From English to RDF - A Meta-Modelling Approach for Predictive Maintenance Knowledge Base Design

Ana Milicic, Dimitris Kiritsis, Nesat Efendioglu

From English to RDF - A Meta-Modelling Approach for Predictive Maintenance Knowledge Base Design

Ana Milicic[1], Dimitris Kiritsis[1], Nesat Efendioglu[2]

[1] EPFL, Lausanne, Switzerland
ana.milicic@epfl.ch
[2] BOC Asset Management GmbH, Vienna, Austria

Abstract. In recent years, ontology for the Product Lifecycle Management (PLM) domain has raised a lot of interest in research communities. In this paper we present the contribution in form of meta-model of the entire ontology design process. By implementing USM methodology as ADOxx model the entire procedure is enhanced though automatization of deterministic steps and strategic procedure conduction.

Keywords: Meta-modelling · Ontology design · User story mapping.

1 Introduction

In recent years, ontology for the Product Lifecycle Management (PLM) domain has raised a lot of interest in research communities, both academic and industrial. It has emerged as a convenient method for supporting the concept of closed lifecycle information loop [1], which is one of the most important issues of PLM. By modeling relevant aspects collected from all lifecycle stages of a product, within one ontology, a common knowledge structure is created accessible to all actors. Information availability within the company might seem like a trivial aspect but it is actually one of the most challenging issues. Large companies are divided into number of departments which have only limited communication protocols, employees change projects or leave company and information is generated over a long span of time and locations. In the same time, having the right information at the right moment when decisions about design, production or maintenance are made, is crucial for business success. The complexity of the information source domain translates directly into complexity of design of the ontology which will be mapping all relevant knowledge and the information sources, present in the domain. There is number of different approaches and methodologies for ontology design addressing different problem setups and domain types, but that is out of scope for this work. In this paper, we address the User Story Mapping (USM)method [2] which is designed specifically for creation of ontologies in PLM domain where most of the actors and information sources are not experts in semantic technology.

A USM is an approach that creates a backlog along scenarios and users. It answers the question on how an employee in the company would like to use

knowledge base that is being designed, what are the activities that he performs and which resources he uses. Defining usage by all relevant users, thus defines the domain knowledge, the existing one, needed one and produced one . A backlog consists of several structure blocks as shown in Fig. 1.

1. Usage dimension – It describes how a user would use the knowledge base, for which activities and with what purpose.
2. User dimension – This dimension defines the types of users that will use the knowledge base
3. Backbone – This section describes the activities that a user performs within a usage step. This section is called backbone as it suits as a guideline for the definition of the user stories, which are actually a refinement of the backbone.
4. User stories as backlog items – This is the actual placeholder for the user stories. The user stories are ordered vertically under each activity and represent a refined ver-sion of an activity. It is recommended that user stories follow the pattern "As ¡user¿ I want to ¡feature¿ so that ¡value¿".
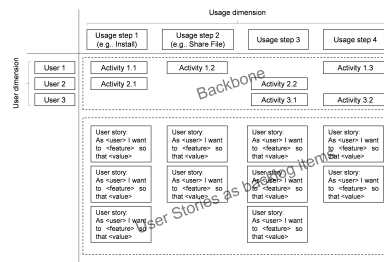

Fig. 1: USM backlog (Source:[2])

USM is the first step of gathering information regarding the domain of interest and the following step can be defined as translating functional needs into list of concepts. This is done manually, by taking into account the whole grasp of the domain,existing templates, resources and standards and by identifying a list of relevant concepts. The process is itterative and done in collaboration between semantic experts and the domain experts until every user story can be expressed though ontology concepts and ontology properties between concepts.

In this paper we present the contribution in form of meta-model of the entire ontology design process. As previously explained, onotlogy design is itterative and can be time consuming esspecially when performed manually. ADOxx meta model enables faster, more controled design proces as well as collaborative environment for distributed teams. Further on, the entire procedure is shown for the use case of predictive maintenance tasks.

2   Meta-Modelling

Conceptual modelling is a technique to "study" the real world – the so-called system under study – and provides concepts to abstract the complexity of the real world and establish a model environment that enables to knowledge processing on models (e.g. simulation, analysis, transformation etc.) of complex systems thanks to the conceptual abstraction.

Meta-modelling is an established approach to specify conceptual modelling languages and a proven technology to enable conceptual integration of information coming from different enterprise levels as well as from different domains. Moreover, meta-modelling makes information available for both machine computation and human interpretation to create value out of that information [3].

As above-mentioned the main objective of conceptual modelling is creation of value out of information stored in the models with applying appropriate knowledge processing algorithms. The ADOxx©is a meta-modelling platform, which enables realization of any kind of modelling method that consists of the conceptual modelling language, algorithms to process models of the modelling language and the procedure that defines the stepwise appropriate usage of the modelling language. Only prerequisite is, that the Generic Modelling Method Specification Framework – GMMSF (shown in Fig. 2) for conceptualization of the method is followed.
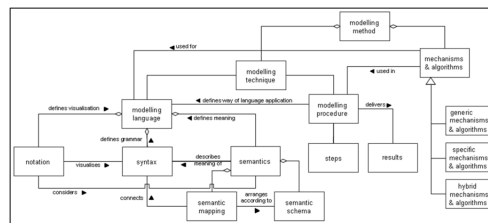

Fig. 2: Generic modelling method specification framework (Source:[4,5])

According to [4] [5] the building blocks of a modelling method include: (1) the modelling language introducing modelling concepts pre-defined according their semantic, their syntax and their graphical notation, (2) the modelling procedure which defines the stepwise usage of the modelling language and may not be always available and (3) generic and domain specific mechanisms and algorithms enabling the computer-based processing of models. By applying this framework (and selecting ADOxx® as a realization platform) it can be guaranteed that defined modelling method can use all of the generic platform functionality (e.g. visualization, publishing, simulation, etc) and can be amalgamated with other modelling methods based on the requirements of the stakeholders (e.g. adding Ontology Modelling to extend User Story Mapping Method) or extended with new concepts.

3 User Story Mapping Modelling Method Realized with Using ADOxx©

The User Story Mapping Modelling Method and its corresponding modelling environment aim to offer a model-based approach supporting User Story Mapping Method (USM Method) during capturing domain specific knowledge, virtualizing and transforming knowledge understandable by humans and machines.

The modelling method shall support all steps of the USM method and it count all steps as challenge to be targeted. Hence, the challenges are (C1) to enable creation of user story map, (C2) to enable gathering other source of information, (C3) enabling creation of unique list of concepts that covers entire domain, (C4) definitions of relations and dependencies among new concepts and concepts in initial knowledge base and (C5) creation of dynamic knowledge base covering domain expressed in standard format, which is machine interpretable.

3.1 Modelling Language

The modelling language of the USM Method consists of two model types; (1) User Story Mapping Model and (2) OWL Diagram. The Fig. 3 depicts the meta-models of both model types and the inheritance from ADOxx©Meta-model.

User Story Mapping Model targets challenge (C1) to enable creation of user story map, hence it consists of the concepts; "User", "User Story", "Activity", "Backlog" and "Group", which have been introduced in the section Fig. 3, and identifies relations among them.

Addition to User Story Mapping Model, we decided to enable modelling ontologies, hence to add so-called the OWL Diagram. The OWL Diagram is sub-set of the OWL Web Ontology Language (W3C, 2004) and we utilized it as it is implemented in (ADOxx.org, 2016). The OWL Diagram targets challenges; (C2) to enable gathering other source of information, (C3) enabling creation of unique list of concepts that covers entire domain, (C4) definitions of relations and dependencies among new concepts and concepts in initial knowledge base and (C5) creation of dynamic knowledge base covering domain expressed in standard format, which is machine interpretable.

The formalized overview of the meta-models defined in FDMM [6], where;

- A meta-model is a tuple $MM = \langle MT, \leqslant, \text{domain, range, card} \rangle$ where MT is the set of the defined model types, i.e. for i=1,...,m we have: $MT = \{MT_1, MT_2, \ldots, MT_m\}$.
- The MT_i's (i=1,..,m) are themselves tuples $MT_i = \langle O_i^T, D_i^T, A_i \rangle$, where:
- O_i^T is the set of object types or classes,
- D_i^T is the set of data types, and
- A_i is the set of the attributes.
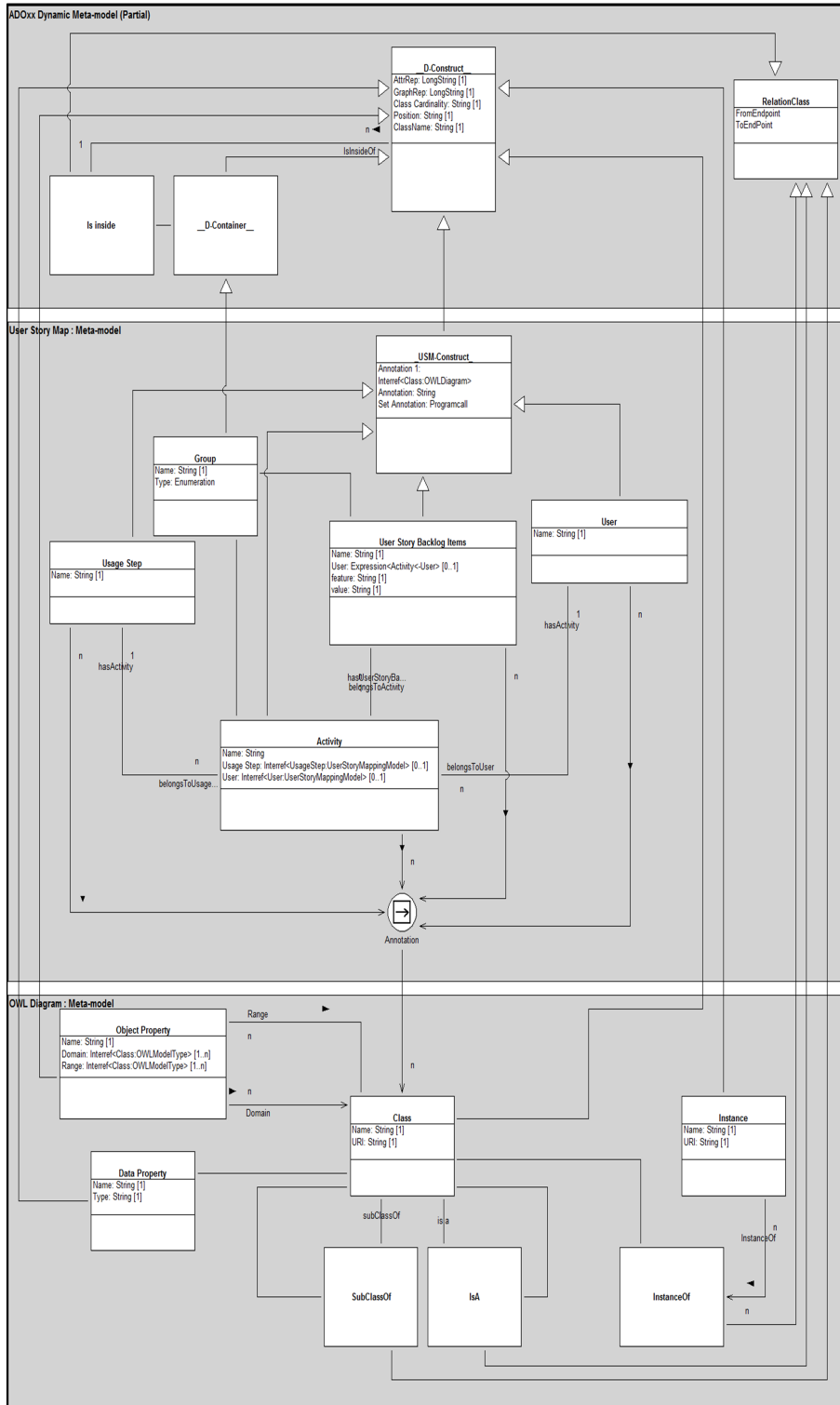- The domain is a function with **domain**: $A \rightarrow P(O^T)$

Fig. 3: Platform specific meta-model of USM modelling method

– The range maps an attribute to the power set of all pairs of classes and model types, all data types, and all model types.
range: $A \rightarrow P(\bigcup_j (O_j^T \times \{MT_j\}) \cup D^T \cup MT)$
– The cardinality function **card :** $O^T \times A \rightarrow P(\mathbb{N}_0^+ \times (\mathbb{N}_0^+ \cup \{\infty\}))$

The formalized overview of the meta-models defined in FDMM [6], where;
The Meta-model of USM Modelling Language is formalized in FDMM as following;

– User Story Map MT_{USM}
– OWL Diagram MT_{OWL}
– $MT_{USM} = \left\langle O_{USM}^T, D_{USM}^T, A_{USM} \right\rangle$
– $MT_{OWL} = \left\langle O_{OWL}^T, D_{OWL}^T, A_{OWL} \right\rangle$

Since we integrated already formalized and implemented OWL Diagram by [7], we describe just the model type User Story Map in details;

– $MT_{USM} = \left\langle O_{USM}^T, D_{USM}^T, A_{USM} \right\rangle$
– $O_{USM}^T = \{\_USM\_Construct\_,\ Group, Usage\ Step,\ User\ Story\ Backlog\ Items,$
– $Activity, User,\ Annotation\}$
– $D_{USM}^T = \{String, ProgramCall,\ Expression, Enum_{Group},\ Interref_{Class},$
– $Interref_{UsageStep},\ Interref_{User}\}$
– $A_{USM} = \{Annotaiton,\ Name,\ Type,\ User,\ Feautre, Value,\ Usage\ Step,\ SetAnnotation\}$

Weaving is a modelling technique where different model types are linked with each other. For challenges (C2) and (C3) we utilize the so-called "Semantic Lifting Approach from [8]. As depicted in, we use weaving technique to enable Semantic Lifting by using Class object types defined in OWL models to annotate the objects modelled in User Story Map models
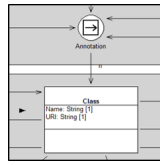

Fig. 4: Weaving in order to use owl diagram into user story map.

The description of weaving depicted in Fig.4 in FDMM is as following;

– **domain**(Annotation)= {_USM_Construct_}
– **range**(Annotation)= {Class}
– **card**(Annotation=$\langle m, n \rangle$) for m,n∈ N

3.2 Mechanisms and Algorithms

Using Mechanism & Algorithms capabilities of ADOxx modelling system, creation of list of ontological concepts is implemented by allowing ADOxx user to annotate Activities from Backbone with concepts from ontology. ADOxx user

is enabled to define his own concepts and sub-concepts. For example, the Activity "Check Business Exploitation Results", needs to be annotated as "Data" → "Digital" → "Resource" → "User Feedback".

These new concepts Fig. 5 can be further used to annotate Activities. By annotating the entire Backbone of activities, list of concepts is created by merging all concepts that were used in annotation process, from those generic to more specific ones. ADOxx model will generate this list automatically and create a visualization.
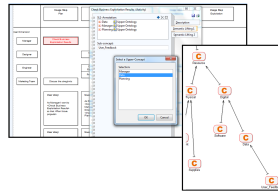

Fig. 5: Addition of new sub-concepts in ADOxx.

The next step in ontology design process is definition of relations between concepts. The challenge with definition of relations (or object properties, according to ontological terminology) is that some level of interaction can be usually found for every two concepts and only a small portion of them is relevant and useful for domain description. That is, it is not the issue of finding relation between two concepts, it is the issue of selecting the ones that are worth defining as object properties. Using ADOxx modelling tool, the recommendation system for this challenge was created, relaying again on Activity Backbone. The reasoning is that if one Activity, was annotated with two or more concepts, then it is reasonable to assume that those two concept should have object property connecting them. The tool is implemented as recommendation system, meaning that the ADOxx user is presented with an option to create object property which he can freely accept or decline Fig. 6.
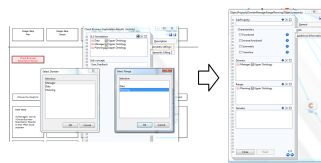

Fig. 6: Object property definition in ADOxx.

4   Application Scenario

Predictive maintenance is a field with ever-growing popularity. Compared to a standard, fixed period maintenance, it saves resources, cost and environment.

The most relevant aspect of predictive maintenance is most certainly its re-liability. Too wide confidence bounds will lead to non-optimal maintenance schedule while the too narrow confidence bounds might lead to costly donw-time and repairs. The key of the solution to this problem is having all relevant information about machine usage conditions, usage patterns as well as machine lifecycle over all. This is especially true for machines which are large invest-ments or when a long down time is not acceptable.

In this paper, we examine the real-life example of company X, which is a producer of healthcare equipment. They specialize in high-end machines such as MRI scanners which are highly complex, sensitive for calibration and usage modes. In the same time, the long or frequent down time is not acceptable since the hospital's patient treatment depends highly on availability of those ma-chines. Today, company has two sources of information about machine usage and calibration that is log-files (stored in the machine and shared periodically) and help desk files (where workers are taking notes of all reported problems with integration with the rest of equipment and usage problems). In the same time, these information are requested by number of actors and with different summarization level requirements. Designers of future products will want to understand the most frequent issues that certain existing line of products ex-hibited, while the service team will want to understand the usage patterns and previous maintenance activities.

In order to aggregate this complex and wide domain, USM method as ADOxx application was applied. In Fig. 7 we show a sample of user activities defined (details of the work had to be omitted due to privacy constraints). Ontology generated based on given backlog is shown in Fig. 8 (again, only upper level concepts in parent-child hierarchy are displayed).

The onotlogy constitutes a knowledge base, where all the data are gath-ered and structured. It enables information retrieval using SPARQL semantic queries as well as integration to existing IT infrastructure of company X. It serves as ground layer for upper level applications preforming data analysis and serving as decision support system.

5   Conclusions and Outlook

By implementing USM methodology as ADOxx model the entire procedure is enhanced though automatization of deterministic steps and strategic procedure conduction. Each step can be performed independently and repeated if needed. Each step is documented and can be revisited or discussed in the future.

Having ADOxx model, opens a field of opportunities for future work that are out of scope of USM as ontology modelling methodology. ADOxx system allows export of models as RDF or XML structures making them available for other software tools. On the other hand, current export mechanism allow ex-port of models while the procedures and algorithms defined on those models are not available outside ADOxx. One potential direction of future work is de-sign of mechanism that will translate existing mechanisms into SWRL syntax.
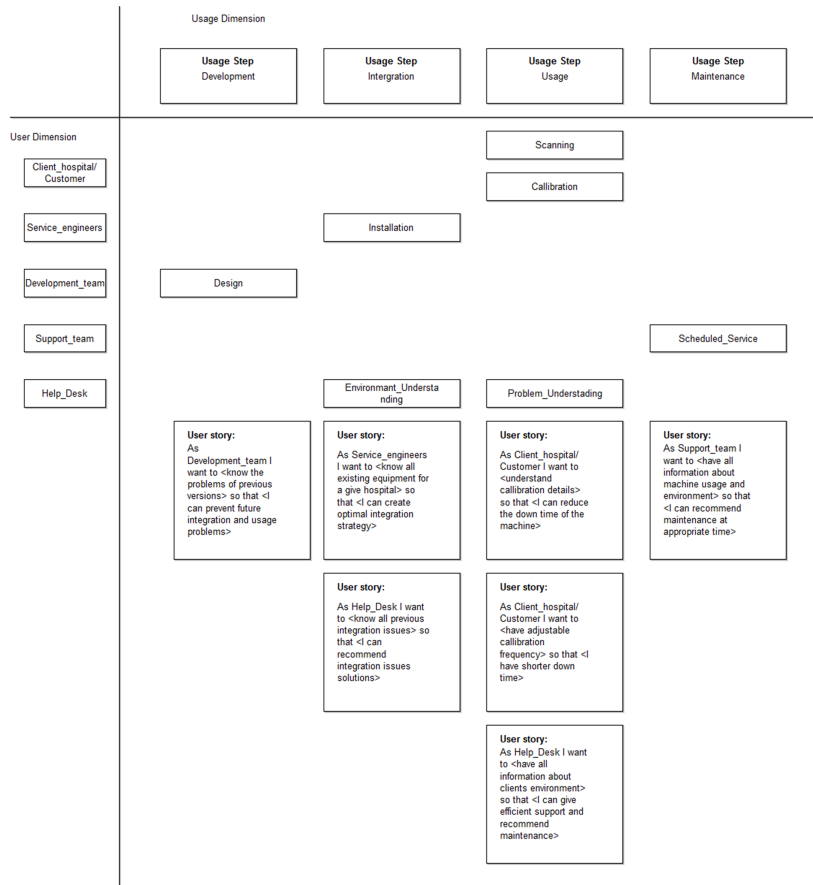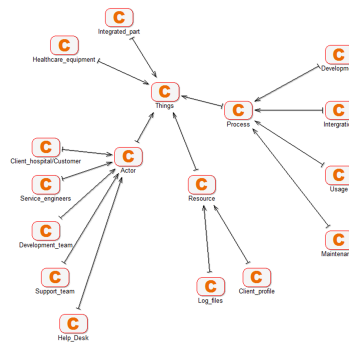
Fig. 7: Sample of user activities

Fig. 8: Upper levels of predictive maintenance ontology

Using this approach, ontology in RDF could be enriched with ontological rules in SWRL giving more complete ADOxx model export and thus, interoperability.

References

1. Jun, H.B., Kiritsis, D., Xirouchakis, P.: Research Issues on Closed-Loop PLM. Computers in industry 58(8), 855–868 (2007)
2. Milicic, A., Perdikakis, A., El Kadiri, S., Kiritsis, D., Ivanov, P.: Towards the Definition of Domain Concepts and Knowledge Through the Application of the User Story Mapping Method. In: IFIP. pp. 58–69. Springer (2012)
3. Woitsch, R.: Hybrid Modeling: An Instrument for Conceptual Interoperability. Revolutionizing Enterprise Interoperability through Scientific Foundations (2014)
4. Karagiannis, D., Kühn, H.: Metamodelling Platforms. In: EC-Web. vol. 2455, p. 182 (2002)
5. Kühn, H.: Methodenintegration im Business Engineering. Ph.D. thesis, University of Vienna, Austria (2004)
6. Fill, H.G., Redmond, T., Karagiannis, D.: FDMM: A Formalism for Describing ADOxx Meta Models and Models (2012)
7. ADOxx Realization Case: Web Ontology Language (OWL), https://www.adoxx.org/live/owl
8. Hrgovcic, V., Karagiannis, D., Woitsch, R.: Conceptual Modeling of the Organisational Aspects for Distributed Applications: The Semantic Lifting Approach. In: COMPSACW, 37th Annual. pp. 145–150. IEEE (2013)