Product Variety Modeling Based on Formal Concept Analysis

Taioun Kim†
Department of Industrial and Management Engineering
Kyungsung University, Nam-gu, Busan, Korea
Tel: +82-51-620-4726, E-mail: twkim@ks.ac.kr

Received, March 11, 2009; Revised, August 29 & November 23, 2009; Accepted, January 13, 2010

Abstract. Increasing product variety based on product family and product platform provides a company with a competitive advantage over its competitors. As products become more complex, short-life cycled and customized, the design efforts require more knowledge-intensive, collaborative and coordinating efforts for information sharing. By sharing knowledge, information, component and process across different families of products, the product realization process will be more efficient, cost-effective and quick-responsive. Formal Concept Analysis (FCA) is used for analyzing data and forming semantic structures that are formal abstractions of concepts of human thoughts. A Web Ontology Language (OWL) is designed for applications that need to process the content of information instead of simply presenting information to humans. OWL also captures the evolution of different components of the product family. The purpose of this paper is to develop product variety modeling to increase the usefulness of common platform. In constructing and analyzing product ontology, FCA is adopted for conceptual knowledge processing. For the selected product family, product variety Ontology is constructed and implemented using protégé-2000.

Keywords: Product Variety, Product Family, OWL, Semantic Web, Ontology, Formal Concept Analysis

1. INTRODUCTION

To survive in global competition, companies need to develop a wide range of products to fit several market segments with different quality levels. At the same time, a new methodology for product variety is required to optimize the product development efforts across product families and generations. This challenge has led to work in product architecture (Ulrich and Eppinger, 2003), product family design (Simpson *et al*., 2001), and product variety design (Martin and Ishii, 2002). A product variety modeling requires horizontal integration across different models as well as vertical integration in ordinal concurrent engineering. Product family methodology provides a useful scheme for product variety modeling.

A product family is defined as a group of related products that share common features, components, and subsystems to satisfy a variety of market niches (Simpson *et al*., 2001). A Product family comprises a set of variables, features or components that remain constant from product to product, and other characteristics that vary from product to product. A Product platform is the set of common parameters, features, and/or components that remain constant from product to product within a given product family. In order to analyze the product variety via product family ideas, disassembly operation is essential. Min *et al*. (2004) proposed a priority assignment rule in multi-product disassembly for the used product recycle. Recent environmental issues put emphasis on the reuse and recycle of the used product.

As products become more complex, short-life cycled and customized, the design efforts require more knowledge-intensive, collaborative and coordinating efforts for information sharing. By sharing knowledge, information, component and process across different families of products, the product realization process will be more efficient, cost-effective and quick-responsive.

A new product design idea is conceptualized based on functional requirements, which is then realized into a product that fulfills customer requirements through behavior. Quality Function Deployment (QFD) has been widely adopted for transforming functional requirements into design parameters. Since the focus of QFD is placed on the early stage of a product development, the uncertainty in the input data of QFD is inevitable. To deal with this issue, robust QFD is proposed which is capable of considering the uncertainty in the input data and the resulting variability of the output (Kim and Kim, 2008). In order to reduce the risks and improve the productivity in the new product development, a methodology to ana-

† : Corresponding Author

lyze user information and determine product specification is proposed (Ishii *et al*., 2003). The method utilizes a stimulus-response model and product characteristics deployment.

Product Data Management (PDM) has been used to capture some types of non-geometric knowledge along with basic parametric information like manufacturing processes, Bill Of Material (BOM), inventory, etc., but these systems lack a formal product representational framework such as function, structure and behavior of a product.

For a formal product representational scheme, the idea of feature is required. A generic definition of feature is defined as anything of interest. More formally, a feature is described by any portion of a structure-behavior-function description of the designed object that may affect the goal of a particular point of view. From this viewpoint, functional, behavioral and structural features are presented in detail (Brown, 2003). Koga and Aoyoma (2004) define product behavior as product component existing in a state at the given moment and shifting between states. A Petri net and reachable tree are adopted for the representation of a product behavior. Representing and storing design data in an efficient way will facilitate the information retrieval and reuse of existing data, and the reduction of product development time.

Semantic web technology based on product family ontology has been adopted to overcome the shortcomings of PDM (Nanda *et al*., 2004).

Formal Concept Analysis (FCA) is used for analyzing data and forming semantic structures that are formal abstractions of concepts of human thoughts. In FCA, two important elements are ‘formal objects’ and ‘formal attributes.’ In an information retrieval application, documents could be considered object-like and terms considered attribute-like. Other examples of sets of formal objects and formal attributes are tokens and types, values and data types, data-driven facts and theories, words and meaning, etc (Priss, 2005).

The purpose of this paper is to develop a product variety modeling using web ontology language. The capturing process of product features is based on FCA. A framework structure for product variety modeling using OWL and FCA is proposed. As an illustrative example, a group of electric shavers is analyzed based on the proposed methodology.

2. ONTOLOGY AND FCA

2.1 Ontology

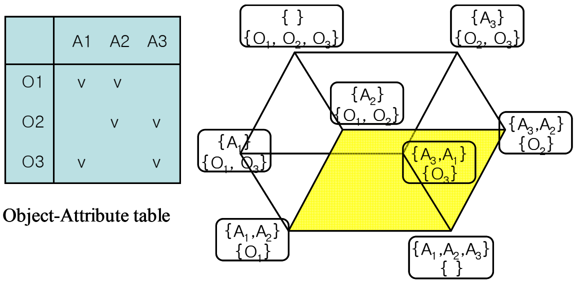
Ontology is defined as a formal, explicit specification of a shared conceptualization of a domain of interest. It consists of a set of concepts, axioms, and relationships that describe a domain of interest. A Semantic Web, which focuses on the intelligent aspect of Web, is originally defined as a web of data that can be processed directly or indirectly by machines, which operate intelligently. A Web Ontology Language (OWL) is designed for applications that need to process the content of information instead of just presenting information to humans. Three major uses of ontologies are communication between machines and human, automated reasoning, and representation and reuse of knowledge. The third case is utilized in this modeling framework.

2.2 FCA

FCA is a principled way of automatically deriving an ontology from a collection of objects and their properties. The method is used for data analysis, knowledge representation and information management. In FCA, a (formal) context, a (formal) concept, concept lattice of a context are used.

A *(formal) context* consists of a set of objects O, a set of attributes A, and a relationship between objects and attributes. It can be regarded as a bipartite graph $R \in O \times A$. A *(formal) concept* for a context is defined to be a pair (O_i, A_i) such that O_i and A_i are subsets of O and A respectively, and every object in O_i has every attribute in A_i. In the domain of formal concept, O_i is called the extent of the concept, A_i the intent. The ordered set of formal concepts of (O, A, R) is called the concept lattice of (O, A, R).

From a philosophical viewpoint, a concept is a unit of thought consisting of two parts, the extension and intention. The extension includes all objects belonging to this concept and the intention comprises all attributes valid for all those objects. A formal context is a combination of objects, attributes, and the incidence relation in a mathematical definition as a tool for the description of those elementary linguistic situations.



	A1	A2	A3
O1	v	v	
O2		v	v
O3	v		v

Object-Attribute table

Figure 1. Formal context and its concept lattice.

An example showing formal context and its corresponding concept lattice is illustrated in Figure 1. The formal context is composed of three objects O_1, O_2, O_3 and three attributes A_1, A_2, A_3. Object O_1 consists of A_1 and A_2 as shown in the Figure. The lattice is drawn as a Hasse diagram. The smallest concept including object O_3 is the one with objects $\{O_3\}$, and attributes $\{A_1, A_3\}$. The largest concept involving the attribute of A_2 is the one with objects $\{O_1, O_2\}$.

In an information retrieval application, documents could be considered object-like and attribute-like. Other examples of sets of formal objects and formal attributes are tokens and types, values and data types, data-driven facts and theories, words and meanings and so on. The sets of formal objects and formal attributes together with their relation to each other form a “formal context”.

Ontology and FCA are compared as the following. First, FCA can be used as a technique for ontology engineering. It supports the structuring of some given data by means of concept lattices. Additionally, FCA can be used for visualizing ontology. Second, in FCA, a distinction between concepts and attributes is clear, it is not so sharp in ontology. The relationship between ontology and FCA is compared in Table 1.

2.3 FCA modeling

Some basic definitions of formal concept analysis are adopted from Eijck and Zwarts (2004) and modified to facilitate the understanding of OWL construct based on FCA.

Definition 1. *A formal context is a triple $K := (O, A, R)$, where O is a finite set of objects, A is a finite set of attributes, R is a binary relation on O x A, with $(o, a) \in R$ expressing o has a.*

Definition 2. *A relation R is a sub-set of the cross product between objects and attributes. If an object o_1 has property a_1, where $o_1 \in O$, and $a_1 \in A$, then the relationship is represented by $(o_1, a_1) \in R$.*

Definition 3. *FCA represents a concept in a given context (O, A, R) as a pair (X, Y) with $X \subseteq O$ and $Y \subseteq A$ satisfying the following conditions,*

$X = \{ x \in O \mid \forall y \in Y, xYy\}$,

$Y = \{y \in A \mid \forall x \in X, xRy\}$.

Definition 4. *If (X, Y) is a concept, X is called its extent, and Y is intent.*

Definition 5. *The concept lattice is determined by sets O and A, and relation $R \subseteq O$ x A is the set $(P, \leq)$, where*

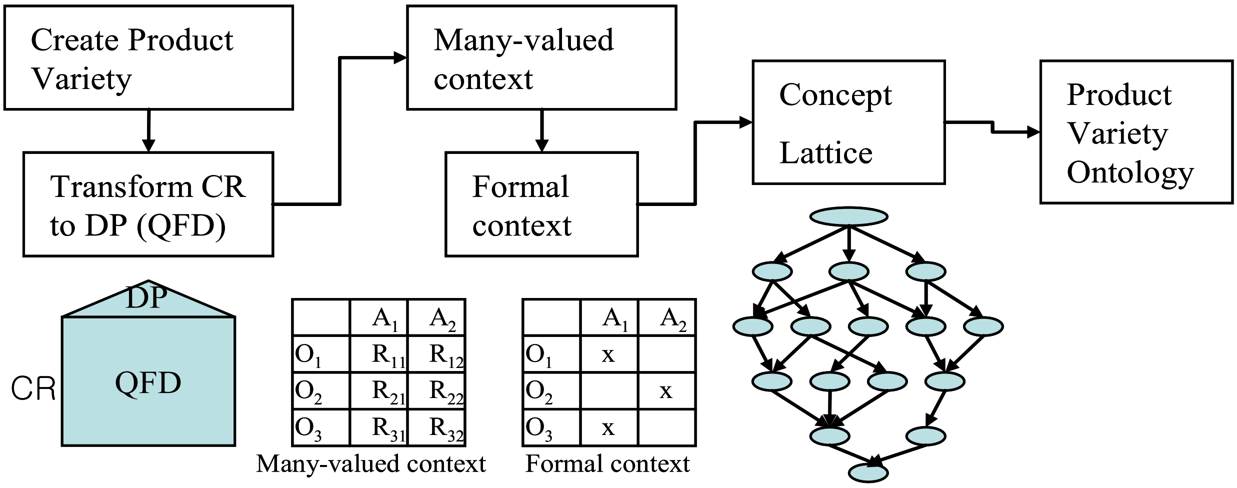
$P = \{(X,Y) \mid X \subseteq O,\ Y \subseteq A,\ (X,Y)$ *as a concept*$\}$

and $\leq$ *is given by*

$(X_1, Y_1) \leq (X_2, Y_2) :\equiv X_1 \subseteq X_2$.

Table 1. Comparison between Ontology and FCA.

	Ontology	Formal Concept Analysis
Purpose	To model a shared understanding of the reality	To analyze and structure the domain
Requirement	none	Objects (data set)
Emphasis	Intentional part (The set of descriptors of a formal concept)	Intentional and Extensional part (objects of a formal concept)
Usage	A formal and explicit conceptualization	As a technique for ontology engineering (Visualizing ontology)
Distinction between concept and attribute	Not obvious	Obvious



	A_1	A_2
O_1	R_{11}	R_{12}
O_2	R_{21}	R_{22}
O_3	R_{31}	R_{32}

Many-valued context

	A_1	A_2
O_1	x	
O_2		x
O_3	x	

Formal context

CR: Customer Requirements, DP: Design Parameters, QFD: Quality Function Deployment

Figure 2. Product variety ontology architecture using FCA.

3. PRODUCT VARIETY ONTOLOGY MODELING USING FCA

A formal architecture for product variety modeling will help to share concepts, ideas, and thoughts among the interested product design group. Figure 2 illustrates a structured architecture to develop a product variety ontology starting from customer requirement to FCA and web ontology language.

The procedures for product variety ontology modeling are proposed as the following steps.

Step 1. Create product variety

Two types of product variety are considered. One type is the variety created within the current product line by changing the quality and market segmentation. The second is the variety across time span for future generation. The first category is referred to as 'spatial variety', and the second one is 'generational variety' (Martin and Ishii, 2003).

In order to expand product variety over a time span, the external drivers for next generation product is forecasted among professional design groups. The case of electric shavers is adopted and analyzed. Component type of electric shavers and generational variety and spatial variety are shown in Figure 3.

Step 2. Transform CR to DP using QFD

To create a generational product variety, QFD is adopted to transform customer requirement into design parameter.

QFD is composed of two phases. The first phase translates the subjective customer requirements into quantifiable engineering metrics. The second phase maps the engineering specifications to the components used in the design. Figure 4 Shows a QFD chart transforming customer requirements of an electric shaver into components. This QFD shows the combined results of the two phases.

Step 3. Create many-valued context

Table 2 shows many-valued context for electric shavers representing the major distinguishing characteristics. The electric shavers encompass many of the characteristics of a typical product family. Performance characteristics adopted are head type, blade type, power source, waterproof, and special functions.

	Compoments						
Customer requirements	# of head	Blade type	Use bettery	Chargeable	Waterproof fn	Trimmer	Lift
Cut smooth	x	x					x
Portable			x				
Use longer time				x			
Use in Bath room					x		
Trim function						x	x

Figure 4. QFD transforming CR to Components.

Step 4. Create formal context

The many-valued context is transformed into formal context, providing the viewpoint necessary to analyze the data. To begin data exploration, the rough view is required in order to get an overview. The second line of Table 3 shows the introduction of a new attribute for the formal context.

Step 5. Create concept lattice

The formal concepts of a context are ordered by the *sub*-and *super*-concept relation. The set of all formal concepts ordered by such relations form a concept lattice. Using a formal context, a concept lattice for the design artifact is created based on FCA. The concept developed from the formal context group clarifies classes at the same level in the subsumption hierarchy.

A concept lattice for an electric shaver is generated in Figure 5. It can be transformed into a partial order in a straightforward way. According to the sequence of a chosen category, the structure of the concept lattice can take different shapes. AS the product family ontology is constructed based on the primitive component, the shape of concept lattice does not influence the ontology modeling.

Component	Lowtech	High tech
Head type	2 head	3 head
Blade type	Standard rotary	Speed XL
Power source	Battery	Battery & charge
Water proof	No	Yes
Function	Lift & cut	Separate lift/trimmer

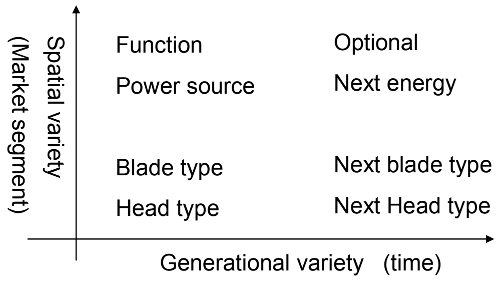


Figure 3. Component type of electric shaver and generational variety and spatial variety.

Table 2. Many-valued characteristics of electric shaver.

Model	Image	Head type	Blade type	Power source	Waterproof	Special functions*
HQ40		2 head	Standard rotary	Battery	No	S1
HQ481		2 head	Standard rotary	Battery + Charge	No	S2
HQ5710		2 head	Standard rotary	Battery	Yes	S1
HQ6675		3 head	Standard rotary	Battery + Charge	Yes	S2
HQ7290		3 head	Standard rotary	Battery + Charge	Yes	S3
HQ560		2 head	Standard rotary	Battery + Charge	Yes	S4
HQ9070		2 head	Speed XL	Battery + Charge	Yes	S5
HQ8150		3 head	Speed XL	Battery + Charge	Yes	S5
HQ9160		3 head	Speed XL	Battery + Charge	Yes	S5

Note) * Special functions; S1: Lift&Cut/Separate lift, S2: Lift&Cut/Separate lift/Trimmer, S3: Lift&Cut/Volume shaving/Separate lift/Trimmer, S4: Lift, S5: Superlift/Volume shaving/Separate lift/Trimmer.

Table 3. Formal context from many-valued context.

model	Head type		Blade type		Power source		Waterproof		Special functions*				
	2 head	3 head	Standard rotary	Speed XL	Battery	Battery + charge	No	Yes	S1	S2	S3	S4	S5
HQ40	X		X		X		X		X	X			
HQ481	X		X			X	X		X	X	X		
HQ5710	X		X		X			X	X	X			
HQ6675		X	X			X		X	X	X	X		
HQ7290		X	X			X		X	X	X	X		X
HQ560	X		X			X		X	X	X	X		
HQ9070	X			X		X		X		X	X	X	X
HQ8150		X		X		X		X		X	X	X	X
HQ9160		X		X		X		X		X	X	X	X

Note) * Special function; S1: Lift&cut, S2: separate lift, S3: trimmer, S4: Super lift, S5: volume shaving.

Step 6. Product variety ontology modeling

Following the concept lattice and product class structure, product ontology can be constructed. The initial product variety hierarchy formed by FCA is enriched by capturing various constraints, characteristics, and relationships between different classes and attributes. Components are declared as individuals that belong to various classes. Component class and lower class are identified.

In order to develop ontology of product variety, components from each product is required to create a class. This requires disassembly of each product into the component level. The relationship between each component is analyzed and compared. FCA process will be applied to each component level and will constitute the product family ontology. Product variety ontology modeling will be processed as the following sequence.

3.1 Decomposition of Product Family

To find a component lexicon of the electric shaver, the product is disassembled to the end component. A component is defined as a part which can not be further disassembled. Figure 6 represents the decomposition of four electric shavers to the end components and their component name respectively.

3.2 Component Lexicon

After the product is disassembled, all the terms associated with each component are collected in a dictionary format, including any synonyms of the component. In the given example, 15 components are identified in case of HQ40 model.

In this model, the lexicon set describing 'blade of HQ40 model' can be L_1 = {HQ40 blade, blade of HQ40, blade for HQ40, etc}. Through the lexicon pruning process, a single term is assigned to a component to identify the component uniquely. Finally, L_1 can be reduced into L_1 = {HQ40 blade}. After applying this process to the product family, the lexicon set of blade class for four models is L = {L_1, L_2, L_3, L_4} = {HQ40 blade, HQ481 blade, HQ7290 blade, HQ9070 blade}.

3.3 Formal Context Composition

Formal context for the 'blade' component can be constructed as a table format. The rows describe the classes and the columns represent properties of the component. Adopting three properties (size, material, assembly) from the product commonality index (Kota *et al*., 2000), a total of four properties are chosen and compared in Table 4. They are product type, part size and shape factor, materials and manufacturing process factor,

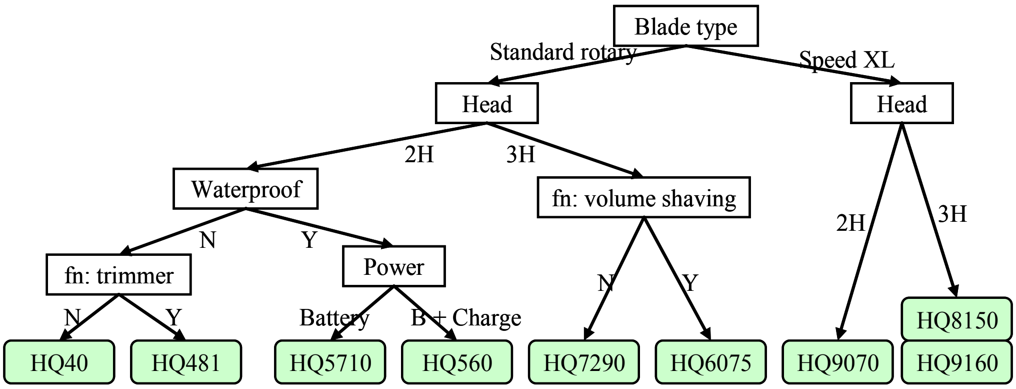
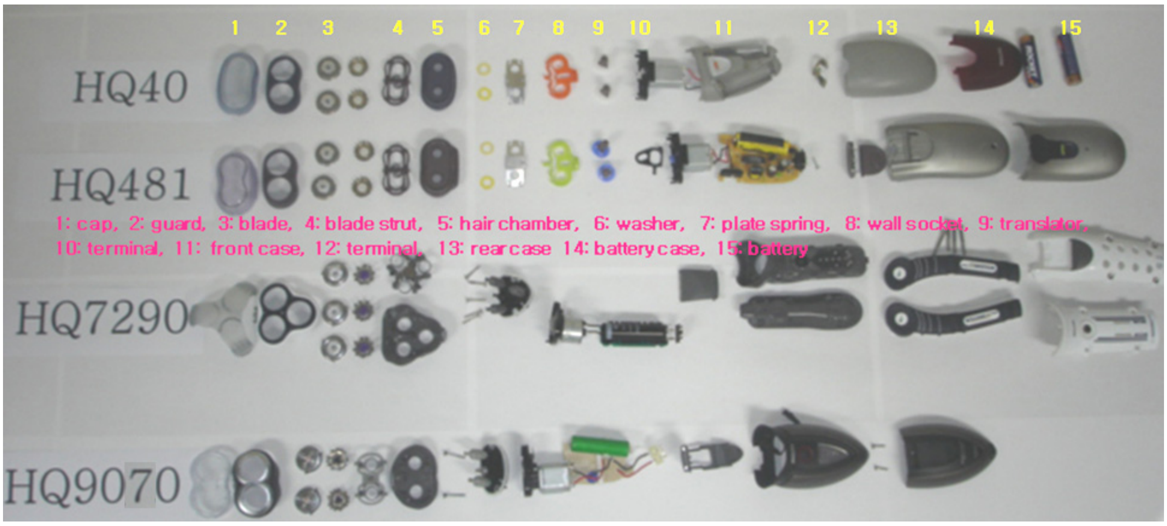


Figure 5. Concept lattice for electric shaver.


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
HQ40
HQ481
HQ7290
HQ9070
1: cap, 2: guard, 3: blade, 4: blade strut, 5: hair chamber, 6: washer, 7: plate spring, 8: wall socket, 9: translator,
10: terminal, 11: front case, 12: terminal, 13: rear case 14: battery case, 15: battery

Note) Part name: 1: cap, 2: guard, 3: blade, 4: blade strut, 5: hair chamber, 6: washer, 7: plate spring, 8: wall socket, 9: translator, 10: terminal, 11: front case, 12: terminal, 13: rear case, 14: battery case, 15: battery.

Figure 6. Decomposition of four electric shavers to the end component and their component name.

and parts assembly and fastening scheme factor. If the properties are the same across the four blade types, '1' is assigned. If they are different to each other, '2' or '3' is assigned.

Table 4. Context table of e-shaver: blade class.

	Type	Size	Material	Assembly
HQ40 blade	Rotary	1	1	1
HQ481 blade	Rotary	1	1	1
HQ7290 blade	Rotary	2	2	1
HQ9070 blade	SpeedXL	3	3	1

3.4 Concept Hierarchy Formation

From the formal context, concept hierarchy is created regarding blade types. The concept hierarchy represents a partially ordered structure or the concept lattice induced by the relationship between the classes and the properties. Figure 7 represents concept hierarchy for blade class component.

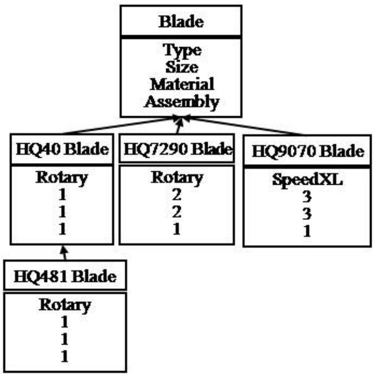


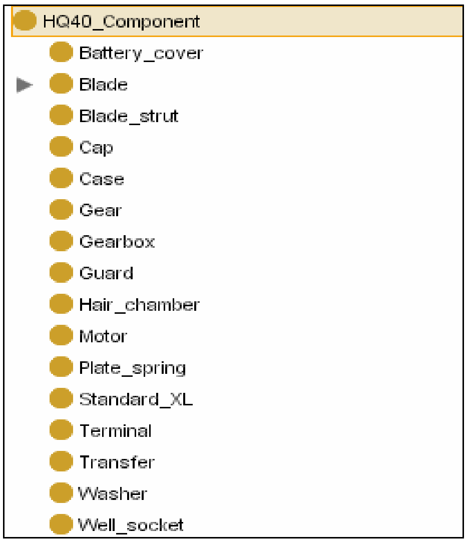
Figure 7. 'Blade class' component hierarchy.

3.5 Product Ontology Formation

Using the component hierarchy, product ontology can be created utilizing the blade class. The component in a sub-class inherits all the properties of the parent class. All components from HQ40 model are shown in Figure 8.

Implementation of product ontology is performed using Protégé 2000. Protégé-2000 is an integrated software tool used to develop ontologies by system developers and domain experts. Protégé-2000 is designed to allow developers to reuse domain ontologies and problem-solving methods, thereby shortening the time needed for development and program maintenance. Several applications can use the same domain ontology to solve different problems, and the same problem-solving method can be used with different ontologies. It is currently being used in any field where the concepts can be modeled as a class hierarchy. In the software, a singular instance can be used on the level of a class definition, and a class can be stored as an instance.

Figure 9 shows the overall screen of electric shaver ontology using Protégé 2000. Component class and its subclass are displayed in the Figure. Properties included in the upper class are also inherited into corresponding lower component.



- HQ40_Component
  - Battery_cover
  - ▶ Blade
  - Blade_strut
  - Cap
  - Case
  - Gear
  - Gearbox
  - Guard
  - Hair_chamber
  - Motor
  - Plate_spring
  - Standard_XL
  - Terminal
  - Transfer
  - Washer
  - Well_socket

Figure 8. Component class and lower class.

4. CONCLUSIONS

This paper synthesizes three distinct ideas, which are product family, FCA and ontology, into a product variety ontology modeling methodology based on FCA. The ideas of product family, ontology and FCA are inve-stigated in depth. The construction of product family ontology helps product developers reduce the efforts for recreating product ontology and minimize the developer's intervention for product analysis. The generated ontology provides a hierarchical clustering for the target product.

The proposed method has been implemented in the domain of an electric shaver product family. The electric shaver product family is disassembled and its product architecture is analyzed. During this process, the class hierarchy, its properties, and instances of the product family are analyzed using OWL.

In the past decade, a successful product family has led the global market very successfully. Sony's Walkmans, One time use cameras of Kodak and Fuji, HP's printers, Gillette's razors, Black Decker's tools, Brown's coffee makers, Volkswagen's A-platform are a few of the typical examples for successful product family. Recent product development teams in the electronics areas such as cell phones, notebook computers, PMPs adopt product family ideas to implement their product variety.

Through the development of product variety ontology in OWL format, the common design vocabularies of a product family can be constructed, which facilitate the

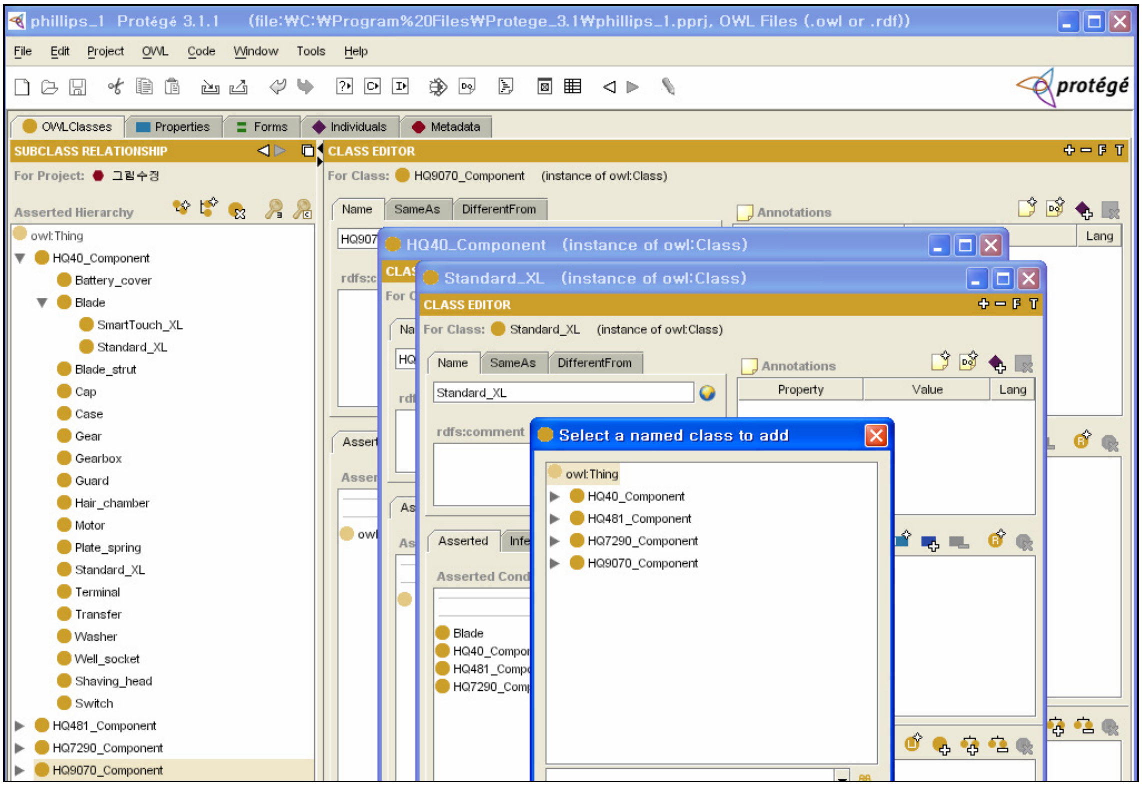


Figure 9. Ontology of electric shaver using protégé-2000.

sharing and reuse of design data across different products and designers even in a distributed location. When a designer wants to develop a new product, he can get all the properties of the product simply by inputting a new instance of the class.

Another contribution of product variety modeling is related to the inherent properties of the ontology. They are communication, reasoning, representation, sharing and reuse of knowledge. Once product ontology is constructed based on the concept analysis, the knowledge for product representation scheme in the design repositories will facilitate the product family design for mass customization. In detail, parts, components, subassemblies and information can be shared and reused among family of products in order to reduce costs, lead time, complexities and efforts.

Based on this analysis of product variety modeling, further research is suggested in the following topics. First, a product variety modeling scheme should be generalized to incorporate the standardization and modularization in the product platform. Second, FCA and ontology should be further utilized in the process of product variety modeling.

ACKNOWLEDGEMENT

This research was supported by Kyungsung Uni-ver-sity Research Grants in 2008.

REFERENCES

Brown, D. (2003), Functional, Behavioral and Structural Features, *Proceedings of the DETC 2003 ASME Design Engineering Technical Conferences*, DETC2003/DTM-48684, Chicago, Illinois.

Eijck, J. and Zwarts, J. (2004), *Formal Concept Analysis and Prototypes*, working paper.

Ishii, K., Ichimura, T., and Hiraki, S. (2003), Analysis of Information Behavior in the Determination of Product Specifications Based on a Conjoint Measurement Approach and Fusion Model, *Industrial Engineering and Management Systems*, **2**(1), 55-62.

Kim, K. and Kim, D. (2008), Research Issues in Robust QFD, *Industrial Engineering and Management Systems*, **7**(2), 93-100.

Koga, T. and Aoyama, K. (2004), Product Behavior and Topological Structure Design System by Step-by-Step Decomposition, *Proceedings of DETC'04 (Design Engineering Technical Conferences)*, DECT2004-57513, September 28-October 2, Salt Lake City, Utah.

Kota, K., Sethuraman, K., and Miller, R. (2000), A Metric for Evaluating Design Commonality in Product Families, *Journal of Mechanical Design*, **122**, 403-410.

Martin, M. V. and Ishii, K. (2003), Design for Variety: Developing Standardized and Modularized Product

Platform Architectures, *Research in Engineering Design*, **13**, 213-235.

Min, S., Matsuoka, S., and Muraki, M. (2004), Priority Assignment Procedure in Multi-Product Disassembly, *Industrial Engineering and Management Systems*, **3**(1), 12-21.

Nanda, J., Thevenot, H., Simpson, T., and Kumara, S. (2004), Exploring Semantic Web Technologies for Product Family Modeling, *Proceedings of DETC '04 (Design Engineering Technical Conferences)*, DE CT2004-57683, September 28-October 2, Salt Lake City, Utah.

Priss, U. (2005), Formal Concept Analysis in Information Science, *Annual Review of Information Science and Technology*, Edited by Blaise Cronin, 1-17.

Simpson, T. W., Maier, R. A., and Mistree, F. (2001), Product Platform Design: Method and Application, *Research in Engineering Design*, **13**, 2-22.

Ulrich, K. T. and Eppinger, S. D. (2003), *Product Design and Development*, Third edition, McGraw-Hill, Inc, Singapore.