Knowledge Discovery to Support Product Family Design

Seung Ki Moon, Timothy W. Simpson, and **Soundar R.T. Kumara**
Department of Industrial & Manufacturing Engineering
Penn State University, University Park, PA 16802 USA

Abstract

Sharing and reusing product design knowledge can help reduce cost and lead time when developing new products and facilitate product family design. Knowledge associated with product design can be represented by combining constraints, functions, rules, and facts. An appropriate representation scheme for products and their design is important to share and reuse the knowledge effectively. The objective in this research is to develop a methodology for knowledge discovery related to product design using an ontology and data mining techniques. An ontology consists of a set of concepts or terms and their relationships that describe some area of knowledge or build a representation of it. An ontology can also be used to build a taxonomy representing products in a database such as a design depository. Data mining can be used in the process of extracting valid, previously unknown, and easily interpretable information from large databases. Fuzzy clustering is employed to determine initial clusters based on the similarity among the functional features of the products. Based on the results of clustering, knowledge related to product family design is identified through association rules. We apply the proposed methodology to develop and then utilize design knowledge for a family of power tools.

Keywords

Knowledge Discovery, Data Mining, Ontology, Product Family Design

1. Introduction

In today's competitive market, companies are increasing their efforts to reduce cost and time for developing new products while satisfying individual customer needs [1]. Many companies strive to maximize resource utilization by sharing and reusing distributed design knowledge and information when developing new products. Product family planning is a way to achieve mass customization by allowing highly differentiated products to be developed around a platform while targeting individual products to distinct market segments [2].

Data mining has been defined as the process of extracting valid, previously unknown, and easily interpretable information from large databases in order to improve and optimize engineering design and manufacturing process decisions [3, 4]. In mass customization, data mining can be used to help identify customer needs, to find relationships between customer needs and functional requirements, and to cluster products based on functional similarity to facilitate modular design [4]. At the conceptual design stage, data mining can aid decision-making when selecting design concepts by extracting design knowledge and rules, clustering design cases, and exploring interactively conceptual designs in large product design databases [4]. An ontology consists of a set of concepts or terms and their relationships that describe some area of knowledge or build a representation of it [5].

In this paper, we describe a novel methodology for knowledge discovery to support product family design using data mining techniques, namely, clustering and association rules. In particular, to define the relationship between functional hierarchies in a product effectively, an appropriate representation scheme must be adopted for the products. We use the Techspecs Concepts Ontology (TCO) to represent the functions of a product as functional hierarchies [6]. Fuzzy c-means clustering is employed to partition product functions into subsets for identifying modules in a given product family. Rules related to design knowledge are developed through association rule mining. Based on the results of using association rules and clustering, the design knowledge is used to help identify a platform and modules for a product family.

The remainder of this paper is organized as follows. Section 2 presents product representation using TCO and a coding approach. Section 3 describes the proposed methodology for knowledge discovery to support product family design using data mining. Section 4 gives a case study using a family of five power tools. Closing remarks and future work are presented in Section 5.

2. Product Representation using Ontology

We assume that a product can be defined by its modules that consist of specific functions, and functions are achieved by the combination of the module attributes (see Figure 1). To effectively define the relationship between functional hierarchies in a product, it is important to adopt an appropriate representation scheme for the products.

We use the Techspecs Concept Ontology (TCO) to represent products and components [6]. TCO provides functional representation-based semantics of products or components to better reflect customers' preferences and market needs. Using TCO, we can develop a module-based functional hierarchy for a product as shown in Figure 1.

Suppose that a product family consists of l products, $PF = (P_1, P_2, \ldots, P_l)$ and a product consists of m_i modules, $P_i = (\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,m_i})$, where $\mathbf{x}_{i,j}$ is a module j in product i and consists of a vector of length n_m, $\mathbf{x}_{i,j} = (x_{i,j,1}, x_{i,j,2}, \ldots, x_{i,j,n_m})$, and the individual scalar components $x_{i,j,k}$ (k=1, 2,…, n_m) of a module $\mathbf{x}_{i,j}$ are called *functional features*. The functional feature consists of several attributes $a_{i,j,k,t}$ (t=1, 2, …, t_n), representing the function, $x_{i,j,k} = (a_{i,j,k,1}, a_{i,j,k,2}, \ldots, a_{i,j,k,t_n})$, where t_n is the number of attributes represented by TCO. For example, we define five functional feature attributes that are: description, input energy, output energy, operand, and medium. Figure 1 shows the functional hierarchy for a drill and the hierarchy level for representing a product.

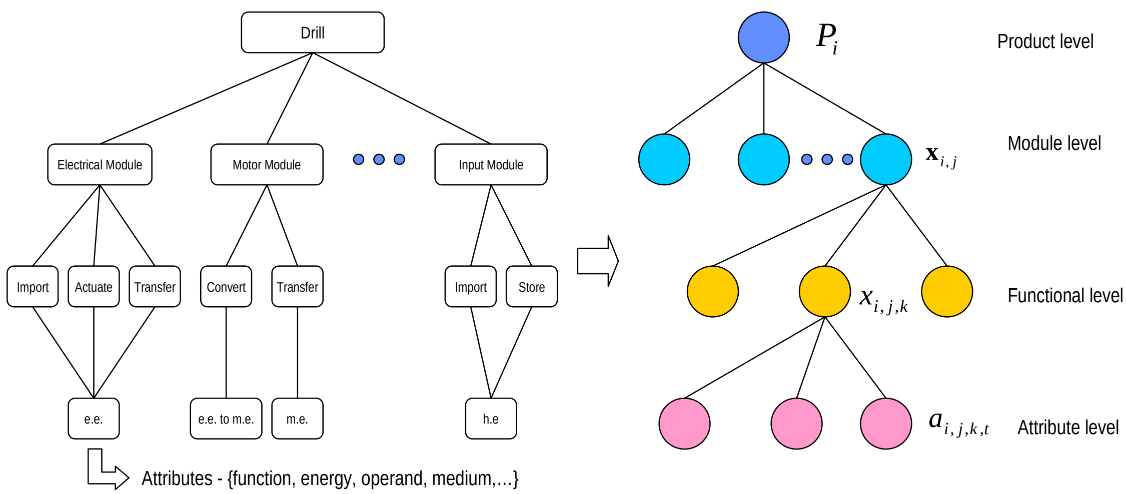


Figure 1: Functional Hierarchy and Product Representation using TCO

In this paper, a coding approach is used to represent the attributes of functional features for a given clustering method. As shown in Table 1, each attribute takes a different code (number). Functional features in Table 1 are developed based on the functional basis proposed by Hirtz, et al. [7]. For instance, if the attributes of a functional feature consist of *convert* (description), *electronic energy* (input energy), *mechanical energy* (output energy), *force* (operand), and *shaft* (medium), then the codes for the attributes are 13, 5, 9, 2, and 2, respectively.

Table 1: Attribute Codes for Functional Features

Function description	Code	Energy	Code	Operand	Code	Medium	Code
Separate	1	Human	1	Effort	1	Wire	1
Distribute	2	Acoustic	2	Force	2	Shaft	2
Import	3	Biological	3	Pressure	3	Gear	3
Export	4	Chemical	4	Affinity	4	Bit	4
Transfer	5	Electrical	5	Electromotive force	5	Sander base	5
Guide	6	Electromagnetic	6	Intensity	6	Magazine	6
Couple	7	Hydraulic	7	Magnetomotive force	7	Grip	7
Mix	8	Magnetic	8	Torque	8	Blade (circle)	8
Actuate	9	Mechanical	9	Temperature	9	Blade (vertical)	9
Regulate	10	Pneumatic	10	Flow	10	Battery	10
Change	11	Radioactive/Nuclear	11	Velocity	11	Nail hitter	11
Stop	12	Thermal	12	Particle velocity	12	Vibration generator	12
Convert	13			Volumetric flow	13		
Store	14			Reaction rate	14		
Supply	15			Current	15		
Sense	16			Magnetic flux rate	16		
Indicate	17			Angular velocity	17		
Process	18			Linear velocity	18		
Stabilize	19			Mass flow	19		
Secure	20			Decay rate	20		
Position	21			Heat flow	21		
				Rotation	22		

3. Data Mining for Knowledge Discovery

3.1 Fuzzy Clustering

Functional decomposition for a product is represented as a hierarchical structure. A hierarchical clustering method can classify a set of objects by measuring the similarity between objects [8]. Because heuristic methods used to define a module may provide overlapping or non-crisp boundaries among module clusters [9], the results of traditional clustering approaches are not appropriate to define clusters as modules in product design. Fuzzy clustering approaches can use fuzziness related to product design features and provide more useful solutions [10, 11]. In this paper, we employ fuzzy c-means clustering (FCM) [12] to determine clusters for identifying modules in the product family. FCM is a clustering technique that is similar to k-means but uses fuzzy partitioning of data that is associated with different membership values between 0 and 1. Since FCM is an iterative algorithm, the aim in FCM is to find cluster centers that minimize a dissimilarity function as follows.

Let X_k for k = 1, 2, …, n be a functional feature and a d-dimensional vector (d is the number of attributes), and u_{ik} the membership of X_k to the i-th cluster (i=1, 2, …, c). The u_{ik} representing a fuzzy case is between 0 and 1. For example, if u_{ik} = 0, u_{ik} has non-membership to cluster i, and if u_{ik} = 1, then it has full membership. Values in between 0 and 1 indicate fractional membership. Generally, FCM is defined as the solution of the following minimization problem [12]:

$$J_{FCM}(U,V) = \{\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m \left\| X_k - v_i \right\|^2\} \tag{1}$$

subject to:

$$\sum_{i=1}^{c} u_{ik} = 1 \quad \text{for all } k \tag{2}$$

$$u_{ik} \in [0, 1] \tag{3}$$

where v_i is a cluster center of the i-th cluster that consists of a d-dimensional vector, and m is a parameter ($m \geq 1$) that plays a central role and indicates the fuzziness of clusters. An algorithm for solving this problem is introduced in Refs. [12, 13]. This FCM algorithm does not ensure that it converges to a global optimal solution; however, it always converges to a local optimum that may lead to a different local minima according to different initial cluster centers [12, 13].

The cluster number can be considered as the number of modules. Based on a maximum membership value in clusters, functional features are assigned to clusters that are considered as modules. In this paper, we categorize the results of clusters into four modules based on a module value ($0 \leq \theta_1 \leq 1$) and a ratio ($0 < \theta_2 \leq 1$): (1) unique module, (2) common module, (3) redesign module, and (4) sub-common module. The module value represents the functional similarity among modules in a cluster and is calculated based on the average of the function's membership values in a cluster. If θ_1 = 1, modules have the same functional features, and if θ_1 = 0, then they are totally different. The ratio indicates the number of total products that include a particular module. If θ_2 = 1, a module can be applied to all products in a product family, and otherwise θ_2 is in proportion to the number of products using the module. A unique module is based on distinctive functions within a product family and cannot be replaced by those in the different module to fulfill their tasks. A common module is based on common functions within a product family and can be shared. A redesign module can be a common module if redesigned to increase the functional similarity. A sub-common module can be a common module or a unique module based on the tradeoff between production and design cost. Figure 2 shows four regions for clusters with two thresholds that indicate a standard for determining a module category according to designer's preference or knowledge.

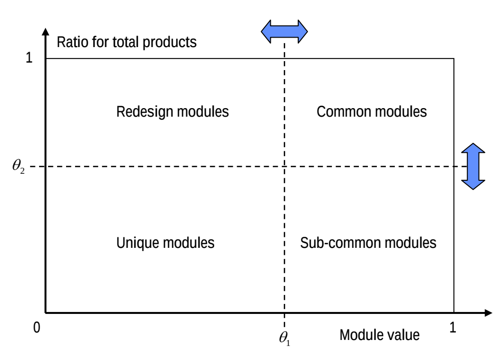


Figure 2: Module Categories based on the Result of Clustering

3.2 Association Rules

An association rule describes an interesting relationship between attributes of different modules [14, 15]. Given a set of transactions, where each transaction is a set of attributes, an association rule is noted as $A \Rightarrow B$, where A and B are sets of attributes. The association rule $A \Rightarrow B$ indicates that transactions that contain attribute A tend to contain attribute B. Support and confidence are introduced to assess the quality of the extracted rules [14]. The support of an attribute A in a set S of transaction data means the probability of transaction data containing attribute A. The confidence of $A \Rightarrow B$ represents the probability of attribute B occurring in S if attribute A occurs in S. An association rule with high confidence and support is called strong and is potentially useful for product design [14].

In association rule mining, transaction data is needed to develop rules related to product design. Based on the results of clustering and TCO, we can develop transaction data that consists of several properties in the hierarchical functional relationship. For example, we can generate transaction data that is composed of module category, module level, functional level, and attribute level in each product. We use the *Apriori* algorithm to generate association rules that use frequent item sets to define the association rules [16]. A designer can extract important design features from association rules, which are classified and translated into knowledge and rules for product design.

3.3 Extensions to Product Family and Platform Design

In this section, we address how design knowledge extends to product family and platform design. Knowledge can be represented as constraints, functions, rules, and facts that are associated with product design. In general, a product family is a group of related products based on a product platform, facilitating mass customization by providing a variety of products for different market segments cost-effectively [17]. A successful product family depends balancing the tradeoffs between the economic benefits and performance losses incurred from having a platform. We can use the design knowledge to identify a platform that consists of common modules and determine design attributes related to the platform during initial and conceptual design phase. In addition, the design knowledge presented by TCO can provide information and specific combinations of related modules and components based on assembly relationship information. It is possible that a designer can also search all of the related components in a module based on this design knowledge. This information plays an important role in conceptual design; therefore, the design knowledge can help develop an effective product platform and product family as demonstrated next.

4. Case Study

To demonstrate the proposed methodology, a power tool family is investigated that consists of five distinct cordless power tools (see Table 2). The products representation for the five tools was developed using TCO. Table 2 shows the 75 functional features of the selected five products. The attributes of these functional features were coded using the values listed in Table 1.

FCM was then used to determine modules for the five products. In this paper, c=13 was selected as the optimal cluster number for determine modules for the products based on the partition coefficient [18]. Then, clustered results were categorized into four modules based on two thresholds in Section 3.1. Suppose a module value and a ratio as (0.7, 0.8). Figure 3 shows the result of the categorized modules for 13 clusters.

Based on the clustered results and TCO, we developed a set of transaction data consisting of category, module, function, and energy for each functional feature. Figure 4 shows the process of association rule mining and the rules generated using Magnum Opus demo version 3.0[1]. The rules can be translated into design knowledge that is used to define a platform and modules for a product family. For example, using the design knowledge, a designer can determine a platform based on common modules, research components related to the platform, and understand characteristics for modules. The current common modules in the actual product family are a battery module and a batter terminal connector in an input module. Based on the knowledge for five tools family, we can determine a platform that consists of functions related to an electronic module, a motor module, a battery module, and an input module. Comparing this to the current platform of five products, we can increase the number of common modules based on common functional features.

5. Closing Remarks and Future Work

In this paper, we proposed a new methodology for knowledge discovery to support a product family using an ontology and data mining techniques. Fuzzy c-means clustering was employed to cluster the functional features of

[1] http://www.rulequest.com

products based on the similarity among them. The resulting clusters can be categorized into four modules that are used to help identify a platform and modules for the product family. Based on the result of clusters and TCO, design knowledge for product family design was developed through association rule mining. We demonstrated the proposed methodology to discover design knowledge using a case study involving a family of five power tools.

Since the proposed methodology uses functional features defined by an ontology, functional requirements related to customer needs can be considered in the conceptual design phase. Using the proposed method, we can determine a module-based platform and modules that can be used for product family design. Therefore, the proposed methodology can help design a variety of products within a product family. Future research efforts will focus on improving the methodology for reflecting customer needs and functional requirements, selecting a proper number of clusters for determining modules, and expanding its application to large databases or design repositories related to product design.

Table 2: Product Representation for Power Tool Family

Product	Module	Functional features
Circular saw 	$x_{1,1}$	1(3, 5, 5, 15, 1), 2(9, 5, 5, 15, 1), 3(5, 5, 5, 15, 1)
	$x_{1,2}$	1(13, 5, 5, 15, 1), 2(5, 9, 9, 2, 2)
	$x_{1,3}$	1(4, 0, 5, 15, 10), 2(5, 5, 5, 15, 10), 3(14, 5, 5, 15, 10)
	$x_{1,4}$	1(3, 1, 1, 2, 8), 2(14, 9, 9, 2, 8), 3(4, 9, 9, 2, 8)
	$x_{1,5}$	1(3, 1, 1, 2, 7), 2(14, 1, 1, 2, 7)
	$x_{1,6}$	1(3, 9, 9, 8, 2), 2(5, 9, 9, 8, 2)
Drill 	$x_{2,1}$	1(3, 5, 5, 15, 1), 2(9, 5, 5, 15, 1), 3(5, 5, 5, 15, 1)
	$x_{2,2}$	1(13, 5, 9, 2, 2), 2(5, 9, 9, 2, 2)
	$x_{2,3}$	1(4, 0, 5, 15, 10), 2(5, 5, 5, 15, 10), 3(14, 5, 5, 15, 10)
	$x_{2,4}$	1(3, 1, 1, 2, 4), 2(14, 9, 9, 2, 4), 3(4, 9, 9, 2, 4)
	$x_{2,5}$	1(3, 1, 1, 2, 7), 2(14, 1, 1, 2, 7)
	$x_{2,6}$	1(5, 9, 9, 8, 3), 2(11, 9, 9, 8, 3)
Jig saw 	$x_{3,1}$	1(3, 5, 5, 15, 1), 2(9, 5, 5, 15, 1), 3(5, 5, 5, 15, 1)
	$x_{3,2}$	1(13, 5, 9, 2, 2), 2(5, 9, 9, 2, 2)
	$x_{3,3}$	1(13, 9, 9, 8, 3), 2(5, 9, 9, 8, 3)
	$x_{3,4}$	1(4, 0, 5, 15, 10), 2(5, 5, 5, 15, 10), 3(14, 5, 5, 15, 10)
	$x_{3,5}$	1(3, 1, 1, 2, 7), 2(14, 1, 1, 2, 7)
	$x_{3,6}$	1(3, 1, 9, 2, 9), 2(14, 9, 9, 2, 9), 3(4, 9, 9, 2, 9)
Nailer 	$x_{4,1}$	1(3, 5, 5, 15, 1), 2(9, 5, 5, 15, 1), 3(5, 5, 5, 15, 1)
	$x_{4,2}$	1(13, 5, 9, 2, 2), 2(5, 9, 9, 2, 2)
	$x_{4,3}$	1(13, 9, 9, 3, 11), 2(5, 9, 9, 3, 11)
	$x_{4,4}$	1(4, 0, 5, 15, 10), 2(5, 5, 5, 15, 10), 3(14, 5, 5, 15, 10)
	$x_{4,5}$	1(3, 1, 1, 2, 7), 2(14, 1, 1, 2, 7)
	$x_{4,6}$	1(3, 1, 9, 2, 6), 2(4, 9, 9, 2, 6)
Sander 	$x_{5,1}$	1(3, 5, 5, 15, 1), 2(9, 5, 5, 15, 1), 3(5, 5, 5, 15, 1)
	$x_{5,2}$	1(13, 5, 9, 2, 2), 2(5, 9, 9, 2, 2)
	$x_{5,3}$	1(13, 9, 9, 22, 12), 2(5, 9, 9, 22, 12)
	$x_{5,4}$	1(4, 0, 5, 15, 10), 2(5, 5, 5, 15, 10), 3(14, 5, 5, 15, 10)
	$x_{5,5}$	1(3, 1, 1, 2, 7), 2(14, 1, 1, 2, 7)
	$x_{5,6}$	1(3, 1, 9, 2, 5), 2(14, 9, 9, 2, 5), 3(4, 9, 9, 2, 5)

Cluster	Circular saw	Drill	Jig saw	Nailer	Sander
1					$x_{5,3,1}$ (0.87), $x_{5,3,2}$ (0.79)
2	$x_{1,2,2}$ (0.99), $x_{1,6,1}$ (0.33), $x_{1,6,2}$ (0.34)	$x_{2,2,2}$ (0.99), $x_{2,4,3}$ (0.67), $x_{2,6,1}$ (0.3)	$x_{3,2,2}$ (0.99), $x_{3,3,2}$ (0.3)	$x_{4,2,2}$ (0.99)	$x_{5,2,2}$ (0.99)
3	$x_{1,4,3}$ (0.48)		$x_{3,6,1}$ (0.23), $x_{3,6,3}$ (0.46)	$x_{4,3,2}$ (0.38), $x_{4,6,1}$ (0.22), $x_{4,6,3}$ (0.46)	$x_{5,6,1}$ (0.21), $x_{5,6,3}$ (0.36)
4	$x_{1,3,3}$ (1)	$x_{2,3,3}$ (1)	$x_{3,4,3}$ (1)	$x_{4,4,3}$ (1)	$x_{5,4,3}$ (1)
5	$x_{1,4,2}$ (0.71)		$x_{3,6,2}$ (0.97)	$x_{4,3,1}$ (0.92)	
6				$x_{4,6,2}$ (0.74)	
7	$x_{1,4,1}$ (0.99), $x_{1,5,1}$ (1)	$x_{2,4,1}$ (0.89), $x_{2,5,1}$ (1)	$x_{3,5,1}$ (1)	$x_{4,5,1}$ (1)	$x_{5,5,1}$ (1)
8	$x_{1,3,1}$ (0.92), $x_{1,3,2}$ (0.89)	$x_{2,3,1}$ (0.92), $x_{2,3,2}$ (0.89)	$x_{3,4,1}$ (0.92), $x_{3,4,2}$ (0.89)	$x_{4,4,1}$ (0.92), $x_{4,4,2}$ (0.89)	$x_{5,4,1}$ (0.92), $x_{5,4,2}$ (0.89)
9		$x_{2,4,2}$ (0.85), $x_{2,6,2}$ (0.24)	$x_{3,3,1}$ (0.92)		$x_{5,6,2}$ (0.57)
10	$x_{1,1,1}$ (0.95), $x_{1,1,2}$ (0.77), $x_{1,1,3}$ (1)	$x_{2,1,1}$ (0.95), $x_{2,1,2}$ (0.77), $x_{2,1,3}$ (1)	$x_{3,1,1}$ (0.95), $x_{3,1,2}$ (0.77), $x_{3,1,3}$ (1)	$x_{4,1,1}$ (0.95), $x_{4,1,2}$ (0.77), $x_{4,1,3}$ (1)	$x_{5,1,1}$ (0.95), $x_{5,1,2}$ (0.77), $x_{5,1,3}$ (1)
11	$x_{1,4,3}$ (0.48)		$x_{3,6,1}$ (0.23), $x_{3,6,3}$ (0.46)	$x_{4,3,2}$ (0.38), $x_{4,6,1}$ (0.22), $x_{4,6,3}$ (0.46)	$x_{5,6,1}$ (0.21), $x_{5,6,3}$ (0.36)
12	$x_{1,5,2}$ (1)	$x_{2,5,2}$ (1)	$x_{3,5,2}$ (1)	$x_{4,5,2}$ (1)	$x_{5,5,2}$ (1)
13	$x_{1,2,1}$ (1)	$x_{2,2,1}$ (1)	$x_{3,2,1}$ (1)	$x_{4,2,1}$ (1)	$x_{5,2,1}$ (1)

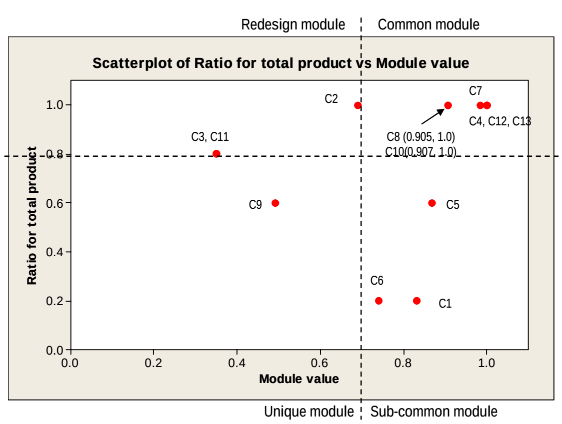
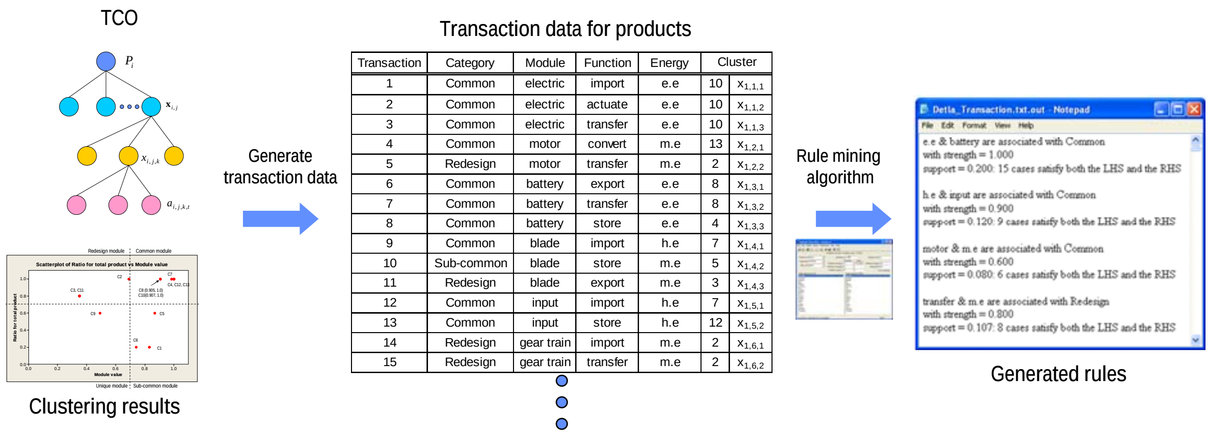


Figure 3: Results of Clustering and Scatter Plot of Categorized Modules



Transaction	Category	Module	Function	Energy	Cluster	
1	Common	electric	import	e.e	10	$x_{1,1,1}$
2	Common	electric	actuate	e.e	10	$x_{1,1,2}$
3	Common	electric	transfer	e.e	10	$x_{1,1,3}$
4	Common	motor	convert	m.e	13	$x_{1,2,1}$
5	Redesign	motor	transfer	m.e	2	$x_{1,2,2}$
6	Common	battery	export	e.e	8	$x_{1,3,1}$
7	Common	battery	transfer	e.e	8	$x_{1,3,2}$
8	Common	battery	store	e.e	4	$x_{1,3,3}$
9	Common	blade	import	h.e	7	$x_{1,4,1}$
10	Sub-common	blade	store	m.e	5	$x_{1,4,2}$
11	Redesign	blade	export	m.e	3	$x_{1,4,3}$
12	Common	input	import	h.e	7	$x_{1,5,1}$
13	Common	input	store	h.e	12	$x_{1,5,2}$
14	Redesign	gear train	import	m.e	2	$x_{1,6,1}$
15	Redesign	gear train	transfer	m.e	2	$x_{1,6,2}$

Figure 4: The Process of Associate Rule Mining

Acknowledgments

This work was funded by the National Science Foundation through Grant No. IIS-0325402. Any opinions, findings, and conclusions or recommendations presented in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Muffatto, M. and Roveda, M., 2000, "Developing Product Platforms: Analysis of the Development Process," *Technovation*, vol. **20**, No. 11, pp. 617-630.

[2] Simpson, T. W., Siddique, Z., and Jiao, J., 2005, *Product Platform and Product Family Design: Methods and Applications*. New York, YN: Springer.

[3] Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., 1996, *Advances in Knowledge Discovery and Data Mining*. Cambridge, MA: AAAI Press/MIT Press.

[4] Braha, D., 2001, *Data Mining for Design and Manufacturing: Methods and Applications*. Dordrecht, Netherlands: Kluwer Academic Publishers.

[5] Swartout, W. and Tate, A., 1999, "Ontologies," *IEEE Transactions on Intelligent Systems*, vol. **14**, No. 1, pp. 18-19.

[6] Moon, S. K., Kumara, S. R. T., and Simpson, T. W., 2005, "Knowledge Representation for Product Design Using Techspecs Concept Ontology," *The IEEE International Conference on Information Reuse and Integration*, Las Vegas, NV, Aug. 15-17, pp. 241-246.

[7] Hirtz, J., Stone, R. B., McAdams, D. A., Szykman, S., and Wood, K. L., 2002, "A functional basis for engineering design: Reconciling and evolving previous efforts," *Research in Engineering Design*, vol. **13**, No. 2, pp. 65-82.

[8] Miyamoto, S., 1990, *Fuzzy sets in information retrieval and cluster analysis*. Dordrecht, Netherlands: Kluwer Academic Publishers.

[9] Stone, R. B., Wood, K. L., and Crawford, R. H., 2000, "A heuristic method for identifying modules for product architectures," *Design Studies*, vol. **21**, No. 1, pp. 5-31.

[10] Xue, D. and Dong, Z., 1997, "Coding and clustering of design and manufacturing features for concurrent design," *Computer in Industry*, vol. **34**, No. 1, pp. 139-153.

[11] Liao, T. W., 2001, "Classification and coding approaches to part family formation under a fuzzy environment," *Fuzzy sets and systems*, vol. **122**, No. 3, pp. 425-441.

[12] Bezdek, J. C., 1981, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York, NY: Plenum.

[13] Torra, V., 2005, "Fuzzy c-means for fuzzy hierarchical clustering," *the IEEE International Conference on Fuzzy Systems*, Reno, NV, May 22-25, pp. 646-651.

[14] Agard, B. and Kusiak, A., 2004, "Data-mining-based methodology for the design of product family," *International Journal of Production Research*, vol. **42**, No. 15, pp. 2955-2969.

[15] Jiao, J. and Zhang, Y., 2005, "Product portfolio identification based on association rule mining," *Computer-Aided Design*, vol. **27**, No. 2, pp. 149-172.

[16] Agrawal, R. and Srikant, R., 1994, "Fast Algorithms for Mining Association Rules in Large Databases," *20th International Conference on Very Large Data Bases*, Santiago de Chile, Chile, pp. 487-499.

[17] Pine, B. J., II, 1993, *Mass Customization: The New Frontier in Business Competition*. Boston, MA: Harvard Business School Press.

[18] Bezdek, J. C., 1974, "Numerical Taxonomy with Fuzzy Sets," *Journal of Mathematics Biology*, vol. **1**, pp. 57-71.