SWRL as a rule language for ontology-based models in power plant design

Virginie Fortineau, Thomas Paviot, Ludovic Louis-Sidney, Samir Lamouri

▶ **To cite this version:**

Virginie Fortineau, Thomas Paviot, Ludovic Louis-Sidney, Samir Lamouri. SWRL as a rule language for ontology-based models in power plant design. 9th International Conference on Product Lifecycle Management (PLM), Jul 2012, Montreal, QC, Canada. pp.588-597, 10.1007/978-3-642-35758-9_53 . hal-00709139

HAL Id: hal-00709139
https://hal.science/hal-00709139

Submitted on 18 Jun 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

SWRL as a rule language for ontology-based models in power plant design

Virginie Fortineau[1], Thomas Paviot[1], Ludovic Louis-Sidney[2], and Samir Lamouri[1]

[1] LCPI/Arts et Métiers ParisTech, Paris, France.
[2] LISMMA/Supméca, Saint-Ouen, France.

Abstract. Current standard models have improved interoperability in Product Lifecycle Management. But the increase of constrains surrounding information management in an industrial context, and notably for power plant design, like the increase of data, actors, systems and the need of automation and collaboration leads to an increasing use of ontology-based models. However, the design of power plants presents a specific issue, which is rules expression and validation. This need of rules is generated by safety, interoperability or collaboration constrains. The Semantic Web Rule Language (SWRL) is an ontological language based on OWL-DL that allows in addition rules expression. This paper presents SWRL rules and their industrial applications but also limits of this language in PLM applications. It proposes in consequence other rules languages.

Keywords: rules; power plant; ontology; Description Logics (DL); SWRL

1 Context of the study

Process and power plants have specific issues involving information management along the product lifecycle, because plants are specific products. Some of these issues are [1]:

- long term information storage (from design to maintenance and disassembly);
- a large number of data produced;
- lots of actors and application systems involved;
- control policies and safety and checking constraints.

Like in many industrial fields, Product Lifecycle Management (PLM) is a global methodology that could provide efficient solutions to these issues [2]. Performing interoperability between information systems is a crucial step of this methodology [3], that must be achieved in terms of organization, technic and models semantic. Moreover, on a semantic viewpoint, expressing and validating rules is a determining issue in today's modeling. A rule is in this context a specification of a requirement or a constraint about the product or its management [4]. These rules are the focus of this study.

Semantic models have been developed to enhance interoperability in plant design, such as the Generic Product Model (GPM) used in nuclear applications in South Korea [1]. Standard semantic models also exist, that are especially adapted to plant specificities, like Part Librairies (ISO 10303) [5] and PLCS (ISO 10303, AP 239)[3] that are both expressed in EXPRESS language. However, the increasing need of safety and process automation and the complexity of these languages led to the development of ontology-based models, like ISO 15926 [6], that is expressed in the Web Ontology Language (OWL). Indeed, in various industrial fields, the use of ontologies has enhanced interoperablity along the product lifecycle. The question is then to wonder if ontology-based models are able to express and validate required rules during power plant design.

As a consequence, the present paper is organized as follow: the section 2 exposes the benefits of ontology-based models in PLM applications. Then the section 3 presents SWRL functionalities, SWRL being an ontological rule language, and its industrial applications. Section 5 discusses the capabilities of SWRL as a rule language in term of rule expression, implementation and validation and proposes other rule languages as research prospects. Finally section 6 concludes this paper.

2 Inference ontologies to enhance Product Lifecycle Management

2.1 Definition of inference ontologies

Ontologies are, according to the widely accepted definition given by Gruber [7],"an explicit specification of a conceptualization". In mathematical words, an ontology is a set of classes, properties connecting classes to one another, restrictions on properties and axioms [8]. As Fankam [9] explains, there exist two kinds of ontologies: storage ontologies and inference ontologies. Storage ontologies are used for information capture, storage, classification and reuse from heterogeneous sources. Contrarily to storage ontologies, inference ontologies can reason to infering information. Inference is the ability to make deductions about instances, regarding the classes, properties and restrictions defined [10] and is performed by inference engines. The Web Ontology Language (OWL) is the only standard language - recommanded by the W3C since 2004 - which expresses inference ontology-based models [11] (see figure 1). Moreover, OWL-DL is the most widespread sub-language of OWL in industrial applications. Since 2004, a large number of inference ontology-based models for PLM applications have emerged. This statement is explained by Ishak [12] by the fact that "ontologies propose a simple, exhaustive, implementable and humanly understandable description of the domain".

[3] http://www.plcs.org/ap239/index.html

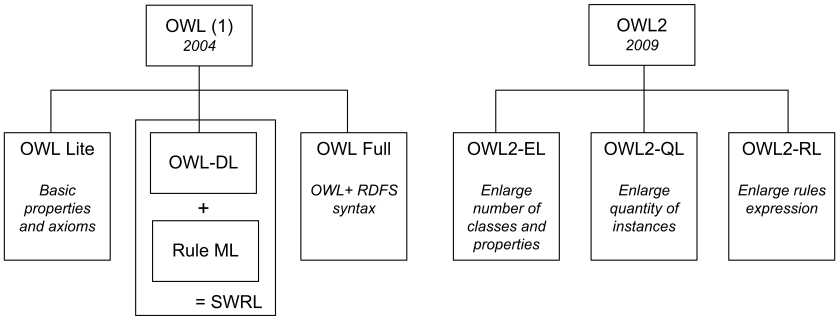

Fig. 1. Standardized OWL Languages

2.2 Expected benefits of inference ontologies in PLM

Expected and noticed benefits of inference ontology-based models in PLM are of various kinds, according to the literature, but can be classified into three categories, as following.

Integration and completeness: current PLM models suffer from a lack of content. It has been estimated that around 50 % of the available information is not stored in current information systems [13]. Moreover, information retrieval and reuse are not efficient in most systems too [14]. Inference ontologies provide a great expressivity that makes it possible to integrate information from different abstraction levels [15]. They improve knowledge capture and reuse [16] because they can deal with different kinds and sources of information. Illustrations of these abilities are the works of Yang [17] and Dong [18] in product configuration, the work of Jeong [19] which integrates different kinds of knowledge to improve design simulations and the work of Matsokis [20] which provides an ontology-based product model.

Reasoning and automation: the explosion of the number of data generated by information systems and the constant reduction of time-to-market raise the need of automated and intelligent systems. Inference ontology-based models provide intelligence, in the sense of the"ability to explicit implicit informations from the model specification" [21]. This functionality of inference ontology-based models comes from reasoning. Vegetti [22] uses for instance reasoning to automate Bill of Materials (BOM) generation for product families. Lim [23] automatically generates design annotations with inference ontologies to improve decision making at the design stage. According to Brandt [24] and Raza [25], deductions that an inference engine provides can be better than that of humans, and transform the model into a source of intelligent data.

Flexibility, extensibility and collaboration: current PLM tools fail to adapt to the perpetual evolutions of product environment [17]. They are too rigid to deal with processes that are by nature dynamic [26]. PLM systems need for instance a "'timely and precise information retrieval" [23]. Ontologies are web semantic tools. Consequently, they are well adapted to web applications and collaboration

and are by nature extendable. This extensibility is a way - according to Gimenez [27] - to prevent inference ontology-based models and information they contain from premature obsolescence. It is therefore possible to continuously adapt the model to product and process evolutions.

These noticed benefits of inference ontologies are even more important in power plant design. Automation, collaboration and information storage are crucial issues in designing plants. Ontologies are consequently a potential solution that might be explored to model plant information. The ISO 15926 proposes for instance OWL as the applicative language and XML as the normative syntax of its model. However, a limiting point to using inference ontology-based models is their ability to express and validate all required rules. The next section proposes SWRL as a rule language for ontology-based models.

3 Expressing rules in ontological models

3.1 SWRL as an ontological rule language

The Semantic Web Rule Language (SWRL) is a standard language based on OWL-DL and on the Rule Markup Language (RuleML) which provides both OWL-DL expressivity and rules from RuleML [28] (see again figure 1). It is the only approach that gathers ontology and rules in product development [29]. Rules in SWRL are implication rules. Hence, the syntax of SWRL is of the following form [30]:

$$antecedent \rightarrow consequent \tag{1}$$

This syntax implies that the consequent must be true when the antecedent is satisfied. OWL expressions can occur in both antecedent and consequent [31]. Different kinds of rules can be expressed in SWRL according to the literature. Next section presents only the one that are involved in case of power plant design, such as control and access policies (because of legal requisitions); configuration, association and naming rules (to enhance interoperation between actors and retrieve information); automation and strategic rules (to improve process and reduce Time-to-Market); assembly rules (related to product assembly constraints); and design rules, like design checking or product consistency checking.

3.2 Expressing rules in SWRL

Control policies have already been implemented using SWRL in healthcare field [32]. In this usecase, OWL-DL part of SWRL is used to provide a model of the hospital organization, and control policies are expressed as implication rules on classes and properties from the OWL-DL part. However, in this case, the organization arround the product must be modelized as well as the product itself to perform these control checking. This enlarge again informations required in the model.

Naming and classification rules are particulary well adapted to SWRL syntax, because OWL expressions are based on classes and properties. As explained

by Fiorentini [29], SWRL provides association rules, and allows to "associate instances to new classes and to create properties between instances". Examples of the use of SWRL in such cases can be found in Zhao [33] that provides a methodology to translate EXPRESS-driven models to OWL and SWRL, and in Dong [18] and Yang [17] in case of product configuration.

Process automation is determining to reduce time-to-market and to manage complex product design. As Raza [25] experimented in case of ERP systems, ontologies and SWRL rules allow knowledge management by providing intelligent data that "feed automated processes to aid agile manufacturing". In another context, Rossello [34] achieves automated actions with a SWRL-based model involving energy management. Elenius [35] automates military events analysis with a SWRL-based reasoning. Finally, Matsokis [20] and Lim [23] enhance the extendability of OWL models with SWRL rules by automating design annotations or knowledge (i.e. classes and properties) creation under a "learnable" approach.

The design of complex products, like power plants, involves a lot of components and assembly relationships. Design assembly rules can be expressed in SWRL. For instance Kim [36] uses SWRL rules to express features relations in product design. In the same way, the National Institute of Standards and Technology (NIST) recently transformed the Open Assembly Model in a OWL-DL based model with SWRL rules [37]. Finally, Mun [30] has expressed dependancy relations in SWRL in case of injection mould design.

Design checking rules and business rules are probably the most complex rules because they depend on a lot of conditions. However, Wicaksono [38] performs SWRL rules to detect design conflicts in product configuration. Even if SWRL is able to provide *restrict* rules, *partof* relationships and other design oriented axioms [37] [29], few applications of SWRL in design checking or modeling exist. Moreover, they involve mostly services or processes than the product itself.

4 SWRL rules execution and implementation

Current OWL editors like Protégé [4] allow OWL-DL-based model expression and reasoning. But reasoning on SWRL rules demands to use an external engine. In some cases, this external engine is specificaly designed for the demanding application like in Kim [36], but most of industrial models use a JESS Rule Engine [5]. JESS means Java Expert System Shell and is based on the Java language [18]. OWL ontology and SWRL rules are translated into Jess rules, *via* a JESS Bridge [17]. The JESS engine can then reason on the knowledge divided in two categories: JESS facts, derived from OWL model, and JESS rules, derived from OWL restrictions and SWRL rules (see figure 2). Once the JESS engine is run, it provides configuration results that are returned to OWL ontology [37]. SWRL-JessBridge is the JESS Rule Engine provided by Protégé [34] but it must be separatly downloaded.

[4] http://protege.stanford.edu/
[5] http://herzberg.ca.sandia.gov/jess/

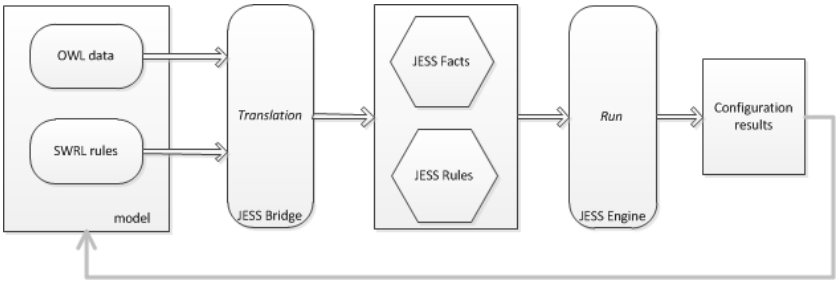

Fig. 2. SWRL rules execution with JESS engine

5 Limits and research prospects

5.1 Limits of SWRL in expressing required rules

OWL-DL, and consequently SWRL, works usually under the Open World Assumption (OWA) [29], because they originally focus on the Semantic Web, that deals with an unlimited knowledge resource (internet). The OWA means that "if a proposition cannot be proved to be true with the current knowledge, the system cannot declare this proposition as false" [39]. Unlike OWA, Closed World systems assume that empty knowledge is false, in fact that the available knowledge in the model is exhaustive. Industrial information systems can be considered on one side as closed worlds in which information should be threated under the Closed World Assumption (CWA) [40], since their information sources are finite. But on the other side, the interest of using ontologies is the extendability of the model, which is based on OWA. This is an important paradox that constraints model design. The same query, on the same model, could give different results regarding if the reasoning is made under a closed or an open world. Expressing control rules in SWRL, implies for instance to reason in a closed world [32]. Works are going on to reason on SWRL-based models, under the CWA, like the work of Wang [40] that proposes a new reasoner called BCAR that for.

SWRL has difficulties dealing with complex rules. For instance, it is not possible in theory to create rules with numbered predicates [35]. This limitation can be overtaken with SWRL built-ins [34]. Built-ins allow to define a rdf list of OWL arguments, that then feed a SWRL rule as an unique antecedent. However, this method is complex. This difficulty in expressing complex rules call into question the ability of SWRL to expressing design rules, that can depend on a lot of arguments. Because of it, preference is sometimes given to C++ language to express design checking rules of nuclear plants.

Finally, "SWRL rules cannot be used to modify existing information in an ontology" [34]. Indeed, SWRL rules originally create new knowledge, i.e. new instances or new properties between existing instances. It cannot change the value of an instance. For example, let imagine a class expressing a machine status, with instances "on" and "off" expressing if the machine is turned on or

off. If the machine status changes from on to off, an SWRL rule can express that "off" is a new instance connected to "machine". Then the machine status owns both "on" and "off" instances, since the SWRL rule does not replace "on" by "off". As for complex rules, this drawback can be avoided by using Built-ins. But it demands an SWRLbuiltinbridge, and a specific OWL model describing built-ins, which is an heavy and complex solution.

5.2 Exploring other rule languages

Other rules languages showed various capacities in modeling and checking rules in information systems. As the case in point, The OMG supports the Object Constraint Language (OCL) [41]. It can be used to describe additional constraints about objects in a model and makes it possible to check intrinsic model coherency and coherency between model and model instantiation. The language being particularly expressive, its use without restriction leads in practice to undecidability when the checking operations are performed automatically. Nevertheless, operating over on an appropriate fragment insure decidability with actual DL reasoners [42]. Model upon which OCL constraints apply are UML models, however translation works have already begun and revealed a compatibility between UML and OWL [43] along with OCL and SWRL [44]. The possibilities of expressing rules provided by OCL, additionally its close world orientation (more in line with industrial information system context, as presented above) could lead to a joint use with SWRL.

At last, another possibility would be to express rules using the EXPRESS modeling language [45]. The EXPRESS-G graphical part of the language looks similar to UML regarding modeling possibilities (e.g. attributes, inheritance, aggregation, polymorphism). However EXPRESS-M provides a rich set of rule modeling capabilities, based upon the concept of local and global rules. Local rules constrain the values of individual attributes or combinations of attributes for every entity instance. Global rules can be applied to check the consistency of a set of instances using queries, functions and procedures. Barbeau et al.[46] as well as Zhao et al [33] studied how to convert an EXPRESS model to an OWL/SWRL representation, but none of them managed to get converted all the semantics of rule restrictions.

6 Conclusion

In industrial applications, and specificaly for power plant design, inference ontologies appear as a relevant way to enhance information management during product lifecycle. But expressing rules is a crucial issue at the design step. These rules are of various kinds and demand a high expressivity. SWRL is the most appropriate rule language for ontology-based models because it is based on OWL-DL. It provides in addition implication rules. Even if several and various industrial applications of this language exist, it presents limitations that are explored in this paper, notably concerning the OWA and design checking. Therefore,

other rule languages should be explored, such as OCL and EXPRESS, that are compliant to UML-based models.

This work is the starting point of a global industrial project involving power plant design, which aims to improving information management. For this project, ontologies could be an efficient way of classifiying, storing and exchanging information, according to ISO 15926 recommandations. Expressing business rules is a crucial issue of this project, and could be achieved by SWRL. However, the literature reveals limitations of this language that lead the authors to investigate the use of other rule languages as a complement. The concurrent use of OWL, SWRL, OCL and EXPRESS and the interoperability of rule models has then to be further investigated, with the purpose to take benefit from each approach while ensuring the consistency of this aggregated model.

References

[1] Mun, D., Hwang, J., Han, S., Seki, H., Yang, J.: Sharing product data of nuclear power plants across their lifecycles by utilizing a neutral model. Anals of Nuclear Energy **35** (2008) 175–186

[2] 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)** (2010) 360–389

[3] Wegner, P.: Interoperability. ACM Computing Survey **28 (1)** (1996) 258–287

[4] Mun, D., Lee, S., Kim, B., Han, S.: ISO 15926-based data repository and related web services for sharing lifecycle data of process plants. Volume 38., International Conference on Product Lifecycle Management, Bath, UK (2006) 713–725

[5] Wei, G., Zhao, X., Ying, Y.: Design of building component library based on ifc and plib standard, International Conference on Computer Engineering and Technology (2010)

[6] Kim, B., Teijgeler, H., Mun, D., Han, S.: Integration of distributed plant lifecycle data using ISO 15926 and Web services. Annals of nuclear energy **38** (2011) 2309–2318

[7] Gruber, T.: Towards principles for the design of ontologies used for knowledge sharing. International Journal of Human Computer Studies **43 (5)** (1995) 907–928

[8] Maedche, A.: Ontology learning for the semantic web. Journal of Intelligent Systems, IEEE **16 (2)** (2002) 72–79

[9] Fankam, C.: Ontodb2: support of multiple ontology models within ontology based database, EDBT Ph.D. workshop, New York, USA (2008)

[10] Krima, S., Barbau, R., Fiorentini, X., R., S., Foufou, S., Sriram, R.: OntoSTEP: OWL-DL ontology for STEP, International Conference of Product Lifecycle Management, Bath, UK (2009)

[11] Antoniou, G., van Harmelen, F.: Web Ontology Language : OWL. Handbook on ontologies, International Handbooks on Information Systems **1** (2009) 91–110

[12] Ishak, K.: Integration of Semantic Interoperability in a Distributed Architecture for Multi-Site Planning. PhD thesis, Université de Toulouse (2010)

[13] Ben Kheder, A., Henry, S., Bouras, A.: An analysis of the interaction among design, industrialization and production, International Conference on Product Lifecycle Management, Bremen, Germany (2010)

[14] Louis-Sidney, L., Cheutet, V., Lamouri, S., Puron, O., Mezza, A.: A conceptual model for the implementation of IKOES (an inter-knowledge objects exchange system) in automotive industry. Journal of Engineering Applications of Artificial Intelligence. (2011) Article in Press. http://dx.doi.org/10.1016/j.engappai.2011.09.004.
[15] Fiorentini, X., Rachuri, S.: STEP-OAGIS harmonization joint working group. Technical report, National Institute of Standards and Technology (2009)
[16] Chang, X., Sahin, A., Terpenny, J.: An ontology-based support for product conceptual design. Journal of Robotics and Computer-Integrated Manufacturing **24** (2008) 755–762
[17] Yang, D., Miao, R., Wu, H., Zhou, Y.: Product configuration knowledge using ontology web language. Journal of Expert Systems with Applications **40** (2008) 863–878
[18] Dong, M., Yang, D., Su, L.: Ontology-based service product configuration system modeling and development. Journal of Expert systems with applications **38 (9)** (2011) 11770–11786
[19] Jeong, S., Wegner, D., Noh, S.: Validation of an ontology-based approach for enhancing human simulation in general assembly environments. Volume 3., Proceedings of the WCE 2010, London, UK (2010)
[20] Matsokis, A., Kiritsis, D.: Ontology applications in PLM. International Journal of Product Lifecycle Management **5** (2011) 84–97
[21] Baader, F., Horrocks, I., Sattler, U.: Description Logics as ontology languages for the semantic web. Journal of Computer science **2605** (2005) 228–248
[22] Vegetti, M., Leone, H., Henning, G.: PRONTO: An ontology for comprehensive and consistent representation of product information. Engineerings Applications of Artificial Intelligence, Article in Press **24(8)** (2011) 1305–1327
[23] Lim, S., Liu, Y., Lee, W.B.: A methodology for building a semantically annotated multy-faceted ontology for product family modeling. Journal of Advanced Engineering Informatics **25** (2011) 147–161
[24] Brandt, S., Morbach, J., Matiadis, M., Theissen, M., Jarke, M., Marquardt, W.: An ontology-based approach to knowledge management in design processes. Journal of Computers and Chemical Engineering **32** (2008) 320–342
[25] Raza, M., Kirkham, T., Harrison, R., Reul, Q.: Knowledge-based flexible and integrated PLM system at Ford. Journal of Information and Systems management **1 (1)** (2011) 8–16
[26] Catalano, C., Camossi, E., Ferrandes, R., Cheutet, V.: A product design ontology for enhancing shape processing in design workflows. Journal of Intelligent Manufacturing **20 (5)** (2009) 553–567
[27] Gimenez, D., Vegetti, M., Leone, H., Henning, G.: PRONTO: An ontology for comprehensive and consistent representation of product information. Computers in Industry **59** (2008) 231–241
[28] Horrocks, I., Patel-Schneider, P., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A Semantic Web Rule Language combininig OWL and RuleML. (2010)
[29] Fiorentini, X., Sudarsan, R., Suh, H., Lee, J., Sriram, R.: An analysis of description logic augmented with domain rules for the development of product models. Journal of Computing and Information Science in Engineering **10** (2010) 1–13
[30] Mun, D., Ramani, K.: Knowledge-based part similarity measurement utilizing ontology and multi-criteria decision making technique. Advanced Engineering Informatics **25** (2011) 119–130
[31] Hirankitti, V., Xuan, T.: A meta-reasoning approach for reasoning with SWRL ontologies, Internation Multiconference of Engineers (2011)

[32] Beimel, D., Peleg, M.: Using OWL and SWRL to represent and reason with situation-based access control policies. Journal of Data and Knowledge Engineering **70** (2011) 596–615
[33] Zhao, W., Liu, J.: OWL/SWRL representation methodology for EXPRESS-driven product information model. Computers in industry **59** (2008) 580–589
[34] Rossellò-Busquet, A., Brewka, J., Dittman, L.: OWL ontologies and SWRL rules applied to energy management, International Conference on Modelling and Simulation (2011)
[35] Elenius, D., Martin, D., Ford, R., Denker, G.: Reasoning about resources and hierarchical tasks using OWL and SWRL, International Semantic Web Conference, Chantilly, USA (2009)
[36] Kim, K., Manley, D., Yang, H.: Ontology-based assembly design and information sharing for collaborative product development. Computer-Aided Design **38** (2006) 1233–1250
[37] Fiorentini, X., Gambino, I., Liand, V., Foufou, S., Rachuri, S., Bock, C., Mani, M.: Towards an ontology for Open Assembly Model, International Conference on Product Lifecycle Management, Milan, Italy (2007)
[38] Wicaksono, H., Schubert, V., Rogalski, S., Ait Laydi, Y., Ovtcharova, J.: Ontology-driven requirements elicitation in product configuration systems, International Conference on Changeable, Agile, Reconfigurable and Virtual Production, Canada (2011)
[39] Drummond, N., Shearer, R.: The Open World Assumption. Technical report, University of Manchester, UK (2006)
[40] Wang, Q., Yu, X.: Reasoning over OWL/SWRL ontologies under CWA and UNA for industrial applications. Advances in Artificial Intelligence **7106** (2011) 789–798
[41] OMG: Object Constraint Language (OCL), version 2.3.1, Object Management Group (2012)
[42] Queralt, A., Artale, A., Calvanses, D., Teniente, E.: OCL-Lite: Finite reasoning on UML/OCL conceptual schemas. Data and Knowledge Engineering **73** (2012) 1–22
[43] OMG: Ontology Definition Metamodel (ODM), Object Management Group (2009)
[44] Milanovic, M., Gasevic, D., Giurca, A., Wagner, G., Devedzic, V.: Sharing ocl constraints by using web rules, International Workshop on Modelling Systems with OCL in the MODELS?07 Conference (2007)
[45] ISO-10303-11: Industrial automation systems and integration - Product data representation and exchange - Part 11: Description methods: The EXPRESS language reference manual., International Standard Organization (1994)
[46] Barbeau, R., Krima, S., Rachuri, S., Narayanan, A., Fiorentini, X., Foufou, S., Sriram, R.: OntoSTEP: Enriching product model data using ontologies. Computer Aided Design (2012) Accepted manuscript. DOI: 10.1016/j.cad.2012.01.008.