Domain Ontology for Expressing Knowledge of Variants of Thermally Modified Wood Products

Hele-Mai Haav(✉) and Riina Maigre

Department of Software Science, Tallinn University of Technology, Akadeemia tee 15a, 12618 Tallinn, Estonia
helemai@cs.ioc.ee, riina@ioc.ee

Abstract. The thermally modified wood producer Thermory AS manufactures about 400 different products, which are ordered in large number of variants that makes the expression of the product variant knowledge and its validation very important. In this paper, we express knowledge of product variants as domain ontology in order to capture the product knowledge in the way that is consistent and shareable between humans and machines. Using Ontology Web Language (OWL) as Description Logics (DL) based ontology representation language enables to detect inconsistency in the product knowledge and customer order requirements. Constraints on valid product variants are expressed as OWL class expressions and as rules in Semantic Web Rule Language (SWRL). The provided knowledge representation method makes it possible to reduce combinatorial complexity of description of product variants and to place correct manufacturing orders saving time and money for the company.

Keywords: Ontology · OWL · Product variant management · SWRL SPARQL

1 Introduction

Today many businesses need to deliver products that have variations in some attribute (or parameter) values. A certain combination of these attribute values on a particular product is called a variant or variation. In wood industry, wood products can have a set of common parameters for some product categories and in addition several variations in values of some other parameters. Customer orders specify values of variant parameters of an ordered product.

In traditional Enterprise Resource Planning (ERP) or Material Resource Planning (MRP) systems product variations are managed using Bill of Materials (BOM) with parameters[1] (or variant/matrix BOM). Product variant management is related to more general problem of product configuration. Several product configuration systems are available as parts of ERP (e.g. SAP[2]) or as standalone systems (e.g. Productoo[3]).

1 www.mrpeasy.com.
2 www.sap.com.
3 https://www.web4industry.com/product-configurator/.

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 161–171, 2018.
https://doi.org/10.1007/978-3-319-97571-9_14

All these traditional product or variant configuration systems represent product variant knowledge in the form of database tables (or matrices) creating, if necessary, a table including huge number of variants of the same product (i.e. for all combinations of values of parameters). This kind of knowledge representation is not well reusable, manageable, shareable, and interoperable with other systems used in a global enterprise. Therefore, one of the most important challenges of solving the product configuration as well as the product variant management problem is related to the knowledge representation that requires the expressive language for describing product variations and their constraints as well as customer preferences.

Ontology languages like OWL [9] provide the expressive and explicit way of capturing domain knowledge as well as reasoning on the basis of the described knowledge. They also enable reusability of the represented knowledge in other systems and semantic interoperability between different possibly distributed systems.

The thermally modified wood producer Thermory AS is a SME that currently does not use any well-known ERP systems but relies on its in-house developed information system (IS). We chose an ontology engineering approach for solving the product variant management problem for Thermory. To do that, we first created ontology based representation of available thermally modified product variations and then used it for resolving inconsistencies in the product variant knowledge as well as to detect inconsistency between the product variant knowledge and customer requirements. Ontology is represented in OWL and constraints on valid product variants are expressed as class expressions in OWL and as rules in Semantic Web Rule Language (SWRL) [4]. The DL reasoner Pellet [11] is used for ontology reasoning as it well supports SWRL rules. The novelty of our approach comparing to the traditional variant BOM lies in the significant reduction of number of product variant combinations to be described and managed as well as in the possibility to use DL reasoning services.

The paper is structured as follows. In Sect. 2 we give a background of the problem and in Sect. 3 we consider some related works. Section 4 is devoted to our original approach of expressing knowledge of product variants using OWL ontologies and SWRL rules. Conclusion and future work are presented in Sect. 5.

2 Motivation and Background

The thermally modified wood producer Thermory AS[4] is an Estonian company specializing in thermally modified solid wood flooring, decking, cladding and sauna products. The Thermory brand has become well-established in the United States and Canada, and has been shipped to over 55 countries around the world. Thermory AS uses chemical-free thermal modification process, where properties of wood are altered using only heat and steam[5]. As a result of the thermal modification process, wood's durability and resistance to mold and rotting increases. Due to its properties, the main wood species used by Thermory AS is ash, but pine, pecan, hickory and birch are also used. Production in Thermory's factory usually starts with unprocessed saw-timber and includes multiple

4 http://thermory.com/en/kontakt/about-company.
5 https://www.thermoryusa.com/modification.

production stages. Main production stages are dehumidification in drying kilns, thermal modification in the thermo-kilns, planing to dimensions and length and planing to profile. In addition to these stages, boards can go through brushing, end-matching and finishing stages.

Depending on the customer order, products can have different thermal modification levels. Two thermal modification levels are used by Thermory AS: medium (peak temperature 190°) and intense (peak temperature 215°). Possible thermal modification levels depend mainly on wood species. Boards have additional variations in dimensions, length, profile and suitable clips. Available profiles and suitable clips depend on dimensions of the board. Thermory's current IS is not able to automatically allocate materials and schedule resources. Therefore, all planning and scheduling is done manually, which is time consuming and costly. We have been working with Thermory in order to develop industry and production specific algorithms for automating the planning of materials and resources in the factory. As a part of this project a need for the list of descriptions of all product variants arose. Such knowledge does not currently exist in Thermory's IS, but it is necessary for automating the production planning.

3 Related Works

Our approach is indirectly related to works on ontology based product configuration. One of the first works (published in late 90s) that uses DL based knowledge bases for product configuration is [8]. They have built configurators based on DL based knowledge representation system CLASSIC [10] for a number of large telecommunications products sold by AT&T and Lucent Technologies.

At the same time the work towards a general ontology of configuration was developed in order to reuse and share configuration knowledge [12]. This ontology includes concepts like components, attributes, resources, ports, contexts, functions, constraints, and relations between these. It is formalized in Ontolingua [3] based on KIF [2] that lacks reasoning mechanism for checking the consistency of a knowledge base that is available in DL based languages.

In [15] an ontology-based product configuration model was developed and formalized using OWL and SWRL. A similar approach can be found in [14], where focus is on the semantics of constraints of product configuration that cannot be expressed by OWL. They provide a rule based ontological formalism for describing product structure and constraints of a product configuration and checking its validity.

Interesting relationships can be found between feature oriented domain analysis (FODA) [1] used basically for software line production and our ontology based method suggested for product variant management in manufacturing. The authors of [5] analyzed similarities and differences of feature models of FODA [1] and ontology based domain analysis methods. According to their work, similarities include using a concept vocabulary, enabling the expression of property and class hierarchies, and providing a constraint definition capability. In FODA, the latter is used for variability reduction but in ontology based domain analysis constraints are used for the description of property restrictions in class expressions. Both analysis methods allow to describe semantics of a domain and can be represented in machine readable form. Therefore, the Authors of [1] conclude that ontologies could effectively replace FODA models. As the advantages, ontology based analysis provides more expressive language than FODA and includes additional capabilities like reasoning and querying (via DL or SPARQL query support).

To the best of our knowledge, the only work that is tightly related to ontology based product variant management in manufacturing domain is devoted to the creation of the product feature ontology in [7]. This is intended to the management of the feature-based product line engineering in very large and complex product line organizations. Their goal is similar to what we have, to create multilevel ontology in order to significantly reduce number of product feature combinations to be managed comparing to using feature matrices. However, the goal and the scope (automotive industry) of their ontology are different from our ontology and there is no information about formalization of this ontology in any formal ontology language.

4 The Approach to Expressing Knowledge of Variants of Thermally Modified Wood Products

4.1 An Example

In a running example we consider a set of thermally modified wood product families like ash decking and cladding boards, pine decking and cladding boards and spruce decking boards that form the largest share of the production of Thermory.

Each product family includes a number of different products (i.e. parent products) that can have variants according to the values of some parameters (see Fig. 1).

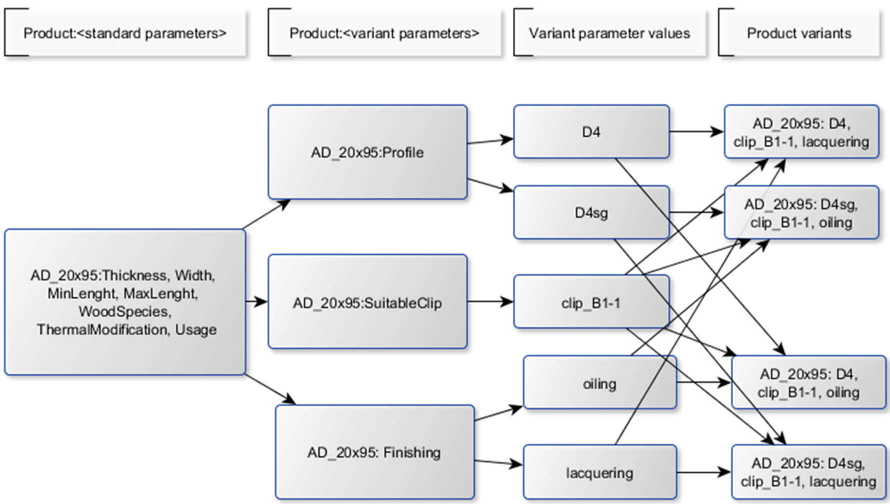

Fig. 1. An example of product variants in Thermory AS

Product families and product hierarchy are illustrated by ontology class hierarchy in Fig. 2.

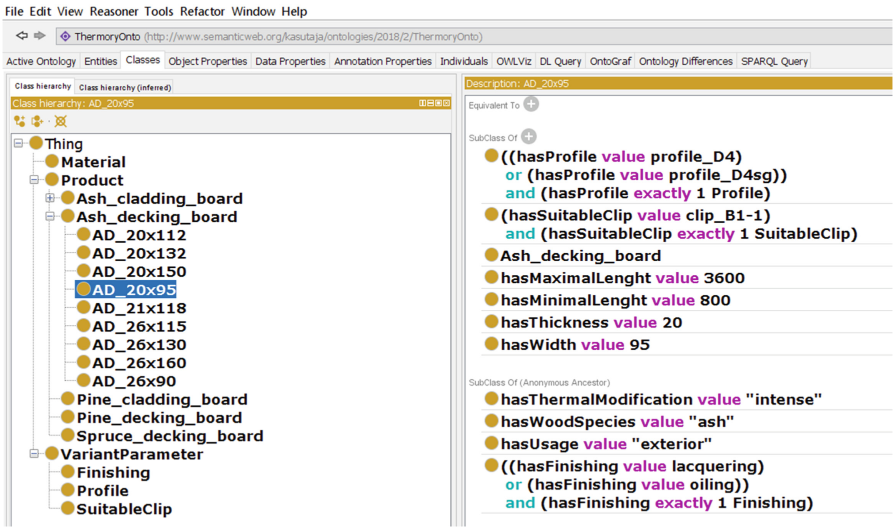

Fig. 2. An excerpt of the product ontology class hierarchy and examples of class expressions

Product variants are made up of a parent product that sets up values for standard parameters and a set of child products that represent different variations of this parent product according the values of variant parameters (see Fig. 1).

For example, standard and variant parameters of the ash decking board AD_20x95 are presented in Fig. 1. Some of standard parameters are common for the whole product family (e.g. ThermalModification, Woodspecies, Usage) and some are not (e.g. Thickness, Width, etc.). There are shown three variant parameters in Fig. 1 as follows: profile, suitable clip and finishing. The latter is variant parameter for the whole product family. The given possible parameter values allow creating 4 different product variants. There are more variant parameters. For example, actual length of an ordered product is a variant parameter too. It has associated constraints expressing that its value should be between minimum and maximum length given as product parameters. For some products, actual length parameter can obtain a value only from a specified set of valid values.

4.2 Principles of Ontological Modeling of Knowledge of Product Variants

We developed principles of capturing knowledge embedded in data collected about product families, their parameters, product variants and customer requirements to OWL ontology. The following general guidelines have been worked out:

1. Terminological knowledge about product families and product variants is expressed as the product ontology class hierarchy, object property and data property definitions and class expressions (TBox in DL). For each product family a subclass of the class

Product is defined as a complex class in OWL (see Fig. 2). Knowledge about product variants is defined in subclasses of the corresponding product family class. All subclasses of classes are disjoint. Object and data properties express either standard or variant parameters of a product. They are associated with the class Product having it as a domain.

2. Individuals (ABox in DL) are used to represent value choices of variant parameters (e.g. profile D4). Other parameter values are represented as data property values. In addition, product descriptions that are specified in orders are represented as distinct individuals of a class of a certain product with provided values for variant parameters. They are used in the reasoning process of checking consistency of ontology itself during its design time and for the verification whether specification of an ordered product is in correspondence with valid values of product variant parameters defined in ontology.
3. Constraints are represented as property restrictions in complex class expressions in OWL or as rules in SWRL. Class expressions define a set of individuals belonging to the class. SWRL rules are used to express constraints that cannot be represented by OWL. For example, if some calculations or comparisons are to be performed on product parameter values, then SWRL rules are used.
4. Reasoning is used for the verification of the validity of product variant parameters of an ordered product and for inferring product standard parameters according to the given product hierarchy. The verification is done by using standard ontology consistency check and evaluation of SWRL rules. Some of the rules assert new values to data properties of individuals. DL reasoners use Open World Assumption (OWA) for reasoning meaning that the model may be incomplete and new knowledge may be added that necessarily is not false. This is good for checking partially defined product variants but it creates problems for checking completeness of an individual product description (an individual that corresponds to an ordered product in Abox). For ensuring that a product description in Abox includes all necessary object property and data property assertions that model variant parameters and their values we propose to use SPARQL [13] queries (see Sect. 4.5) to retrieve individuals that do not include necessary properties. After corrections in Abox, if needed, the verification of the validity of the description of the given individual product in ABox can be performed by a DL reasoner.
5. During the evolution of ontology (according to the evolution of product variants) its consistency needs to be checked again by a DL reasoner before it can be used for the validation of parameters of an individual ordered product.

4.3 Definition of Product Ontology Classes

The product ontology class hierarchy corresponding to our running example is presented in Fig. 2. The class hierarchy contains disjoint classes for product families, product variants and variant parameters. A product family class is defined as a complex class with class expression including data property value restrictions for standard parameters of the product family products.

In addition, according to the specific product family, this class expression may define common property restrictions over object properties corresponding to variant parameters of a product family. Product variants are defined as subclasses of a product family class and their class descriptions specify only data property values and object property restrictions that correspond to the specific variant parameters of this product. They inherit common properties from their product family class.

For example, in Fig. 2, the Ash_decking_board product family class that is the superclass of the product variant class AD_20x95 includes property restrictions over hasFinishing object property and value restrictions on the hasUsage, hasThermalModification, and hasWoodSpecies data properties. In Fig. 2, we use the format of the ontology editor Protégé[6] to illustrate property restrictions as it is easy to read and short.

Ash_decking_board defines the class of products as the set of individuals that are linked to a finishing option by the hasFinishing property by using the cardinality restriction, which specifies that exactly one element can be in this relation. In addition, this class expression says that the class contains individuals that are connected by the hasFinishing property with an individual lacquering or oiling. In the similar way the specific class expressions are defined for the product variant class AD_20x95 for the hasProfile and the hasSuitableClip object properties.

Using such principles of construction of class expressions makes it possible to use a DL reasoner to automatically infer predefined data property values for an ordered product as well as to check if an individual is expressing an ordered product consistent with ontology (i.e. does it satisfy the conditions given in the class expression).

4.4 Constraints as SWRL Rules and Reasoning

Class expressions are a convenient way to represent constraints in OWL. However, OWL [9] is not able to describe all relations needed to express constraints. The expressivity of OWL can be extended by adding SWRL [4] rules to ontology. SWRL rules are Horn clause like rules in what atoms can be basically of the form C(x) and P(x, y), where C is an OWL description, P is an OWL property, and x, y are either variables, OWL individuals or OWL data values [4]. SWRL includes a number of built-in predicates for individuals to manipulate with data values.

We define the following data properties to capture constraints for violation of maximum and minimum lengths given in the corresponding product class definition: isViolatedMaxLenghtConstraint and isViolatedMinLenghtConstraint. These data properties are used in rules to check whether the corresponding constraint was violated or not. The reasoner Pellet [11] asserts values for these data properties according to the SWRL rules that represent constraints for violation of maximum and minimum lengths. The corresponding rules are presented in Fig. 3.

[6] https://protege.stanford.edu/.

```
Product(?x1), hasActualLenght(?x1, ?y1), hasMaximalLenght(?x1, ?z1), greaterThan(?y1, ?z1) ->
isViolatedMaxLenghtConstraint(?x1, true)

Product(?x1), hasActualLenght(?x1, ?y1), hasMinimalLenght(?x1, ?z1), lessThan(?y1, ?z1) ->
isViolatedMinLenghtConstraint(?x1, true)

Product(?x), hasOrderedQuantityPcs(?x, ?z), hasActualLenght(?x, ?y), divide(?j, ?m, 1000),
multiply(?m, ?y, ?z) -> hasOrderedTotalMeters(?x, ?j)

Product(?x), hasActualLenght(?x, ?y), hasWidth(?x, ?z), multiply(?m, ?y, ?z) -> hasAreaMM2(?x, ?m)

Product(?x), hasAreaMM2(?x, ?y), hasThickness(?x, ?z), divide(?c, ?m, 1000000000), multiply(?m, ?y, ?z) ->
hasCubicMeter(?x, ?c)
```

Fig. 3. SWRL rules (variables in rules are denoted using a question mark as a prefix)

We also developed some rules for lumber calculations as for the production and thermo-kiln management several measurements of lumber quantity are used. In general, measures of a board are given in millimeters. The reasoner fires rules and asserts data property values to the properties hasAreaMM2, hasCubicMeter, hasOrderedTotalMeters that are used for production orders and optimal packing of boards into thermo-kiln. In Fig. 4 we see the description of the individual named product1 that includes object and data property assertions related to the product order (marked with bold) and those that are asserted by the reasoner (marked with yellow background). The reasoner asserted data property values that correspond to standard parameters of the product and its product family as well as values that are asserted by rules.

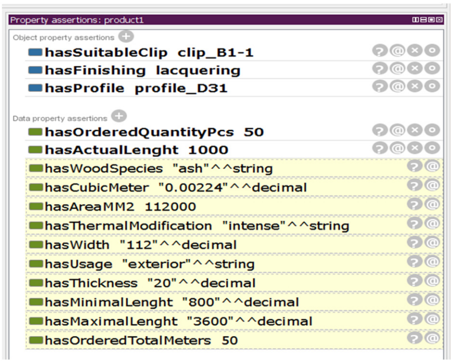

Fig. 4. Reasoning results showing data property assertions (Color figure online)

4.5 Completeness of Descriptions

As mentioned in Sect. 4.2, to ensure that the description of a specific product variant in Abox is complete with regard to certain variant parameters we need to use Closed World Assumption (CWA), which assumes that the specification information is complete. However, CWA reasoning and its combinations with the OWA are not well supported by now. Therefore, we propose to use SPARQL queries [10] for checking completeness of descriptions of individuals as depicted in Fig. 5.

```
PREFIX ThermoryOnto: <http://www.semanticweb.org/kasutaja/ontologies/2018/2/ThermoryOnto#>
SELECT ?product
WHERE
{?product rdf:type ThermoryOnto:AD_20x95.
FILTER NOT EXISTS {
?product ThermoryOnto:hasProfile ?profile.
?profile rdf:type ThermoryOnto:Profile } }
```

Fig. 5. A SPARQL query example

This SPARQL query returns the list of individuals of the AD_20x95 class that do not have link via the hasProfile object property to any individual of the Profile class.

4.6 Lessons Learnt and Future Visions

Our experience of using the combination of OWL ontologies, SWRL and SPARQL to solve the product variant management problem described in this paper shows that combining CWA and OWA is not very convenient to work with in this framework. OWL is good for the description of the model and SWRL for the expression of additional constraints of the model. DL reasoning is well suited for validation of the model during the design time. However, using SPARQL queries (or query templates) for checking completeness of descriptions of Abox individuals (i.e. CWA) before using DL reasoning (i.e. OWA) in order to check correctness of descriptions of individuals wrt to the model is not convenient. Main reason is that OWA does not make it possible to check integrity constraints, such as whether a property has a value or object property has a link to an individual, etc. To overcome this limitation, for each affected object property we need to create a corresponding SPARQL query and run it to get the resulting set of individuals satisfying the given criteria (e.g. see Fig. 5).

In order to make our approach simpler and prepare it for an industrial use we are seeking for possibilities related to CWA that are offered by SPARQL inference notation SPIN [6] and its new development Shapes Constraint Language (SHACL) that is W3C recommendation since 2017[7]. SHACL allows expressing rules and checking integrity constraints that individuals need to satisfy as well as includes possibilities for expressing mathematical computations.

We are planning to combine both OWL ontology and SHACL statements in order to make our approach to meet industrial needs. This may lead us to the method, where we exclude SWRL rules and use only OWL ontologies and SHACL. SHACL can be integrated with SPARQL if necessary.

From the business point of view, we see several applications of this approach in the product variant management system in Thermory and in other enterprises. In addition, Thermory's B2B site can benefit from this ontology enabling to provide valid variant parameter value options and combinations for a customer to choose from. Using it within material resource planning is foreseen but this requires the extension of ontology with knowledge about material consumption.

[7] https://www.w3.org/TR/shacl/.

5 Conclusions

This paper presented the approach to expressing knowledge of variants of thermally modified wood products for solving product variant management problems in Thermory AS. We provided the ontology based representation of possible thermally modified product variants and used it for resolving inconsistencies in the ontology as well as in checking consistence between product variants and customer order requirements. We combined OWL and SWRL to represent constraints on valid product variants. We suggest using SPARQL to check the completeness of the description of an individual product variant before checking whether this individual is consistent with ontology.

The approach is general and enables to use its principles in many industries where product variant management is important issue. The benefits of the ontology based approach comparing to database based solution lie in the fact that class expressions and rules enable to express the same knowledge more efficiently and to reduce combinatorial complexity of describing and using of the product variant knowledge.

Acknowledgement. This work was partially supported by the Institutional Research Grant IUT33-13 of the Estonian Research Council.

References

1. Acher, M., Collet, P., Lahire, P., France, R.: Comparing approaches to implement feature model composition. In: Kühne, T., Selic, B., Gervais, M.-P., Terrier, F. (eds.) ECMFA 2010. LNCS, vol. 6138, pp. 3–19. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13595-8_3
2. Genesereth, M.R., Fikes, R.E.: Knowledge Interchange Format Reference Manual. Technical report, Computer Science Department, Stanford University (1992)
3. Gruber, T.R.: Ontolingua: A Mechanism to Support Portable Ontologies. Technical report, Stanford University (1992)
4. Horrocks, I., et al.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML. https://www.w3.org/Submission/SWRL. Accessed 05 Mar 2018
5. Ines, C., Crepinšek, M., Kosar, T., Mernik, M.: Ontology driven development of domain-specific languages. Comput. Sci. Inf. Syst. **8**(2), 317–342 (2011)
6. Knublauch, H., Hendler, J.A., Idehen, K.: SPIN - Overview and Motivation. https://www.w3.org/Submission/spin-overview/. Accessed 15 May 2018
7. Krueger, C., Clements, P.: Enterprise feature ontology for feature-based product line engineering and operations. In: Proceedings of SPLC 2017, 10 p. ACM, Spain (2017)
8. McGuinness, D., Wright, J.R.: An industrial - strength description logic-based configurator platform. IEEE Intell. Syst. **13**(4), 69–77 (1998)
9. Motik, B., et al.: OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax. http://www.w3.org/TR/owl2-syntax. Accessed 04 Mar 2018
10. Patel-Schneider, P.F., McGuinness, D.L., Brachman, R.J., Resnick, L.A.: The CLASSIC knowledge representation system: guiding principles and implementation rationale. SIGART Bull. **2**(3), 108–113 (1991)
11. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A Practical OWL-DL Reasoner. Technical report, CS 4766, University of Maryland, College Park (2005)

12. Soininen, T., Tiihonen, J., Mannisto, T., Sulonen, R.: Towards a general ontology of configuration. Artif. Intell. Eng. Des. Anal. Manuf. AI EDAM **12**(4), 357–372 (1998)
13. SPARQL 1.1 W3C Recommendation Page. https://www.w3.org/TR/sparql11-overview/. Accessed 04 Mar 2018
14. Xuanyuan, S., Li, Y., Patil, L., Jiang, Z.: Configuration semantics representation: a rule-based ontology for product configuration. In: Proceedings of SAI Computing Conference, pp. 734–741. IEEE (2013)
15. Yang, D., Dong, M., Miao, R.: Development of a product configuration system with an ontology-based approach. Comput. Aided Des. **40**(8), 863–878 (2008)