Journal of Southeast University (English Edition) Vol. 22, No. 3, pp. 418 –422 Sept. 2006 ISSN 1003—7985

Configuration knowledge modeling of customizable products based on semantic web technologies

Ye Yan[1,2] Yang Dong[1] Jiang Zhibin[1]

([1] Department of Industrial Engineering and Management, Shanghai Jiaotong University, Shanghai 200240, China)

([2] College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou 310032, China)

Abstract: In order to solve the problem of modeling product configuration knowledge at the semantic level to successfully implement the mass customization strategy, an approach of ontology-based configuration knowledge modeling, combining semantic web technologies, was proposed. A general configuration ontology was developed to provide a common concept structure for modeling configuration knowledge and rules of specific product domains. The OWL web ontology language and semantic web rule language (SWRL) were used to formally represent the configuration ontology, domain configuration knowledge and rules to enhance the consistency, maintainability and reusability of all the configuration knowledge. The configuration knowledge modeling of a customizable personal computer family shows that the approach can provide explicit, computer-understandable knowledge semantics for specific product configuration domains and can efficiently support automatic configuration tasks of complex products.

Key words: knowledge modeling; semantic web technology; product configuration; mass customization

Product configuration has been an important activity in the implementation of the mass customization strategy. It builds product variants from a fixed set of well-defined component types, which satisfy individual customer requirements and complex configuration constraints of a product family. From the viewpoint of knowledge engineering, product configuration is a knowledge-intensive activity. The prerequisite for its efficient and automated execution is to explicitly model configuration knowledge. There have been some efforts in this direction. Earlier product configuration systems, such as R1/XCON[1], used production rules to represent configuration knowledge. Felfernig et al.[2] adopted UML to develop a conceptual configuration model. The object-oriented method was applied to model product configuration knowledge in Ref. [3]. However, there are few works addressing semantic modeling of configuration knowledge. It is essential for applications to perform configuration reasoning automatically and efficiently. Semantic web technologies[4] may serve as a meaningful enabler for its realization. Semantic web technologies provide explicit, machine-processable semantics for knowledge representation and enable computers and the web to provide much more automated services. Their application to configuration knowledge modeling can facilitate semantic consistency and reusability of knowledge and can improve the effectiveness and quality of automatic product configuration processes.

This paper presents an approach to ontology-based product configuration knowledge modeling and applies semantic web technologies, especially OWL web ontology language[5] and semantic web rule language (SWRL)[6], to capture the configuration ontology as well as the configuration knowledge and constraints in specific product domains with the aim of providing structural, consistent and reusable knowledge at the semantic level for efficient product configuration reasoning. Moreover, a product family example of configurable personal computers is used as a study case to describe the proposed approach.

1 Ontology-Based Product Configuration Knowledge Modeling

1.1 Modeling framework

An ontology describes a shared conceptualization formally and explicitly. The conceptualization is an abstract representation of the knowledge structure internal to the domain. Actually, an ontology identifies terms representing domain concepts and relationships between the concepts and allows formal and declarative descriptions of intended meanings of the terms. The ontology can serve as an important semantic foundation for

Received 2006-04-12.

Foundation item: The National Natural Science Foundation of China (No. 70471023).

Biographies: Ye Yan (1976—), female, graduate; Yang Dong (corresponding author), male, doctor, associate professor, dongyangcn@hotmail.com.

knowledge representation, sharing and reuse and can provide facilitates for reasoning on domain knowledge.

The approach of ontology-based configuration knowledge modeling is shown in Fig. 1. The configuration ontology provides general concept structures and a common semantic foundation for modeling configuration knowledge and rules of specific product domains. It can be reused in different application domains, such as computers and automobiles. Moreover, it helps to enhance reusability and maintainability of configuration knowledge and rules.

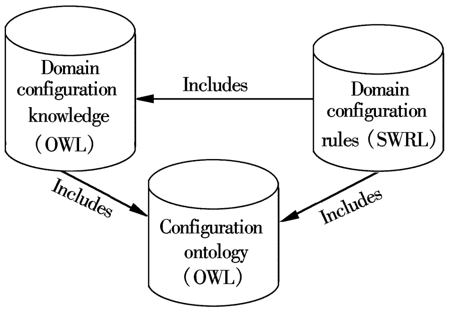
Domain configuration knowledge (OWL)
Includes
Domain configuration rules (SWRL)
Includes
Configuration ontology (OWL)
Includes

Fig. 1 Framework of ontology-based product configuration knowledge modeling

The approach uses OWL to formally define the configuration ontology and domain configuration knowledge. OWL is based on RDF/XML syntax and has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full. The reasons for using OWL are that its formal set of semantics supports the reasoning on concepts and individuals and that a growing number of automated tools are available to process OWL documents, including editors, such as Protégé-OWL[7] and reasoning engines. Although OWL provides rich vocabularies for representing knowledge, it cannot capture all types of knowledge relevant to a given domain. In particular, it does not provide a way to represent complex rules. Therefore, SWRL is used to formally define domain configuration rules based on the advantages of its close association with OWL and formal semantics.

1.2 Configuration ontology

The configuration ontology identifies concepts and relations common to all the product configuration domains, which extend the set of modeling concepts presented in Ref. [2]. One extension is to introduce classes relevant to product family development viewpoint to enable consistent exchange between knowledge bases of product development and configuration and to facilitate the maintenance of configuration knowledge. The other extension is to add classes explicitly defining different customer requirements to guide configuration tasks to generate the most suitable products. The classes and some relations in the configuration ontology that have been captured using the Protégé-OWL and displayed using the OntoViz[8] are shown in Fig. 2.

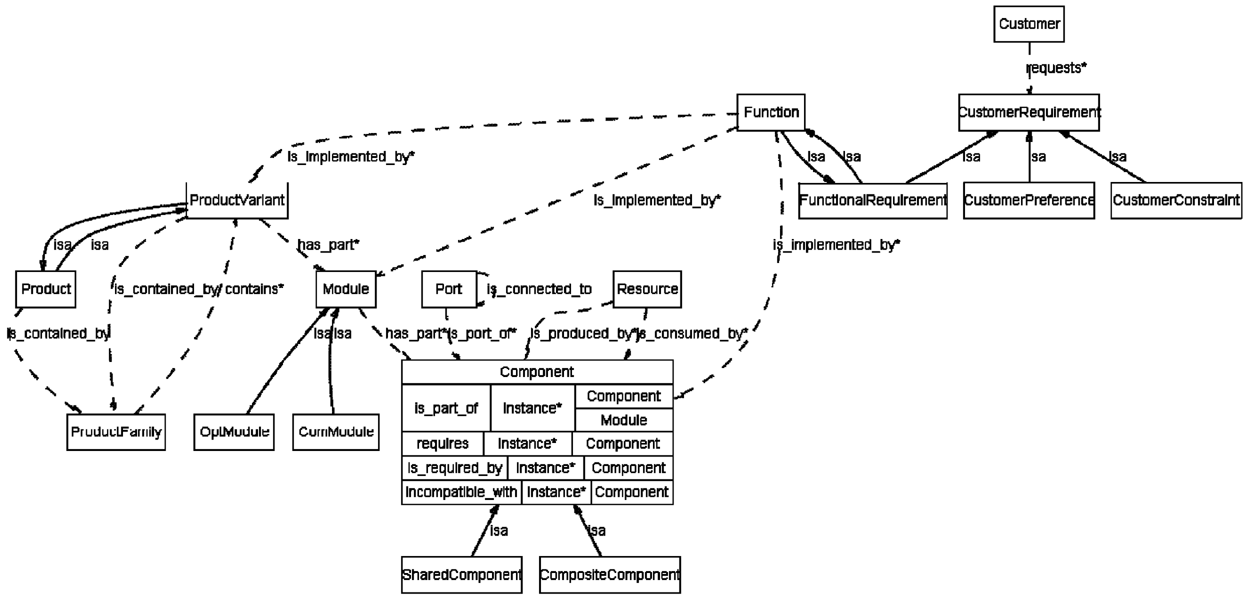

Fig. 2 Classes and relations in configuration ontology

A product family is a set of products that are derived from a common platform and are equivalent with product variants. A module is a sub-system in a product family and can be decomposed to sub-modules and/or components. The class has two sub-classes. One is the *ComModule* class that is shared by all the product variants of a product family and forms the common platform. The other is the *OptModule* class that represents the set of modules with specific features/functionality meeting particular customer requirements.

A component is a set of concrete parts with similar shapes, functions and assembly relationships and represents the basic building block of a product. It can be characterized by attributes with a predefined value domain. For example, a component can be described by an attribute *has_price* with float values. Although the attribute is not defined in the ontology to assure minimal ontological commitment, it can be represented as a datatype property in OWL. There exist *requires* and *incompatible_with* relations between components. The former expresses that the usage of one component type needs the existence of the other component type and the latter represents that two component types can not exist in a product at the same time. In addition, two types of part-whole relationships are widely used in the configuration knowledge representation, namely, composite and shared part-whole relationships. Components that are associated with other components and/or modules by the two relationships are represented by the classes *CompositeComponent* and *SharedComponent*, both of which are subclasses of *Component*. Each composite component must be connected to exactly one whole, while each shared component can be shared by at least two different wholes. Moreover, a composite component cannot be a shared component at the same time.

The *Function* class is an abstract characterization of uses that products or components provide to customers. Complex functions can be decomposed into simple functions through the *has_part* relation. In addition, *implements* and *is_implemented_by* express the relationships between functions and technical concepts, namely, components, modules and products. The resource refers to a kind of service provided by components, such as storage capacity of hard disks. In fact, product configuration can be seen as a resource balancing process, where the produced resources and consumed resources must be balanced by selecting the appropriate set of components. The port represents the assembly structure of products, namely, the way in which components and modules are connected to each other. Actually, functions, resources and ports can be represented as attributes of components and modules. The reason for representing them as classes is that they may be characterized by their own generalization-specialization hierarchies and attributes.

Customers request configuration activities by putting forward their requirements for configurable products, which are described by the following subclasses of the *CustomerRequirement* class. The *FunctionalRequirement* class involves requests for product functions. The class *CustomerConstraint* expresses global constraints or attribute restrictions of local entities. The *CustomerPreference* class represents priority criteria for selecting components and modules. For example, a customer may require a hard disk made by Maxtor may prefer to a product with a lower price.

2 Domain Configuration Knowledge

The configuration ontology can be refined to model the configuration knowledge of specific product domains. This is described by modeling configuration knowledge of a personal computer family in this section.

First, function hierarchies are identified. Three basic function classes *DataProcessing*, *DataStorage* and *DataDisplay* are created based on product family analyses. They are the subclasses of the *Function* class and can be specialized to represent more specific functions. Such a hierarchy enables customers to easily specify function requirements and facilitates the identification of components and modules composing products by mapping functions to technical structures. Secondly, components, modules and their properties are captured. The personal computer family has two main component classes, namely, *Hardware* and *Software*. Each class represents a hierarchy of subclasses. For example, twelve subclasses of the *Hardware* class are created and can be further classified into their subclasses. For example, *DiskDrive* is specialized to two disjoint subclasses *HardDiskDrive* and *FloppyDiskDrive* and *CPU* can have the *IntelCPU* subclass. Moreover, attributes and relations of the classes are defined as properties in OWL. For instance, a datatype property *has_version* defines the version attribute of the *Software* class. Finally, the classes *Port* and *Resource* are defined. Based on personal computer structures, specific ports and resources are defined as the subclasses of the two classes, respectively, for the components identified above. For example, motherboards have the port *PrimaryIDESlot* that is connected to the port *Data-Connector* of hard disk drives. Hard disk drives produce the resource *HDCapacity* that is consumed by other components.

The domain configuration knowledge captured above is coded in the OWL DL language. For example, OWL specification of the class *IntelCPU* is shown as follows:

```
〈owl: Class rdf: ID = "IntelCPU"〉
  〈rdfs: subClassOf rdf: resource = "#CPU"/〉
  〈rdfs: subClassOf〉
    〈owl: Restriction〉
```

```
            〈owl: onProperty〉
              〈owl: DatatypeProperty rdf: about = "#has _ maker"/〉
            〈/owl: onProperty〉
              〈owl: hasValue
rdf: datatype = "http: //www. w3. org/2001/XMLSchema#string"〉Intel
〈/owl: hasValue〉
            〈/owl: Restriction〉
          〈/rdfs: subClassOf〉
        〈/owl: Class〉
```

In addition, a description logic reasoner is used to check the consistency of the knowledge and automatically compute the class hierarchy after the domain configuration knowledge is defined. The produced results show that the defined knowledge is consistent, that is, there are no classes that can never contain any individuals, and several classes are reclassified. For example, the class *CPU* is not only the subclass of the *Hardware* class but also that of the *CompositeComponent* class. These reasoner services help to keep knowledge in a maintainable and modular state and promote the reuse of the knowledge by applications.

3 Domain Configuration Rules

During the configuration of mass-customizable products, complex configuration constraints and customer requirements restrict valid and satisfactory configuration solutions. These constraints and requirements are described by SWRL rules to provide unified representation of configuration restrictions.

SWRL is mainly used to represent Horn-like rules. The rules are of the form of an implication between an antecedent (body) and consequent (head), which means that the conditions specified in the consequent must hold whenever the conditions specified in the antecedent are satisfied. Both the antecedent and consequent consist of zero or more atoms. Atoms are the form $C(x)$, $P(x, y)$, *sameAs*(x, y) or *differentFrom*(x, y), where C is an OWL description; P is an OWL property; x and y are variables, OWL individuals or OWL data values. Multiple atoms are treated as a conjunction. Moreover, based on the OWL syntax and semantics, SWRL provides extended high-level abstract syntax, XML syntax, RDF concrete syntax and model-theoretic semantics. Consequently, domain configuration rules in SWRL contain terms in the configuration ontology and domain configuration knowledge represented by OWL. This enables a good combination of configuration knowledge bases in OWL with corresponding configuration rule bases in SWRL at the syntactic and semantic level.

Based on the knowledge modeling and representation described in section 2, rules related to technical restrictions and customer requirements in the personal computer configuration are identified and represented by SWRL, as shown in the following rules:

VideoCard(?x1) ∧ has _ port(?x1, ?x2) ∧ AGPConnector(?x2) ∧ HardDiskDrive(?x3) ∧ produces(?x3, ?x4) ∧ HDCapacity(?x4) ∧ has _ value(?x4, ?x5) ⇒

swrlb: greaterThanOrEqual(?x5, 200);

CPU(?x) ⇒ IntelCPU(?x)

The first rule expresses a technical restriction that video cards with an AGP connector require at least 200 M hard disk capacities. *VideoCard*, *AGPConnector*, *HardDiskDrive*, *has _ port, produces* and *has _ value* are vocabularies in the configuration knowledge defined in OWL and *swrlb*: *greaterThanOrEqual* is a built-in predicate in the SWRL document. The second rule represents that the customer requires that the maker of the CPU be Intel.

4 Conclusion

The key to automatic product configuration is to explicitly model and represent configuration knowledge at the semantic level. This paper presents an approach of ontology-based configuration knowledge modeling supported by semantic web technologies, especially OWL and SWRL. The designed configuration ontology provides common terms for semantically describing configuration knowledge and rules in a configurable personal computer family. The OWL DL sublanguage is used to formally define the ontology and specific product configuration knowledge. In addition, complicated configuration rules are defined in SWRL. This facilitates syntactic and semantic interoperability between configuration knowledge bases and rule bases due to the tight connection of the two languages. The consistent knowledge in OWL and rules in SWRL can be directly applied to configuration systems or easily mapped to facts and rules in configuration engines, such as JESS facts and rules, to implement efficiently automatic reasoning of product configuration tasks.

References

[1] Barker V, O'Connor D. Expert systems for configuration at digital: XCON and beyond [J]. *Communications of the ACM*, 1989, **32**(3): 298 − 318.

[2] Felfernig A, Friedrich G, Jannach D. Conceptual modeling for configuration of mass-customizable products [J]. *Artificial Intelligence in Engineering*, 2001, **15**(2): 165 − 176.

[3] Ong S K, Lin Q, Nee A Y C. Web-based configuration design system for product customization [J]. *International Journal of Production Research*, 2006, **44**(2): 351 − 382.

[4] Antoniou G, van Harmelen F. *A semantic web primer* [M]. Massachusetts: The MIT Press, 2004. 1 − 18.

[5] Smith M K, Welty C, McGuinness D L. OWL web ontology language guide [EB/OL]. (2004-02-10) [2006-03-15]. http://www. w3. org/TR/owl-guide.

[6] Horrocks I, Patel-Schneider P F, Boley H, et al. SWRL: a semantic web rule language combining OWL and RuleML [EB/OL]. (2004-12-21) [2006-03-15]. http://www. daml. org/rules/proposal/.

[7] Horridge M, Knublauch H, Rector A, et al. A practical guide to building OWL ontologies with the Protégé-OWL plugin and CO-ODE tools [EB/OL]. (2004-08-27) [2006-03-15]. http://www. co-ode. org/resources/tutorials/ProtegeOWLTutorial. pdf.

[8] Sintek M. OntoViz tab: visualizing Protégé ontologies [EB/OL]. (2005-02-03) [2006-03-15]. http://protege. stanford. edu/plugins/ontoviz/ontoviz. html.

基于语义 web 技术的定制产品配置知识建模

叶　艳[1,2]　杨　东[1]　江志斌[1]

([1] 上海交通大学工业工程与管理系,上海 200240)
([2]浙江工业大学机电工程学院,杭州 310032)

摘要:为成功实施大规模定制战略,研究解决了在语义层次建模产品配置知识的问题. 结合语义 web 技术,提出了基于本体的配置知识建模方法. 开发了通用配置本体为建模特定产品领域的配置知识和规则提供公共的概念结构,采用 OWL web 本体语言和语义 web 规则语言(SWRL)形式化地表示了配置本体、领域配置知识和规则,以促进各种配置知识的一致性、可维护性和重用性. 可定制的个人计算机族的配置知识建模说明了该方法能为特定的产品配置领域提供明确的、计算机可理解的知识语义,高效率地支持自动的复杂产品配置任务.

关键词:知识建模;语义 web 技术;产品配置;大规模定制
中图分类号:TP18;TH16