

Semantic Relationship Based Knowledge Management and Reuse in Collaborative Product Development

Ibrahim Assouroko^{1,3}, Guillaume Ducellier²,
Benoît Eynard¹, and Philippe Boutinaud³

¹ Université de Technologie de Compiègne, Department of Mechanical Systems
Engineering CNRS UMR 7337 Roberval

{ibrahim.assouroko, benoit.eynard}@utc.fr

² Université de Technologie de Troyes, UMR CNRS 6279

Institut Charles Delaunay – LASMIS

guillaume.ducellier@utt.fr

³ CADeSIS, 37 rue Adam Ledoux, Bat. B, 92400 Courbevoie, France

{iassouroko, pboutinaud}@cadesis.com

Abstract. Latest technological advances, driven by an increased industrial competitiveness, lead to the development of more complex technical products, which implement technologies from multiple fields of expertise (mechanics, electronics...). The successful design of such kind of products requires the ability of the involved team to communicate, collaborate and integrate their knowledge and know how. This usually causes data integrity problems, making it more difficult to access product information and knowledge distributed in various and heterogeneous systems, for knowledge management and reuse purposes. In such a context of knowledge management, methods and tools for information extraction and visualization play a major role, given their capacity to extract, represent and organize engineering information and knowledge throughout product life-cycle. The paper proposes a semantic relationship based management approach to improve product Beginning Of Life (BOL).

Keywords: Collaborative Product Development, Knowledge Management, Ontology, Semantic Relationship, Data Graph.

1 Introduction

Nowadays, the increasing heterogeneity of CAx (Computer Aided x) applications and Product Lifecycle Management (PLM) systems raises the problem of data and information exchange, in enterprise systems interoperability and knowledge management issue, in collaborative product development context. Being able to implement effective data and information exchange solutions still remains a major stake for product development companies, in their way to maximize the information asset they have and optimize related processes, in order to gain competitiveness by quickly designing and bringing cheaper and higher qualified product on the market. To this end, PLM vendors' systems with adequate methodologies and processes are considered strategic tools to allow collaboration within the product development process, but they still fall

short of providing the full range of desired functionalities. As result of the rapid evolution of digital engineering software supporting engineering activities focused on product-process integration, there exist today, efficient methods and appropriate systems i.e. Computer Aided Design (CAD), Computer Aided Engineering (CAE), Computer Aided Manufacturing (CAM), Simulation Data Management (SDM), Enterprise Resource Planning (ERP) systems...at many phases of the product lifecycle. However, the links from one phase to another (in terms of management of product data of each field of expertise) are often not operational thus their full integration not fully effective [1]. Although the current CAx tools, PDM and PLM systems (reinforced by advances in computer network technologies) have laid the foundation for the emerging fields of collaborative ICT environment in design, the heterogeneity of engineering design knowledge management tools is still a major obstacle in sharing and exchanging design knowledge for multi-disciplinary design activities, across applications borders. In an alternative solution by using visual languages, information and knowledge visualizations have become effective methods for representing complex bodies of knowledge [2], giving that in both areas, human factors play an important role, including ability to navigate, relate, remember and understand complex information, and possibly to collaboratively share and utilize knowledge. In a same perspective, the paper proposes a collaborative semantic environnement - a Relationship Manager (RsM) - for product information and knowledge management and reuse, with a focus on the semantics that exist or may exist between information and knowledge related to product lifecycle, with respect to Semantic Web and graph visualization techniques implementation.

Next section presents and discusses some related works on Semantic Web, Ontology and data visualization issues. Section 3 presents the RsM digital repository concept and related engineering data. Section 4 and 5 present and discuss conceptual, methodological and graph ontology aspects of the proposed approach and software environment. Finally section presents a case study to illustrate the RsM prototype in use, and section 7 summarizes and outlines ongoing and future works on the research.

2 Related Work

Nowadays, the definition of ontology and its application to the product development process has grown significantly. In the field of PLM, various and important works, implementing different approaches to product design and related interoperability issues has been carried out on Ontology. Since the last fifteen years the National Institute of Standards and Technologies (NIST) has worked on a Core Product Model (CPM) [3] to support intelligent design repositories for sharing and reusing the knowledge on design Artifacts, focussing on the aspects of form, function and behaviour. Based on the CPM of the NIST, authors demonstrate in [4] how to implement ontology in the model of an existing product. Moreover, with respect to the CPM and within the Product Development Process (PDP), [5] proposed a Product Design Ontology (PDO) which especially addresses researchers in product design and engineering analysis who need to share shape data and to develop software tools. The authors

actually formalized the task-specific information associated to a shape, and the functionality and usage of shape processing methods in specific tasks of the design workflow.

From product requirement point of view, [6] have developed a requirement meta model that synthesizes and simplifies many relevant concepts included in well-known Requirement Engineering research works. The proposed metamodel (that we use as basis for requirement domain ontology in the RsM approach) stresses the importance of performing requirements management activities as a first-order workflow for web methodologies and paves the way towards considering usability requirements from the first stages of Web Information Systems (WIS) development process.

In the context of this research, semantic interoperability is an important issue that we consider. Semantic interoperability implies the existence of a common and shared understanding of the meaning of the information that is being exchanged; and to achieve perfect semantic interoperability, all communicating systems must use symbols and definitions that are identical or can be accurately translated, thus making a common ontology the ideal solution for semantic interoperability [7]. Ontology is a main component of Semantic Web (SW) in which content can be expressed not only in natural language, but also in a format that can be read and used by software agents, thus permitting them to find, share and integrate information more easily. In addition to the fact that SW allows direct access to data, within applications without needing the integration of the applications themselves [8], they provides a range of services for conducting ontological arguments on conceptual semantics models [9], because of ontologies capacities of reasoning and inference, i.e. their ability to make inferences on explicitly defined instances from classes, properties and axioms of the given ontology [10].

Due to the increasing information overload brought about by the large-scale digitization of information, visualizations have become effective methods for representing and organizing knowledge, and information rich scenarios [11]. Visualizations themselves are tools for knowledge management which make use of the human cognitive processing system in order to create and convey content more efficiently [2]. Information and knowledge visualization both employ similar techniques. "Based on mapping rules, resource objects are translated into visual objects as meaningful representations, offering easy and comprehensive access to the subject matter presented" [12]. On the basis of these elements, authors in [2] define a Visual Wiki to be a combination or integration of two representations of the same underlying body of knowledge - visual and textual - which are sharable and group-editable in a wiki style.

The research methods and approach discussed in this section, such as Ontology, semantic interoperability and concepts related to information and knowledge visualization, lay the basis for the contribution, and addressed the main functionalities implemented through the RsM proposed methodology. The RsM methodology combines semantic interoperability to traditional interoperability approaches to integrate in real time and across applications border, lifecycle information or metadata and knowledge, and provide designers with an encompassing knowledge base in support to design activity.

3 Digital Repository and Engineering Data in Product's Beginning Of Life

The digital repository, in terms of product data storage, combines on the one hand, a set of existing data, information and knowledge dispersed in a galaxy of databases, sheets, files drawings, notes... inside and outside the company [13]; and on the other hand a set of business tools and applications (with related processes and methodologies) handling and managing the company information and knowledge as its business asset. The aim being to allow design project actors to contribute in early stage of the product development and throughout the product lifecycle as well, in order to reduce product development cost by taking earlier the right decision in design activities. Unfortunately, interoperability barriers, reinforced by the growing heterogeneity issues between ICT systems in product development environment, still remain a barrier for an efficient collaborative management and capitalization of product information and knowledge.

The idea of the digital repository resulting from the proposed approach in this research (focused on the product BOL) is to reduce the existing heterogeneity and provide a consistent platform with a quite coherent data flow (during the product design progress and a set of clear traceability between information) avoiding duplication of data, and supporting different aspects of the product which are processed and analyzed separately.

In order to improve the current management of engineering knowledge, we proposed in [14], a category of data that we called semi-structured data (in between structured and unstructured data). The idea is to provide design team members, with data whose structure is initially unfrozen, and that can evolve with the design activity. Such a data (implementing Semantic Web and graph technics) gives the designers the possibility to contribute with their own knowledge, in order to enrich the available data, and share the thus created knowledge assets with other designers for capitalization and reuse purposes.

4 Conceptual Approach for Product Design Semantic Relationship Management

This section presents the conceptual model related to the proposed approach, and the methodological framework that describes main elements and steps in the implementation of the Relationship Manager solution. The model defines the manipulated entities and their instances within the RsM, the classes and objects that contribute in the construction and evolution of the entities, and how they support designer's knowledge capitalization and reuse activities. The methodological framework presents the considered entities, the type of applications and tools to which they are related, as well as processes and possible interactions between previous tools and applications in the way they manipulate and make evolve product data and information with a given set of semantic relationships within the digital repository.

4.1 Conceptual Model

This model (Fig. 1) extends the model we previously proposed in [14]. Entity (E) is the key object of the RsM, and represents any type of design object in the product BOL phases.

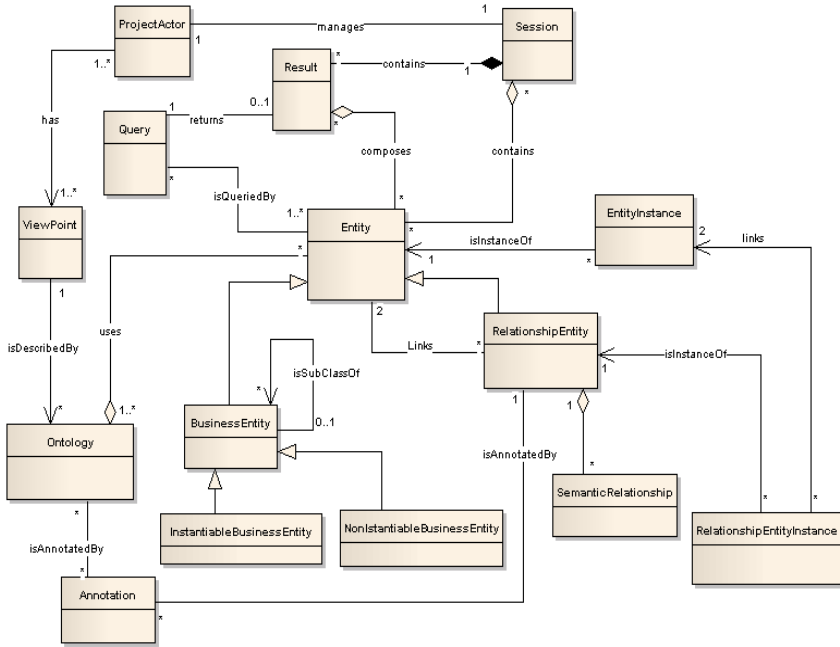


Fig. 1. The RsM extended conceptual model

Entity generalizes both BusinessEntity (BE) and RelationshipEntity (RE); BE represents any metadata and/or document resource stored in existing business tools and/or management system or platform (requirement management applications, PLM, PDM, SDM...), and is mainly described by its URI (Uniform Resource Identifier) – a hypertext link to the corresponding data in the related business tool. RE is the core object of the model, and represents any entity used to link two other Entities. Thus, a RelationshipEntity is not just a predicate or semantic relationship, but can be a triples (BE-predicate-BE or BE-predicate-RE or RE-Predicate-RE), as far as the resulting entity, i.e. a RelationshipEntity, can be reused either in the current Session(S) or by another designer, to create a new RE for knowledge capitalization purposes.

In this version of the model, we introduce an extension related to meta model elements, for the creation of future dedicated ontological models, as support to the BusinessEntity instance (and related semantic relationships) creation, as well as to the RelationshipEntity storage in the RsM database. The meta model is composed of the class Entity, which generalizes BusinessEntity and RelationshipEntity; then we have EntityInstance and RelationshipEntityInstance which instantiate real word data

created by the designer (instances of design entities and related semantic relationships, that the user create during a design Session, and save to the RsM database). Here, we introduce some specificity of BusinessEntity objects which in turn generalizes InstantiableBusinessEntity and NonInstantiableBusinessEntity. This to say that the meta model is implemented in a way that, when creating the set of data instances on a given ontological model, the user is given the possibility to map his entity instances (file, requirement information, design information, meshing file...) only to InstantiableBusinessEntity of the corresponding ontological model. We also introduce inheritance on BusinessEntity, in order to allow some kind of BusinessEntity to be Subclasses of a given BusinessEntity.

5 The RsM Methodology for Product Semantic Relationship Management

The proposed methodology, as part of the RsM approach, is mainly dedicated to the creation and scalable management of semi-structured entities (as explained in Section 3) and to an adaptive enrichment of the entities, to support knowledge capitalization task inherent to the daily activity of people involved in product development.

5.1 The RsM Methodological Framework

The methodological framework defines and explains implementation scenarios of the RsM, the type of manipulated entities and the software environment contributing to the collaborative processes in the RsM.

As shown on Fig. 2, the RsM methodology comprises four scenarios described as following:

- **1** Perform Query (on search engine) on BE and RE. Regarding search possibilities, each Query is performed according to a given ViewPoint (requirement manager, mechanical design engineer, analysis engineer) necessarily associated with a Session. Every Query goes first to the RsMDataBase, and if any Entity is found according to the search criterion, then the Query is passed to the PLMXQuery module for execution on the right Business tool. The PLMXQuery module is combined with a Format Adapter module, to verify that the Query addressed to the Business tool is in the right format, as well as for the query Result, that has to be formatted and display to the user in scenario 2
- **2** Let's consider the case where results have been found for the Query on more than one Business tool: through the PLMXQuery and Format Adapter modules, the results are displayed to the user as graph of data (each graph in blue represents a BusinessEntity graph with internal relationships from the tool)

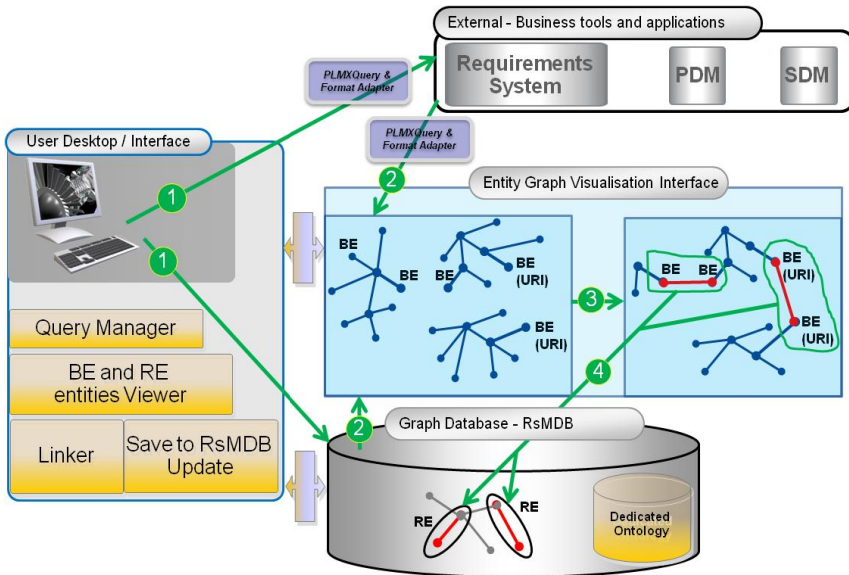


Fig. 2. RsM methodological framework description through scenarios implementation

- 3 Between step (2) and (3), the user can perform some given visualization actions such as navigation, URI edition on BE (in order to access related metadata, information, knowledge...). The user can also perform some semantic enrichment on the entities in the graph, by creating semantic relationships between BE and generating semi-structured entities (new or enriched Relationship Entities) to be stored in the RsMDataBase.
- 4 Once the user have finished visualization and enrichment actions on the displayed graph(s) of entities, the results of the enrichment (only the newly created triples – RE) are saved to the RsMDataBase and mapped to the existing RE in the database. The mapping is also performed based on specific meta modelling rules of the dedicated (Business) ontology implemented within the RsMDataBase. Thus, the definiton and implementation of an ontological model is not restrictive and can be done in adequation with the lifecycle phase and related business rules and activities.

The implementation of the proposed approach is made through Semantic Web (SW) technics and graph concepts, to allow a semantic integration of a set of information and knowledge sources from various digital engineering applications. As one can remark through the previous steps of the RsM approach, Ontology and graph concept play a major role in the Relationship Entity creation and management in a rather semantic interoperability for knowledge capitalisation purposes. Main ideas of the proposed ontology and its application are explained in next section.

5.2 Dedicated Ontology and Data Graph, as Support to Knowledge Capitalization

Ontologies play a key role in the mechanisms of the Relationship Manager for semi-structured data creation and management. The nature of the proposed ontological model facilitates, not only, the handling of the evolving character of the relationship entities, and the handling of their maturity during the design activity; but also supports product BOL engineering knowledge representation and visualization. The basic elements of the ontology proposed in the RSM consists of a set of business concepts with a clear definition of the hierarchy (taxonomy) between concepts, a set of relationships (properties) between concepts, all being supported by a Logic Language (OWL-DL in our case) providing a clear semantics and rather high reasoning capacities, allowing a good level of inference on Relationship Entities in the RsM database. It is important to keep in mind that the definition of dedicated ontological model for a given domain addressed by the RsM, relies on the meta-model or meta-ontology, the core of the model proposed in Section 4.

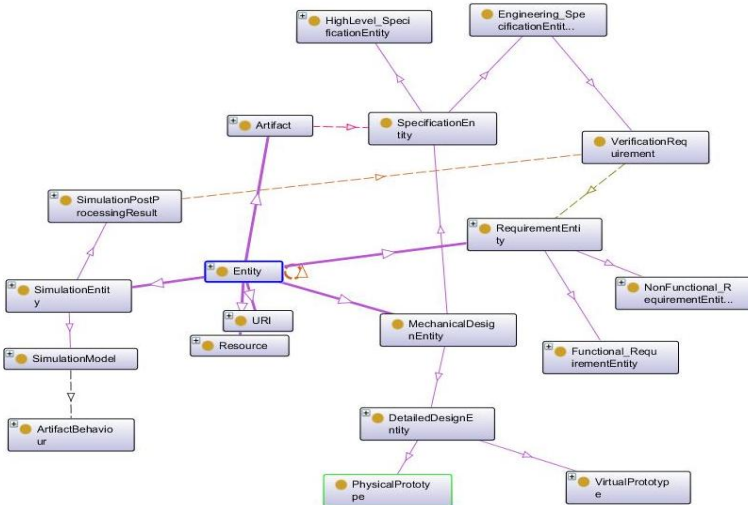


Fig. 3. A view of main entities of the proposed product BOL ontology graph

The main Entities of the proposed Ontology have been formalized with the Protégé software, in an ontological graph illustrated in Fig. 3. With respect to the conceptual model in section 4, the key element of the ontology is the class Entity which has three subclasses RequirementEntity, MechanicalDesignEntity and SimulationEntity. The class Entity defines two basic semantic relationships, *hasURI* and *hasResource*, respectively to URI and Resource concepts; which states that every instance of Entity and subclass of Entity is characterizes by an uri scheme (as unique identifier) and associated with one or more document resource(s). The Artifact class represents any structural entity of the product that can represent an assembly component, a part or a given

assembly of the considered product. An instance of *Artifact* is related to a given instance of *SpecificationEntity* (subclass of *MechanicalDesignEntity*), since the *Artifact* models a real world object that satisfies the instance of *SpecificationEntity*, itself deriving from a *Functional_RequirementEntity*. On the other hand, any instance of *Artifact* has a behaviour simulated by an instance of *SimulationModel* which *Simulation-PostProcessingResult* (subclass of *SimulationModel*) satisfies an instance of a *VerificationRequirement* systematically associated with a corresponding instance of *Engineering_SpecificationEntity* (subclass of *SpecificationEntity*).

5.3 The RsM Capitalization Graph as Support to Knowledge Management

In the RsM approach, the capitalization graph in respect to a given entity stored in the RsM database, is the resulting graph from a set of enrichment actions (creation of instances of *RelationshipEntity*, semantic annotations...) containing Business Entities, Relationship Entities and the set of uri that identify metadata/information and knowledge, related to the given entity. This graph provides not only a global and quite rich view on the concerned entity at a given time, but also capitalizes all the information and knowledge (from heterogeneous sources) in relationship with the entity, thus facilitating traceability of the entity. The capitalization graph also gives the opportunity to the user to reuse the available body of knowledge to perform needed design task, or to extract elements for performance or maturity evaluation in a decision making process on the concerned design entity.

According to reasoning rules and axioms expressed in the ontological models (on the concepts, their organization in the model and the relationships they implement in the RsMDataBase), inferences can be performed on entities composing a capitalization graph within the RsM database, i.e.: automatic generation of new *RelationshipEntity* with respect to a design Session, thus enriching the content of the RsMDataBase and offering new possibilities of results to designers Queries, in the way they wouldn't have thought. In addition, the capitalization graph structure defined according to the proposed ontological meta modelling rules (combined with entity instances mapping rules within the RsM database and implemented query algorithms) facilitates the graph traversal and analysis, during query execution to get the right instances of entities to be displayed to the designer.

6 The RsM in Use

This section gives an insight of the RsM in use and describes main elements of the software environment of the RsM prototype, developed to test and validate the proposed approach and methodology. The RsM user Desktop or Graphical User Interface contains a search engine interface to perform Queries on BE and RE, and an integrated entity graph visualization interface to display search results and allow metadata/information navigation in order to access related knowledge. Queries can be performed either in natural language (*find all Simulation model that simulates the behavior of the Artifact "Crankshaft_xyz"*) or using well-known semantic or concept name implemented in the ontological model of the RsM (*find PostAnalysis_SimulationModel*

of “Crankshaft_xyz”). The proposed user interface for entity graph visualization and enrichment, and the RsM database for Relationship Entities storage and mapping have been developed within Neo4j, a graph database tool that can store design data in the nodes and relationships of a graph [<http://neo4j.org/learn/>].

6.1 Case Study

The case study proposed in this section presents the use of the RsM platform, in re-designing phases of the “Rotor” of a volumetric and rotary-piston engine: the Wankel engine, whose rotors-crankshaft sub-assembly is shown on Figure 4.

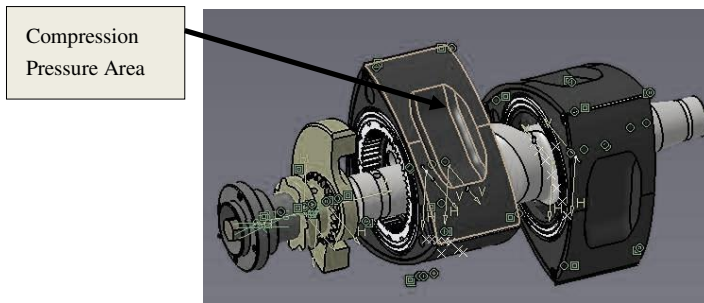


Fig. 4. A Rotors-Crankshaft sub-assembly of Wankel engine with compression pressure area highlighted on Rotor1

In the collaborative environment dedicated to the engine design project, data is distributed across different business and data management applications as following: the set of product functional requirements is managed with PTC Windchill RequirementsLink. Catia V5 is used to design the 3D digital models of the Wankel engine. Concerning the management of the design, data related to (rotors + crankshaft) sub-assembly is managed within Teamcenter Engineering, and those related to (engine-block + oil-pump) is managed with Windchill PDMLink. The verification numerical analysis are performed using Ansys-Workbench software, and resulting simulation data are stored and managed within an integrated Simulation Data Management (SDM) Folder reachable by the RsM search query engine. The main requirement for the redesign is to increase the allowable compression pressure on the sides of Rotors to 1,000,000 Pa (initially designed as following: $686000 \text{ Pa} \leq \text{Compression Pressure} \leq 833000 \text{ Pa}$), during the phases of compression/expansion-explosion. According to the verification requirement associated to this new specification, the maximum directional deformation (on Axis X) should not be higher than 0.0006 meter. To check the validity of the maximum value of the compression pressure proposed to the modification request, a Finite Element numerical analysis, with a pressure of 1,000,000 Pa, has been performed on the “Rotor”. As shown on Figure 5, post processing results give a maximum directional (on axis X) deformation value of 0.00050938 meter, a value less than the maximum deformation value (0.0006 meter) specified in the Verification Requirement related to the specification.

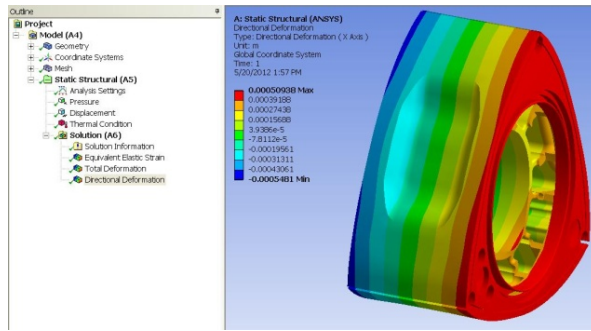


Fig. 5. Post processing results of Rotor1 maximum deformation verification simulation

In short, in this case study, the RsM platform orchestrates the links (via semantic relationships and aggregation of URI) between heterogeneous design entities of the Wankel engine, managed differently over various business and data management applications.

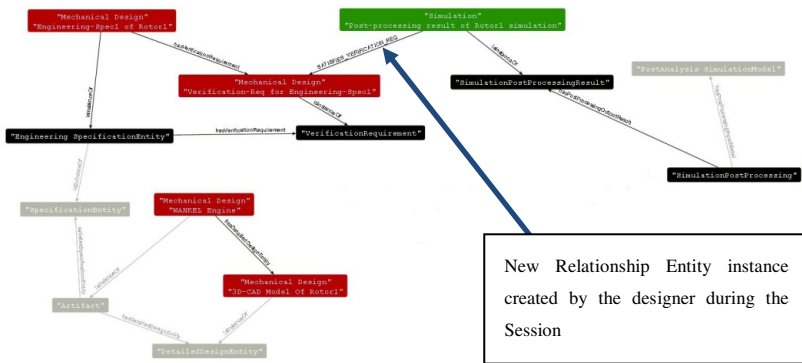


Fig. 6. Capitalisation graph of Wankel engine with BE and RE instances within the Viewer

A capitalisation graph resulting from a design Session is illustrated on Figure 6. The actual graph combines search results from the following queries: "find Rotor1Engineering Specification" (result on the left) and "find Rotor1 Post processing Simulation Results" (result on the right). Moreover, the graph display the semantic relationship "SatisfiesVerificationRequirement" created by the designer between the instance "Verification Requirement for Engineering Specification1" of VerificationRequirement and the instance "Post Processing Results of Rotor1 Analysis" of SimulationPostProcessingResult, and saved to the RsM database. On the capitalization graph, design entity instances are displayed in red, simulation entity instances in green, and in black, entities from the ontological model implemented within the RsM database, which normally are fully transparent to the end user, and that we intentionally choose to display here for overall logical view. At this stage of the process, a more complete structure of the capitalization graph with more data would give further

possibilities to designers to enrich the engine capitalization graph and share so created design knowledge with other designers working on the project.

7 Conclusion

In this paper, an ontology-based semantic relationship management approach has been proposed to improve product knowledge management and reuse in collaborative design context. Herein we focused on designing dedicated methodology and Ontology for capturing and sharing product BOL knowledge, by the mean of so called semi-structured data. The implementation of the proposed approach is made through Semantic Web (SW) technics and graph concepts, to allow a semantic integration of a set of heterogeneous information and knowledge sources (from various digital engineering applications) through the specification and implementation of a Relationship Manager. Currently, some aspects of the development of the RsM prototype, such as visualization of graph of Relationship Entities, data mapping to the RsM database, and the extraction and formalization of Business Entities during a query execution, are being tested, for validation. Even if the actual work focuses on product BOL, the RsM application is not limited to it, and will eventually cover entities related to the whole lifecycle, given the future objectives of the research and the implementation technics used to design the RsM.

References

1. Assouroko, I., Eynard, B., Troussier, N., Ducellier, G., Boutinaud, P.: Survey on standards for product data exchange and sharing: application in CAD/CAE interoperability. *International Journal of Design and Innovation Research* 5(1), 9–15 (2010)
2. Hirsch, C., Hosking, J., Grundy, J., Chaffe, T., MacDonald, D., Halytsky, Y.: The Visual Wiki - A New Metaphor for Knowledge Access and Management. In: *Proceedings of the 42nd Hawaii International Conference on System Sciences, Hawaii* (2009)
3. Fenves, S.J., Fofou, S., Bock, C., Sriram, R.D.: CPM2: A Core Model for Product Data. *Journal of Computing and Information Science in Engineering* 8(1), 014502–014508 (2008)
4. Fiorentini, X., Rachuri, S., Mahesh, M., Fenves, S., Sriram, R.D.: Description Logic for Product Information Models. In: *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, New-York City* (2008)
5. Chiara, E., Catalano, C.E., Camossi, E., Ferrandes, R., Cheutet, V., Sevilmis, N.: A product design ontology for enhancing shape processing in design workflows. *Journal of Intelligent Manufacturing* 20, 553–567 (2009)
6. Molina, F., Toval, A.: Integrating usability requirements that can be evaluated in design time into Model Driven Engineering of Web Information Systems. *Journal of Advances in Engineering Software* 40, 1306–1317 (2009)
7. Terzi, S., Bouras, A., Dutta, D., Garetti, M., Kiritsis, D.: Product lifecycle management – from its history to its new role. *International Journal of Product Lifecycle Management* 4(4), 360–389 (2010)

8. Lin, H.K., Harding, J.A.: A Manufacturing System Engineering Web Ontology Model on the Semantic Web for Inter-enterprise Collaboration. *Computers in Industry* 58(5), 428–437 (2007)
9. Sriti, M.F., Eynard, B., Boutinaud, P., Matta, N., Zacklad, M.: Towards a semantic-based platform to improve knowledge management in collaborative product development. In: *Proceedings of 13th International Product Development Management Conference, Milano (2006)*
10. Krima, S., Barbau, R., Fiorentini, X., Rachuri, S., Foufou, S., Sriram, R.D.: *OntoSTEP: OWL-DL ontology for STEP*. In: *International Conference of Product Lifecycle Management, Bath (2009)*
11. Keller, T., Tergan, S.-O.: *Visualizing Knowledge and Information: An Introduction*. In: Tergan, S.-O., Keller, T. (eds.) *Knowledge and Information Visualization*. LNCS, vol. 3426, pp. 1–23. Springer, Heidelberg (2005)
12. Jaeschke, G., Leissler, M., Hemmje, M.: *Modeling Interactive, 3-Dimensional Information Visualizations Supporting Information Seeking Behaviors*. In: Tergan, S.-O., Keller, T. (eds.) *Knowledge and Information Visualization*. LNCS, vol. 3426, pp. 119–135. Springer, Heidelberg (2005)
13. Grieves, M.: *Product Lifecycle Management: Driving the Next Generation of Lean Thinking*. McGraw Hill, New York (2005)
14. Assouroko, I., Eynard, B., Ducellier, G., Boutinaud, P.: *Semantic-based approach for the integration of product design and numerical simulation integration*. In: *International Conference of Product Lifecycle Management, Eindhoven (2011)*