED*. (2008).

[12] Vander Sande, M., Colpaert, P., Verborgh, R., Coppens, S., Mannens, E. and Van de Walle, R. 2013. R&Wbase: git for triples. *Linked Data on the Web Workshop* (2013).

[13] Serral, E., Mordinyi, R., Kovalenko, O., Winkler, D. and Biffl, S. 2013. Evaluation of Semantic Data Storages for Integrating Heterogenous Disciplines in Automation Systems Engineering. *IECON 2013* (Vienna, Austria, 2013).

[14] Stojanovic, L. 2004. *Methods and tools for ontology evolution*. PhD Thesis. University of Kalsruhe, Germany.

[15] Stojanovic, L. and Maedche, A. 2002. User-driven ontology evolution management. *Knowledge engineering and knowledge management: ontologies and the semantic web*. (2002).

[16] Wache, H., Voegele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H. and Hübner, S. 2001. Ontology-based integration of information-a survey of existing approaches. *IJCAI-01 workshop: ontologies and information sharing* (2001), 108–117.

[17] Zablith, F. 2011. *Harvesting Online Ontologies for Ontology Evolution*. PhD Thesis. The Open University, UK.