

Chapter 6

Quantifying the Relevance of Product Feature Classification in Product Family Design

Conrad S. Tucker

Abstract The methodology proposed in this chapter aims to address the link between the evolution of product feature relevance and the implications to product platform and product family design. By quantifying relevant/irrelevant product features to be included in next-generation product platform design, designers can identify the stand-alone or platform sharing components required to achieve desired product functionality. A data mining algorithm is introduced that uses time series data (consisting of product features) to determine the *standard*, *nonstandard*, and *obsolete* product features in the design of next-generation products. Product features are then mapped to engineering components/modules by employing data mining Natural Language Processing techniques that quantify the functionality requirements that are needed for a given set of product features. The goal of this work is to demonstrate the value of incorporating evolving product feature trends in the market space directly into product platform and product family sharing decisions.

6.1 Introduction

The increase in globalization and the prevalence of low-cost communication infrastructure present ever-increasing challenges for enterprise decision makers aiming to satisfy customer needs. In recent years, companies have had to consider the impacts of a socially connected digital age in shaping customer preferences and expectations in the market space (Tucker and Kim 2011a). The evolution of product preferences in the market space can be highly dynamic and difficult to capture using traditional customer-driven frameworks employed by design engineers. Mass customization has been proposed as a viable approach to accommodate the diverse

C.S. Tucker (✉)
Industrial Engineering and Engineering Design, The Pennsylvania State University,
University Park, PA, USA
e-mail: cst14@psu.edu

product preferences in the market space. From an engineering perspective, however, mass customization presents the added challenge of establishing design and manufacturing processes to meet the needs of mass customization. Product family design is an enterprise-driven strategy aimed at mitigating the added costs that arise due to product customization. The two design strategies that have been proposed in the product family design literature are the *Bottom-Up* approach and the *Top-Down* approach (Simpson et al. 2001). Commonality indices proposed in the literature investigate component/module sharing strategies for existing products within a product family and are well suited for Bottom-Up product family design. In Top-Down product family design, a product family emerges from an existing market-driven need. The data mining component classification framework in this work will enable designers to identify components that are well suited for sharing in the product family design process through the use of large-scale market-driven product feature preference data.

6.2 Related Work

This section presents work relevant to the three main aspects of this research: (1) Data mining-driven product design, (2) Translating customer needs into engineering targets, and (3) Product platform and sharing decisions.

6.2.1 *Data Mining-Driven Product Design*

Data Mining-Driven Product Design is an emerging field of research aimed at incorporating large-scale data in the design of next-generation products (Braha 2001). Agard and Kusiak employ data mining association rules to cluster product functions in the design of product families (Agard and Kusiak 2004). Tucker and Kim (2008) employ Naive Bayes Classification techniques that enable designers to identify novel product feature combinations in a high dimensional product feature space. Moon et al. (2006) employ data mining Fuzzy c clustering techniques as a platform identification strategy in product family design. Data Mining techniques have been employed by Tucker and Kim to determine the optimal product feature combination for product family optimization (Tucker and Kim 2009; Tucker et al. 2010). Moon proposes a Data Mining framework for extracting design knowledge for product platform and variant design (Wang 2008).

While the aforementioned data mining techniques proposed in the literature aim to address product family design problems, they are static in nature and primarily consider large-scale data at an instant in time, hereby omitting the changes in product feature preferences that may occur in the market space over time. In order to accommodate evolving product trends in the market space, Tucker and Kim propose a temporal product feature classification algorithm that classifies

product features as *Standard*, *Nonstandard*, or *Obsolete*, based on their time series predictive power (Tucker and Kim 2011b). The classification of product components will enable design engineers to determine when to retire certain components (classified as obsolete in the metric), include in the design of a product platform (classified as standard in the metric), or aid in the creation of modules for product variants (classified as nonstandard in the metric).

6.2.2 Translating Customer Needs into Engineering Targets

Quality function deployment (QFD) is a well-established approach employed in the design community for translating customer preference requirements into engineering design targets/functional specifications (Pullmana et al. 2002). A house of quality (HOQ) would be designed, mapping the customer requirements into tangible engineering design targets (Bouchereau and Rowlands 2000). Customer preferences towards certain product features can be weighted through feature rankings acquired through surveys or focus groups (Kwong and Bai 2003). As a result, a QFD matrix can be used to depict the interdependence between customer requirements and the engineering metrics (EM).

The QFD model is highly dependent on the domain expert (engineers) translating the customer wants into engineering metrics. As the complexity of modern technology increases, so does the availability of product features and customization options. The increased product feature space (high dimensional feature space) and the highly dynamic nature of many consumer markets today make traditional translation of customer preferences into engineering metrics cumbersome. Furthermore, the expertise of these techniques is limited to the domain expert(s), making the process highly dependent on a subset of the product development team. By employing a data mining-driven approach to customer preference modeling and then translating the knowledge gained into tangible engineering metrics, design engineers will be able to incorporate market-driven trends during the translation of customer wants into engineering specifications. Instead of relying on survey or focus group feedback in an effort to quantify the evolution of product preferences in the market space, designers can employ the data mining methodology proposed in this chapter as a means of generating predictive models about evolving product feature preferences that can then be used for product family optimization.

6.2.3 Product Platform and Sharing Decisions

A *product platform* can be defined as a set of parameters/features or components that are shared across products within a product family (Simpson et al. 2001). Meyer and Lehnerd provide guidelines for product platform development and encourage companies to design products around a shared platform, rather than

subsequent independent designs (Meyer and Lehnerd 1997). *Commonality* refers to the level of sharing of components/subassemblies, processes, etc. across different products within a family of products (Boas 2008). Commonality therefore has the ability of reducing manufacturing and design costs (by sharing the same component across different products), while concurrently providing the level of product diversity expected within the market space. The trade-off between product commonality and product diversity has been studied extensively in the literature and is discussed in Simpson et al. (2001). de Weck highlights the challenges that exist in determining the extent of product platforming in product family design (Simpson et al. 2006).

Several commonality metrics have been proposed in the literature in an effort to quantify the effects of platform sharing decisions on product family design. For example, Collier proposed the degree of commonality index (DCI) as a way to measure the ratio of common components existing among products within a product family to the total number of components (Collier 1981). A modified version of the DCI called the total constant commonality index (TCCI) has absolute bounds (0–1), hereby making commonality comparisons within and between product families more quantifiable (Wacker and Trelevan 1986). The commonality index (CI) proposed by Martin and Ishii measures the ratio of unique components in a product family and the total components in a product family (Martin and Ishii 1996, 1997). The Percent Commonality Index (%C) measures product commonality within a shared product platform, rather than across product families using a weighted sum of multiple variables for a total commonality scale ranging from 0 (no commonality) to 100 (complete commonality) (Siddique et al. 1998). Another extension of the DCI called the component part commonality index (CI^(C)) takes into account factors such as the cost of each component, product volume, and quantity per operation in determining the effects of component sharing decisions on a product family (Jiao and Tseng 2000). The product line commonality index (PCI) is a departure from traditional commonality indices that penalize broad product variation and instead, penalizes products with nonunique components within a product family (Kota et al. 2000). The generational variety index (GVI) proposed by Martin and Ishii measures the level of redesign work needed for future iterations of a product and helps designers determine which components may change over time (Martin and Ishii 2002). The comprehensive metric for commonality (CMC) is a data intensive approach to product commonality based on the components', size, geometry, material, manufacturing process, assembly, cost, and allowed diversity in the family (Thevenot and Simpson 2006). Alizon et al. (2009) propose a commonality diversity index (CDI) that compares components relating to a specific function(s) and investigates the trade-off between commonality and diversity based on the product family's functional requirements. With a plethora of commonality metrics proposed in the literature, Simpson et al. (2012) approach the product platforming problem by proposing an integrative approach that incorporates a market segmentation grid, the GVI, design structure matrix (DSM), commonality indices, mathematical modeling and optimization, along with multidimensional data visualization tools.

The methodology proposed in this chapter aims to address the link between the evolution of product feature relevance and the implications to product platform and product family design. Specifically, this work aims to:

- Translate a product feature classification from the market-driven domain to the detailed engineering domain.
- Determine the optimal product platform sharing decisions based on the market-driven evolution of product features and customer preferences.

6.3 Methodology

The methodology proposed in this work (Fig. 6.1) aims to guide product family design by linking *Temporal Market-Driven Responses* relating to product feature trends with *Engineering Design Optimization* objectives such as product platforming and commonality decisions. As presented in Sect. 6.2.3, there are well-established metrics for evaluating commonality decisions in product family design. However, temporal, market-driven forces are typically not included in these models.

Market-Driven Responses can be a critical design input to product family design by quantifying the evolution of product features in the market space and identifying product features that are *Standard*, *Nonstandard*, or *Obsolete*. In the proposed methodology, the *Product Domain* in the *Market-Driven Response* step refers to the results of a data mining-driven approach to modeling the relevant/irrelevant product features across a wide array of products existing in that domain.

6.3.1 Level 1: Temporal Market-Driven Preferences

Level 1 of the proposed methodology is based on a knowledge discovery in databases (KDD) framework. KDD is the umbrella term used to describe the

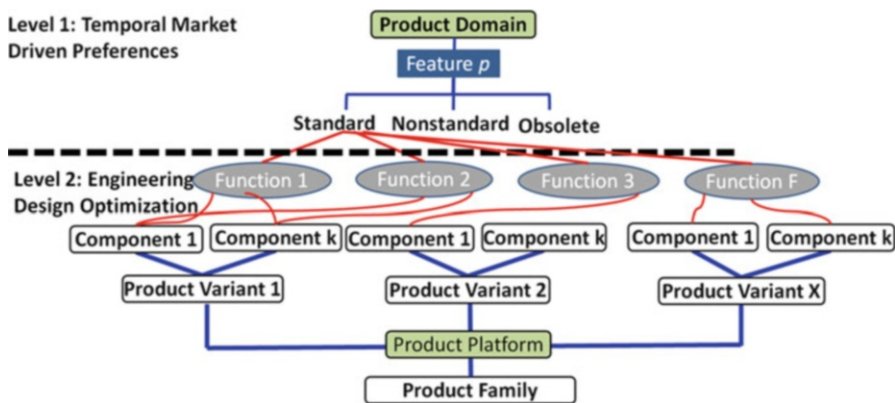


Fig. 6.1 Linking market-driven response with engineering design optimization

Time t_1					Time t_n				
Feature 1	Feature 2	...	Feature p	Class	Feature 1	Feature 2	...	Feature p	Class

Fig. 6.2 Time series product data containing product features and class

sequential steps of *Data Acquisition* \rightarrow *Data Selection and Cleaning* \rightarrow *Data Transformation* \rightarrow *Data Mining/Pattern Discovery* \rightarrow finally leading to the *Interpretation* and *Evaluation* of the resulting model. This data-driven approach to modeling will enable designers to understand the temporal changes in the market space relating to product preferences and use this knowledge in the design of next-generation product families. The sequence of the KDD steps will now be expounded upon:

6.3.1.1 KDD Step 1: Data Acquisition

The data in the proposed methodology represents structured, time series data that exists within a company's database or acquired online through publicly available customer product preference websites using automated data acquisition techniques (Tucker and Kim 2011a). The two types of data employed in the proposed methodology are *structured* and *unstructured* data.

Structured data typically refers to data that can be conceptualized using an Entity-Relationship structure and easily stored in a Database Management System (Chen 1976). The Entity-Relationship Model is an example of the format of structured data where the *entity* (e.g., product domain) is *related* to certain features (e.g., product features).

Figure 6.2 is an example of time series structured data suitable for the proposed methodology, where each column (1, P) at time t_i is defined by a unique product feature (j). The last column containing the *Class* variable represents the dependent/output variable which is influenced by the levels/values of the product features. Examples of a class variable could be a market price segment ($> \$199$, $\$99$ – $\$199$, $\$0$ – $\$99$) or a purchasing decision (purchased, not purchased), etc. The proposed methodology assumes that the product features (j) can be categorical or numeric in nature, while the class variable is considered categorical for the subsequent data mining algorithm. In the proposed methodology, *structured data* will be used to

Fig. 6.3 Unstructured data of a cell phone product review

<p>Pros: Lightweight Vivrant 4.8 inch display 4G Connectivity</p> <p>Cons: May be oversized for some SD card does not come with the phone</p>
--

quantify the relevance of product features in the market space over time (Level 1: Temporal Market-Driven Preferences), which will then help guide Product Family decisions in Level 2 (Engineering Design Objective).

Unstructured data on the other hand refers to data that is not well suited for DBMS due to a lack of a well-formed entity-relation model (Buneman et al. 1996). Unstructured data primarily includes text data found in documents, web pages, numeric values, etc. An example of unstructured data would be a product review containing both textual and numeric information.

As can be seen in Fig. 6.3, the information contained in textual data does not have a well-defined feature/class relation found in Fig. 6.2, hereby making traditional data mining classification algorithms ill-suited for such data. However, unstructured data contains extremely valuable information regarding the domain of investigation and can be mined to quantify patterns using Natural Language Processing techniques that will be presented in the Data Mining step in the KDD process. In the proposed methodology, Natural Language Processing will be employed to understand the relation between product feature and component function in Level 2 (Engineering Design Objectives).

6.3.1.2 KDD Step 2: Data Selection and Cleaning

The second step in the KDD process aims to minimize noise in the data set that may arise due to missing data values, erroneous/ambiguous features, etc. Data selection and cleaning techniques should be employed for each data type used in the proposed methodology. For categorical features/class found in the *structured data* in Level 1, missing/erroneous values can be addressed by either replacing them with global constant values or the most probable values (based on the frequency of occurrence of a particular feature/class value) (Han et al. 2011). For *unstructured data*, used in Level 2 of the methodology, data selection and cleaning techniques may include text grammatical correction processing for nonword error detection, isolated-word error correction, and context-dependent word correction (Kukich 1992). For example, word corrections could be as straightforward as correcting “cel phne → cell phone” or more complex in trying to determine context and correct “real time *whether* updates → real time *weather* updates.”

6.3.1.3 KDD Step 3: Data Transformation

Step 3 of the KDD process is where the data is transformed into acceptable forms for the subsequent Data Mining/Pattern Discovery (Step 4) process. For *structured data* for example, *binning* techniques can help smooth data values of a feature by first sorting and placing feature values in predefined bin categories, where each bin category can be represented by the mean of the feature values in the specific bin (Han et al. 2011). For *unstructured data*, data transformation techniques may include *Stemming*; a process that aims to reduce morphological variants of words to their root form so that word variants can be mapped together (Paice 1994). For example, the words *charger* and *charging* both have the same root word *charge*, which could refer to a phone charger in product design. Data transformation techniques will reduce the noise caused by redundant feature values or words in a large data set.

6.3.1.4 KDD Step 4: Data Mining/Pattern Discovery

The Data Mining/Pattern Discovery step in the KDD process is where statistical/machine learning algorithms are employed to the transformed data (from Step 3) in order to discover novel, previously unknown knowledge about the domain of interest. The methodology begins with Phase 1, the iterative evaluation of the *relevance* of product features to the final class variable. Phase 2 presents the classification of the product features deemed *irrelevant* by the data mining predictive model that are then classified as *Standard*, *Nonstandard*, or *Obsolete*. The subsequent product family optimization step is guided by the predictive data mining results that are based on the temporal market-driven product preference data.

Phase 1: Iterative Feature Evaluation

The feature classification metric is modeled based on a time series decision tree induction algorithm that captures the emerging product feature trends over time (Tucker and Kim 2011b). *Phase 1* in Fig. 6.4 sequentially tests each product feature's entropy (at each iteration of the algorithm) using n time-stamped data sets. The calculation of the entropy values are used to rank each product feature's *relevance* to the class variable and also used as the test statistic to classify irrelevant product features in phase 2 of the methodology. In this work, the term *relevance* is defined as a product feature's relationship to the class/output variable.

Given n time intervals, t_1 to t_n , each time interval t_i contains a training data set T . For training data set T at time t_i , each of the feature is tested in order to determine that feature's ability to reduce the uncertainty of the class variable (please see Fig. 6.2). There are several metrics proposed in the literature for evaluating a feature's relation to a class variable including the Gini Index, Gain Ratio,

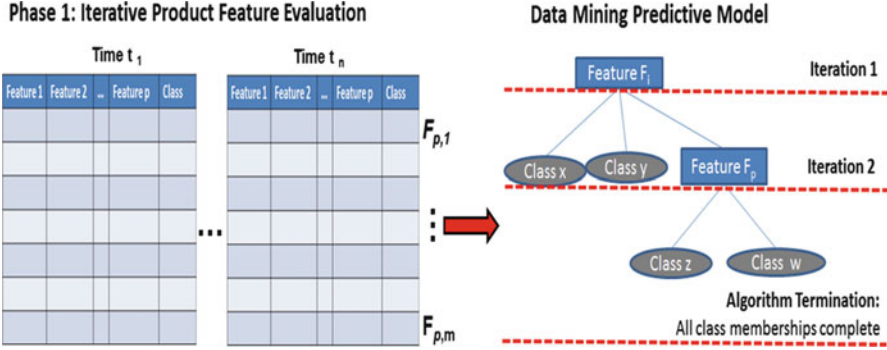


Fig. 6.4 Data Mining model generation based on time series product feature data

Likelihood-Ratio Chi-Squared Statistics, DKM Criterion, Twoing Criterion, etc. (Maimon and Rokach 2005). The methodology proposed in this work employs the *Gain Ratio* metric, although the algorithm is not limited to this metric.

The *Gain Ratio* is a well-established feature evaluation metric for determining the *best split* of the data set at each iteration. The assumption is that both the class variable and product features have values that are mutually exclusive of one another. Also, it is assumed that the variables are categorical or if continuous, can be discretized using existing statistical discretization techniques (Dougherty et al. 1995). The goal of the feature classification algorithm is to iteratively test each product feature for its ability to reduce the uncertainty/randomness of the class variable, generate a decision tree model, and then classify the features that do not show up in the resulting decision tree model as *Standard*, *Nonstandard*, or *Obsolete*.

Given a training data T set at time t_i , each with n features (continuous or discrete) and a class variable c_i , the *Gain Ratio* is defined as (Quinlan 1992):

$$Gain\ Ratio(X) = \frac{Entropy(T) - Entropy_X(T)}{Split(T)} \tag{6.1}$$

where:

$$Entropy(T) = - \sum_{i=1}^q p(c_i) * \log_2 p(c_i) \tag{6.2}$$

$p(c_i)$ represents the probability (relative frequency) of a class variable c_i in the training data set T .

q : represents the number of mutually exclusive class values within the data set.

$$Entropy_X(T) = - \sum_{j=1}^j \frac{T_j}{T} * Entropy(T_j) \tag{6.3}$$

T_j : represents a subset of the training data T that contains one of the mutually exclusive outcomes of a product feature. For example, if product feature X is

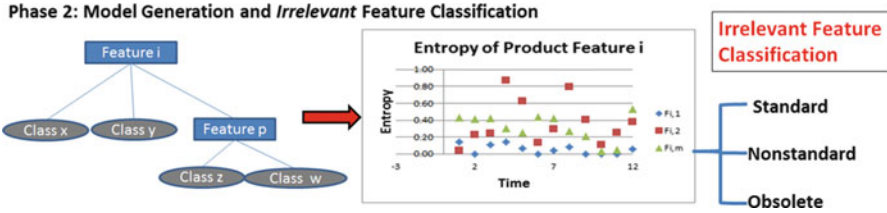


Fig. 6.5 Phase 2: Model generation and irrelevant feature classification

wireless connectivity containing 3 mutually exclusive outcomes (*WiFi, Bluetooth, NFC*), then T_j represents all the instances in T that contain one of those outcomes.

J : represents the number of mutually exclusive outcomes for a given feature.

The denominator of the *Gain Ratio* metric, $Split(T)$ normalizes the numerator, hereby reducing the bias of the metric towards features with a large number of mutually exclusive outcomes (T_j).

$$Split(T) = - \sum_{j=1}^j \frac{T_j}{T} * \log_2 \frac{T_j}{T} \tag{6.4}$$

From time periods t_1 to t_n , the *Gain Ratio* values for each product feature are computed and stored. A time series predictive model is then use to predict which product feature will have the maximum *Gain Ratio* values at future time periods t_{n+k} , where k represents the length of time before the next generation of products are to be launched (Tucker and Kim 2011b). Therefore feature F_i , appearing at the top of the Data Mining Predictive model in Fig. 6.4, represents the product feature with the highest predicted *Gain Ratio*, given a history of stored *Gain Ratio* statistics from time periods t_1 to t_n . Product feature F_p at iteration 2 represents the product feature with the highest predicted *Gain Ratio*, given a history stored *Gain Ratio* statistics from time periods t_1 to t_n . The algorithm continues to partition the original data time series data sets until a homogeneous class distribution exists for each leaf of the Data Mining Predictive Model, as seen in Fig. 6.4. The resulting model is represented as a decision tree, which can be read as a sequence of decision rules by traversing down each unique path of the tree. The resulting model will help design teams determine the specific product feature combinations that yield a particular outcome in the market space (e.g., price).

Phase 2: Model Generation and Irrelevant Feature Classification

Once the Data Mining Predictive Model has been generated from Phase 1, Phase 2 of the methodology (Fig. 6.5) introduces a technique to classify *irrelevant* product features based on the evolution of their importance to future product launches. A challenge in traditional engineering decision support models has been the understanding of the relationship between product features with low model relevance and

the effects on product family design decisions. Phase 2 in Fig. 6.5 overcomes these challenges by utilizing the time history entropy values (calculated and stored at each iteration in Phase 1) to determine the best course of action for *irrelevant* product features. Product feature *irrelevance* is defined simply as product features that do not show up in the resulting predictive model in Fig. 6.5. These product features are classified as either a *Standard Feature*, *Nonstandard Feature*, or an *Obsolete Feature*, with the pseudocode for the algorithm provide below (Tucker and Kim 2011b).

Start: Iteration $j = 1$

1. *If predicted Gain Ratio of Feature F_i is not the highest, Feature F_i is considered irrelevant*
2. *Employ Mann–Kendall (MK) trend test for Feature F_i*
 - a. *If MK τ is negative (with p -value $<$ alpha), irrelevant classification = Standard*
 - b. *Else If MK τ is positive (with p -value $<$ alpha), irrelevant classification = Obsolete*
 - c. *Else If MK τ is positive/negative (with p -value $>$ alpha), irrelevant classification = Nonstandard*
3. *While data set/subset does not contain a homogeneous class*
 - a. *Split the data set into subsets based on the number of mutually exclusive values of the feature with the highest Gain Ratio from Step 2*
 - b. *$j = j + 1$ and revert to Step 2 for each data subset*
4. *End Tree, Classify Irrelevant Feature F_i based on highest variable value (SF $_{t=1, \dots, n}$; NF $_{t=1, \dots, n}$; OF $_{t=1, \dots, n}$)*

In order to classify product features, the emerging predictive power of each product feature must be quantified over time (i.e., each product feature's relevance to the class variable over time as seen in Fig. 6.5). This is achieved by employing the nonparametric Mann–Kendall trend test, mathematically represented as (Kendall and Gibbons 1990):

$$\tau = \frac{S}{\frac{1}{2}n(n-1)} \quad (6.5)$$

where

$$S = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \text{sgn}(x_j - x_i) \quad (6.6)$$

n : represents the total number of time series data points

x_j : represents the data point one time step ahead

x_i : represents the current data point

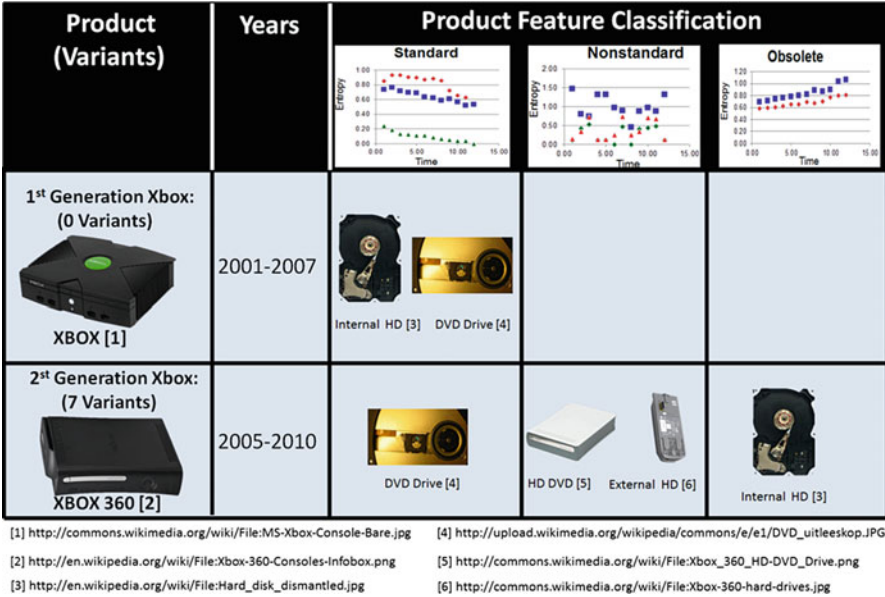


Fig. 6.6 Examples of product feature classification in consumer electronics

$$sgn = \begin{cases} 1 & \text{if } (x_j - x_i) > 0 \\ 0 & \text{if } (x_j - x_i) = 0 \\ -1 & \text{if } (x_j - x_i) < 0 \end{cases} \quad (6.7)$$

The Mann–Kendall begins with a null hypothesis of no trend and rejects or does not reject the null hypothesis based on the resulting p-value and level of significance (α).

The product feature classification framework relies on the results of the Mann–Kendal trend test to quantify the magnitude of the relationship between a given product feature and the output (class) variable. The 3 product feature classification categories are provided below with an application example in Fig. 6.6, illustrating the how the product feature classification could be used to guide enterprise level product family design decisions.

Standard Feature (SF)

A feature F_s is defined as standard if it does not show up in the final decision tree model (as seen in Fig. 6.5) and subsequent tests of the times series Entropy statistics using data from t_1, \dots, t_n (acquired at each iteration of the model generation process) reveal a monotonically decreasing trend. The Mann–Kendall trend detection test is used as the statistical measure to detect trends. If a monotonically decreasing trend is detected by the Mann–Kendall trend test, this means that despite Feature F_s 's absence from the decision tree model in Fig. 6.5, it is consistently gaining relevance over time and should therefore be considered as a candidate to be included in the

product platform decision in Level 2 of the methodology (Engineering Design Optimization level). The Mann–Kendall would return a negative τ and a p-value below the significance level (α).

A binary variable (SF) is defined for the Standard Feature classification that represents the results of the Mann–Kendall trend test at each iteration j . That is, if the Mann–Kendall trend test determines that product feature F_s has a monotonically decreasing entropy trend at iteration j , the binary variable (SF) assumes a value of 1, otherwise 0. Each iteration of the Standard Feature classification SF_j is weighted based on the number of supporting instances in the data set (T_j/T)

$$SF(t = 1, \dots, n) = \sum_{j=1}^n SF_j \cdot \left(\frac{T_j}{T}\right) \quad (6.8)$$

A product feature with a *Standard* classification could be considered for the product platform integration during the product family design process. The engineering components providing the functionality for this product feature could be shared across multiple products within the product family. For example, Fig. 6.6 shows 2 product features that were integrated into the 1st-generation Xbox platform: *Internal Hard Drive* and a *DVD player*. Other video game manufacturers such as Sega opted not to include DVD player functionality as a standard product feature in their video game platform (Sega Dreamcast) which contributed to the failure of the system, and ultimately the company as a whole (Aoyama and Izushi 2003). Understanding when to make product features standard (part of the product platform) or nonstandard (modular design that can be replaced/removed) is extremely critical to market success as will be seen in the following classification definitions.

Nonstandard Feature (NF)

A feature F_n is defined as Nonstandard if it does not show up in the final decision tree model (as seen in Fig. 6.5) and subsequent tests of the times series Entropy statistics using data from t_1, \dots, t_n (acquired at each iteration of the model generation process) reveal no discernible trend pattern. The Mann–Kendall trend detection test is used as the statistical measure to detect trends. If no discernible trend is detected by the Mann–Kendall trend test, this means that despite Feature F_n 's absence from the decision tree model in Fig. 6.5, Feature F_n exhibits inconsistent relevance patterns through time and should therefore be investigated during the detailed Engineering Design process (Level 2 of the methodology). The Mann–Kendall trend test would return a p-value *above* the significance level (α) (which would mean that we do not reject the null hypothesis of no trend). A binary variable (NF) is defined for the Nonstandard Feature classification that represents the results of the Mann–Kendall trend test at each iteration j . That is, if the Mann–Kendall trend test determines that product feature F_i has no discernible entropy trend at iteration j , the binary variable (NF) assumes a value of 1, otherwise 0. Each iteration of the Nonstandard Feature classification NF_j is weighted based on the number of supporting instances in the data set (T_j/T).

$$NF(t = 1, \dots, n) = \sum_{j=1}^j NF_j \cdot \left(\frac{T_j}{T}\right) \quad (6.9)$$

As opposed to having product variants share the same component addressing a given *Nonstandard* product feature, designers should avoid component sharing decisions within a product family, and instead develop unique components for each product variant in the product family. The engineering components providing the functionality for this product feature could therefore subsequently be replaced, upgraded, or removed altogether if the product feature eventually becomes obsolete in the market space. Figure 6.6 shows 2 product features of the 2nd-generation Xbox (Xbox 360) that had a modular design that was not shared between product variants within a product family: *HD-DVD player* and *Removable External Hard Drive*. During the video game console wars in the mid-2000s, two competing media formats were in direct competition with one another: the *Blu-ray* and *HD-DVD* (Brookey 2007). With the uncertainty of a clear winner in the market space, Microsoft opted for a modular add-on HD-DVD device (see Fig. 6.6) that could seamlessly integrate with the Xbox 360 product variants in the market space if High Definition media consumption was desired by consumers. The add-on HD-DVD was discontinued soon after it was clear that Sony's Blu-ray format had won the next-generation media platform wars (Daidj et al. 2010), making the HD-DVD modular device for the Xbox 360, *Obsolete*. If Microsoft had made the decision early on in the Xbox 360 product design process to integrate the HD-DVD player into the product platform, shared across multiple product variants, an entire redesign of the Xbox 360 system may have resulted after the HD-DVD product feature failed in the market space. The *Removable External Hard Drive* also allowed different variants of the Xbox 360 to have different storage capacities (20 GB, 60 GB, 120 GB, etc.), allowing greater customization in the market space, while keeping the core Xbox 360 relatively unchanged (Microsoft Inc. 2007; Farkas 2009).

Obsolete Feature (OF)

A feature F_o is defined as obsolete if it does not show up in the final decision tree model (as seen in Fig. 6.5) and subsequent tests of the times series Entropy statistics using data from t_1, \dots, t_n (acquired at each iteration of the model generation process) reveal a monotonically increasing trend. The Mann–Kendall trend detection test is used as the statistical measure to detect trends. If a monotonically increasing trend is revealed by the Mann–Kendall trend test, this means that despite Feature F_o 's absence from the decision tree model in Fig. 6.5, it is consistently losing relevance over time and should therefore be investigated during the detailed Engineering Design process in Step 3. The Mann–Kendall trend test would return a positive τ and a p-value below the significance level (α). A binary variable (OF) is defined for the Obsolete Feature classification that represents the value results of the Mann–Kendall trend test at each iteration j . That is, if the Mann–Kendall trend test determines that product feature F_o has a monotonically increasing entropy trend

at iteration j , the binary variable (OF) assumes a value of 1, otherwise 0. Each iteration of the Obsolete Feature classification OF_j is weighted based on the number of supporting instances in the data set (T_j/T).

$$OF(t = 1, \dots, n) = \sum_{j=1}^j OF_j \cdot \left(\frac{T_j}{T}\right) \quad (6.10)$$

The final classification of a feature (that does not show up in the decision tree model in Phase 2 of Fig. 6.5) is achieved by summing across all iterations of each of the feature classification variables (Standard, Nonstandard, and Obsolete) and selecting the variable with the highest value.

A product feature with an *Obsolete* classification indicates that it has little market-driven significance over time. The engineering components providing the functionality for this product feature can be considered candidate components to be removed in next-generation product designs. Figure 6.6 shows 1 product feature that was initially part of the 1st-generation Xbox platform, characterized as *Obsolete* in the 2nd-generation Xbox platform. The 2nd-generation Xbox platform (Xbox 360) launched in 2005 without an internal hard drive, making this product feature obsolete to the Xbox 360 platform. Microsoft opted for modular hard drives (External HD in Fig. 6.6) that could be replaced or upgraded at a certain price (Farkas 2009). This enabled Microsoft to discontinue an obsolete component (the internal hard drive), while creating a product family of Xbox 360 s that served different consumer market segments based on the technical capabilities of the Xbox 360 platform [platforms came with no External Hard Drive, 20 GB External Hard Drive and 100 GB Hard Drive (Farkas 2009)].

The methodology proposed in this chapter aims to address the link between the evolution of product feature relevance and the implications to product platform and product family design. Specifically, this work aims to:

- Translate a product feature classification from the market-driven domain to the detailed engineering domain.
- Determine the optimal product platform sharing decisions based on the market-driven evolution of product features and customer preferences.

6.3.2 Level 2: Engineering Design Optimization

6.3.2.1 Mapping Product Feature Space to Engineering Design Space

From the resulting Data Mining Predictive Model from Level 1 of the methodology, product features will either:

1. Be part of the predictive model and therefore considered relevant product features.

2. Be omitted from the predictive model and be classified as:

- (a) Standard
- (b) Nonstandard
- (c) Obsolete

Level 2 of the product feature classification framework plays a vital role in mapping market-driven, product feature preference trends to engineering design specifications. The ability of a product family to address market-driven demand is highly dependent on the evolution of product feature preferences over time. While component commonality decisions in product family design aim to provide the optimal configuration of product platforms within a product portfolio, mathematical models often omit evolving product feature preferences in the market space, hereby increasing the risk of product failure when launched. Unlike individually designed products, the market failure of a product family (e.g., due to an unwanted product feature) could result in the redesign of an entire product family (as opposed to just a single product) due to the components shared between product variants. In many real life scenarios, the characteristics of the product feature space (as defined by the customer) significantly differs from the technical characteristics of the engineering product design space. For example, a large-scale data set containing product preference data may contain a product feature such as 8 h *battery life*. For the same product feature however, the technical components used to achieve such functionality of a product's feature would be described by more technical terms such as *60 Whr 6-Cell Lithium-Ion Battery* (Dell Inc. 2012).

The aim is to first map the *Standard*, *Nonstandard*, and *Obsolete* product feature classifications to the *Functions* of a component(s) in a product family, as shown in Fig. 6.7. The *Standard* classification includes all features included in the Data Mining Model, in addition to the SF *irrelevant* feature classifications. The *Non-standard* classification includes all features classified as NF, and the *Obsolete* classification includes all features classified as OF. It is important to note that a feature can have one and only one classification.

The textual descriptions of each product feature will then be compared with the textual description of all functions in a product family to determine which components functions are providing the market-driven product preferences. Each component function F_i existing in a product family is assumed to be defined by a synonym set $\{s_1, s_2, \dots, s_s\}$ that describes its technical purpose to the product/product family as a whole. A product feature-function matrix is then created as Table 6.1.

Latent Semantic Analysis is employed to make semantic comparison between the vector of terms characterizing a product feature and those of a product function. LSA not only compares the original vector of textual terms but also their *semantic* meaning and makes the assumption that terms with similar meanings will occur close to each other. Therefore, despite the fact that a product feature term “charge” and the product function “battery” are not identically the same, LSA may quantify the related meaning between the two.

Table 6.1 can be represented by one of two vectors

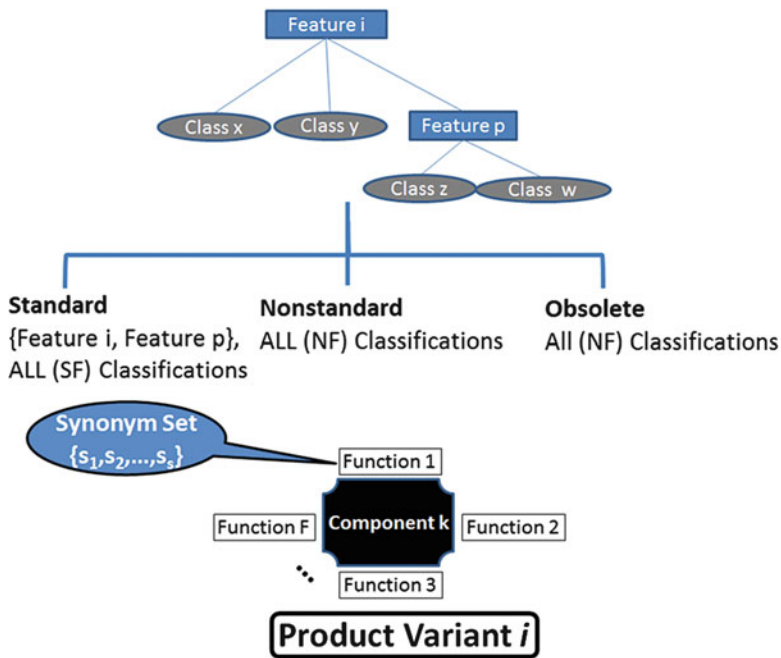


Fig. 6.7 Component function identification

Table 6.1 Product feature-function comparison matrix

	Product feature P	Product function 1	...	Product function F
Term 1	$C_{1,1}$	$C_{1,2}$...	$C_{1,n}$
Term 2	$C_{2,1}$	$C_{2,2}$...	$C_{2,n}$
⋮	⋮	⋮	...	⋮
⋮	⋮	⋮	...	⋮
⋮	⋮	⋮	...	⋮
Term T	$C_{m,1}$	$C_{m,2}$...	$C_{m,n}$

- Semantic term vector (each row of Table 6.1):

$$t_i^T = [c_{i,1} \dots c_{i,n}] \tag{6.11}$$

- Product feature-function Comparison (each column of Table 6.1):

$$f_j = \begin{bmatrix} C_{1,j} \\ \cdot \\ \cdot \\ \cdot \\ C_{m,j} \end{bmatrix} \tag{6.12}$$

Table 6.1 can be defined as X , where $c_{i,j}$ represents the frequency/occurrence of a particular term in the description of either a product feature or product function.

$$X = \begin{bmatrix} c_{1,1} & \cdots & c_{1,n} \\ \vdots & \ddots & \vdots \\ c_{m,1} & \cdots & c_{m,n} \end{bmatrix} \quad (6.13)$$

The singular value decomposition (SVD) of X can therefore be represented as (Deerwester et al. 1990):

$$X = T_0 S_0 D'_0 \quad (6.14)$$

where

X : is the term (t) by function/feature (f) matrix (i.e., $X = t \times f$)

T_0 : represents the term (t) by rank (m) matrix, having orthogonal, unit-length columns ($T_0' T_0 = I$)

S_0 : is the diagonal matrix of singular values ($m \times m$)

D_0 : is the rank (m) by function (f) matrix, having orthogonal, unit-length columns ($D_0' D_0 = I$) (i.e., $D_0 = m \times f$)

m : is the rank of X ($\leq \min(t, d)$)

LSA therefore provides lower-dimension estimates of the original high-dimension space which then enables a comparison of the semantic meaning (beyond just simple term matching) between a product feature and a product's function using similarity metrics such as *cosine similarity* (Tucker and Kang 2012).

Once the market-driven product features have been mapped to specific product functions using SVD, a detailed function-component analysis must be performed. A component-function matrix representation is used to quantify the relationships/interactions between components. A majority of the literature reviewed in this work typically focus on component sharing optimization within and between a family of products. By employing the market-driven product feature classification methodology presented in the previous section, engineering design decisions relating to product family optimization can be guided by emerging product preferences in the market space. The DSM has been employed extensively in the design community to represent interactions among products/processes in a design process (Browning 2001). Examples of interactions captured by the DSM framework include *Spatial* (associations relating to the physical location of elements), *Energy* (energy transfer/exchange between elements), *Information* (data/signal exchanges between elements), and *Material* (material exchange between elements) (Pimpler and Eppinger 1994).

The first step is to determine which functions of a product relate to specific component(s). There may be some components that perform more than one specific function, even across product variants as can be seen in Fig. 6.8. For example, unlike the *60 Whr 6-Cell Lithium-Ion Battery* that supplies electrical energy to a

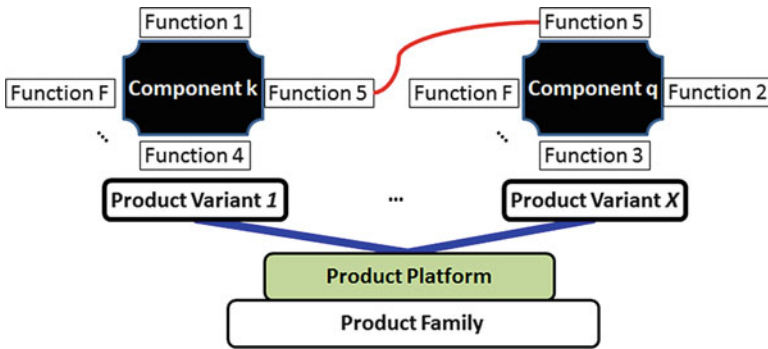


Fig. 6.8 Function-component analysis in a product family

Table 6.2 Component-function interaction matrix

	Function 1	Function 2	...	Function N
Component 1	1			
Component 2				1
...		0		
Component m	0			

product as its *function*, other components such as a DVD super drive component provides multiple functions such as reading media, recording media, and erasing media. Each of these would be considered a unique function of this component. Atomicity is a desired property of the functional decomposition process. It is assumed that designers have a database of components with their individual functions; therefore, the component-function relationship in Table 6.2 will ensure that design teams understand the interactions among components within a product variant and across product variants existing in a product family.

where

1: represents a component *critical* to achieving a specific function

0: represents a component that is *complimentary* to achieving a specific function

The product family optimization problem can be solved using a *quasi-separable*, bi-level optimization model, where the coordination level handles the shared variables (components common to the system) and the product platform level handles the individual product variant optimization problems (each with local objective functions such as cost minimization) (Kim et al. 2003; Tosserams et al. 2006). three binary feature classification variables are included in the objective function to help guide the optimization problem. The *Standard Feature (SF)*, *Nonstandard Feature (NF)*, and *Obsolete Feature (OF)* are modeled as follows.

Optimization Level 1: Product Family Sharing Level

Minimize

$$\boldsymbol{\varepsilon}_y \quad (6.15)$$

Subject to:

$$g1 : SF_p \cdot \sum_{k \in Q} \left\| \mathbf{y}_p - \mathbf{y}_{p,k}^{Eng} \right\|_2^2 - \boldsymbol{\varepsilon}_y \leq 0 \quad (6.16)$$

Here,

p : product feature from the market-driven Data Mining Predictive Model.

SF_p : binary variable for *Standard Feature* classification (1 if the product feature is deemed *relevant* by the Data Mining Predictive Model and 0 otherwise).

\mathbf{y}_p : linking variable at the product family sharing level that maintains consistency between k product variant values. The component or variable \mathbf{y}_p corresponds to function-component map for the specific product feature p .

$\mathbf{y}_{p,k}^{Eng}$: the shared variable/component value associated with product variant k providing product feature p . This is constant at each iteration in the above formulation that is subsequently updated at the product variant optimization level after each iteration.

k : the k th candidate product variant that has been identified for component sharing.

Q : the total number of products attempting to share design variables/components $\mathbf{y}_{p,k}^{Eng}$.

$\boldsymbol{\varepsilon}_y$: deviation tolerance between linking variables that is minimized in the objective function.

Optimization Level 2: Product Variant Optimization Level

Minimize

$$F(x)_{Variant_k} = f_k + \left\| \mathbf{y}_p^U - \mathbf{y}_{p,k} \right\| \quad (6.17)$$

Subject to:

$$\mathbf{g}_k(\mathbf{x}_k, \mathbf{y}_{p,k}) \leq \mathbf{0} \quad (6.18)$$

$$\mathbf{h}_k(\mathbf{x}_k, \mathbf{y}_{p,k}) = \mathbf{0} \quad (6.19)$$

Here,

p : product feature from the market-driven Data Mining Predictive Model.

NS_p : binary variable for *Nonstandard Feature* classification (1 if the product feature is classified as *Nonstandard Feature* by the Data Mining Predictive Model and 0 otherwise).

OF_p : binary variable for *Obsolete Feature* classification (1 if the product feature is classified as an *Obsolete Feature* by the Data Mining Predictive Model and 0 otherwise).

f_k : local product design objective function (s).

g_k : inequality design constraints.

h_k : equality design constraints.

x_k : design variables local to product variant k . x_k is a function of the product feature variables NS_p and OF_p . NS_p and OF_p are presented here in a general form as the mathematical formulation and inclusion in the optimization model will be highly dependent on the structure of the product family model.

y_p^U : linking variable target value cascaded down to the *Level 2* from *Level 1*; a constant value at each iteration that is subsequently updated with each successful iteration.

$y_{p,k}$: linking variable at *Level 2* that attempts to match the target linking variable value y_p^U used to achieve the product feature p .

k : the k th candidate product variant that has been identified for component sharing.

For each unique product feature that is included in the product family optimization model has to satisfy the equality constraint: $H1: SF_p + NS_p + OF_p = 1$, indicating a single state during the product family optimization model (variable/component sharing, module, or exclusion from the optimization model).

6.4 Case Study of a Family of Aerodynamic Particle Separators

Particulate Matter (PM)/particle pollution is a complex mixture of very small particles such as acids, organic chemicals, metals, soil, or dust particles (US EPA 2012). Severe health problems can be caused to the heart, lungs, and other organs when PM sizes are 10 μm in diameter or smaller.

Aerodynamic particle separators are devices developed to separate Particulate Matter from the clean air stream, typically by employing centrifugal forces on the particles (Zhang 2005). The case study presented in this methodology is based on an aerodynamic particle separator market segment including applications such as agriculture, industrial, and manufacturing processes. The global market for air cleaning technologies has exceeded \$7 Billion and continues to rise (Parker 2006). The diverse operating conditions and preferences of customers make product standardization a challenge. Customized solutions for aerodynamic particle separators are typically used to solve the wide range of market segments (Fig. 6.9).

This case study aims to investigate the feasibility of employing the proposed Data Mining-Driven product family design methodology to help:

- Quantify the evolution of product feature characteristics over time.
- Develop a Data Mining predictive model of *relevant* product features for future product family designs.



Fig. 6.9 Aerodynamic particle separator market segments [adapted from (Barker 2008)]

Table 6.3 Sample data set for aerodynamic particle separator

Product features					Environmental features				Class variable
Q	Delta p _{max}	L _{max}	AF _{max}	N _{max}	F(dp)	Rho _p	T _{air}	P _{air}	Efficiency
3	2259	0.16	1.2	27	MMD ₃₅ - GSD _{1.9}	2181	142	138	85–90 %
1	923	0.02	0.76	27	A1	2977	218	94	85–90 %
1	753	0.53	0.01	37	A4	2022	279	118	85–90 %
3	2082	0.69	0.13	10	Limestone	2133	50	130	85–90 %
4	2890	0.64	1.01	44	MMD ₁₀ - GSD _{1.8}	2714	473	199	85–90 %

- Classify product features as *Standard*, *Nonstandard*, or *Obsolete*.
- Investigate how product feature classification influences product family sharing and optimization decisions.

For more details regarding this case study, please see references Barker (2008) and Tucker et al. (2010).

6.4.1 Level 1: Temporal Market-Driven Preferences

Table 6.3 above presents a snapshot of the structure of the data set for the aerodynamic particle separator for one instant in time (t_i), where:

Q : air flow rate (m^3/s)

ΔP_{max} : maximum allowable change in pressure drop/airflow restriction (Pa)

L_{max} : total allowable length of the system (m)

AF_{max} : maximum allowable face area perpendicular to air flow direction (m^2)

N_{max} : maximum number of aerodynamic particle separator units in one module (#)

$F(d_p)$: particle size distribution (%)

ρ_p : particle density (kg/m^3)

T_{air} : air temperature (°C)

P_{air} : air pressure (kPa)

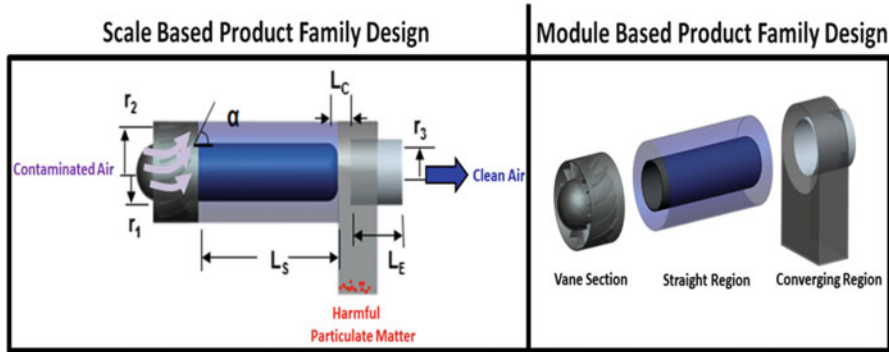


Fig. 6.10 Uniflow aerodynamic particle separator design [Augmented from Tucker et al. (2010)]

The data set contains nine features relating to the aerodynamic particle separator (five of which are related to the physical design of the system while the remaining four are related to the environmental conditions that the system will perform under). The product features will help guide the product family optimization process by suggesting candidate components for sharing or displacement. The Environmental Features will serve as the design constraints of the model. The *class* variable here is *Efficiency* which is defined as the total amount of particulate matter that a system is able to separate from clean air.

The data in Table 6.3 is mined for emerging product feature trends in the market space by quantifying the *relevance* of product features over time. A Data Mining predictive model is generated that helps guide Level 2 of the product family design methodology: Engineering Design Optimization.

6.4.2 Level 2: Engineering Design Optimization

The aerodynamic particle separator has a fan system downstream (attached to the radius r_3 in Fig. 6.10) and operates by pulling contaminated air from upstream into the system. The contaminated air (composed of clean air and particulate matter) enters the vane section (Fig. 6.10) causing the particles to rotate based on the angle of the vane section (Barker 2008). The straight section in Fig. 6.10 is designed to increase the separation (centripetal acceleration and inertia) between the particulate matter and the clean air. The clean air particles enter the *converging region* of the particle separator, leaving the particulate matter to collect in the storage bunker in Fig. 6.10.

Figure 6.10 presents 2 approaches to the product family design, *Scale based* and *Module based*. *Scale-based* product family is where a product platform is “stretched” or “shrunk” in one or more dimensions in order to satisfy a market need (Simpson et al. 2006). Design/scaling variables are formulated in the product

family optimization model to achieve such scalability. *Module-based* product family design on the other hand creates product variants by adding, replacing, or substituting one or more functional modules from a product platform. A product architecture is considered *modular* if there is a clearly defined mapping of functional elements to physical structures (1-1 or many-1) (Simpson et al. 2006).

The design objectives of each aerodynamic product variant will be influenced by the market-driven Data Mining model. For scale-based product family design, the efficiency of the system can be influenced by altering (*stretching* or *shrinking*) the design variables that make up the physical system (such as length of straight region, inner and outer radii, etc.)

The optimization approach (module vs. scale based) is left up to the design team based on the technical resources and available mathematical models.

Optimization Level 1: Product Family Sharing Level

The *Product Family Sharing Level* will coordinate the component sharing among product variants of the aerodynamic particle separator by minimizing the tolerance deviation variable of each shared component.

Minimize

$$\epsilon_y \quad (6.20)$$

Subject to:

$$g1 : SF_p \cdot \sum_{k \in Q} \left\| \mathbf{y}_p - \mathbf{y}_{p,k}^{Eng} \right\|_2^2 - \epsilon_y \leq 0 \quad (6.21)$$

Here,

p : product feature from the market-driven Data Mining Predictive Model.

SF_p : binary variable for *Standard Feature* classification (1 if the product feature is deemed *relevant* by the Data Mining Predictive Model and 0 otherwise).

\mathbf{y}_p : linking variable at the product family sharing level that maintains consistency between k product variant values. The component or variable \mathbf{y}_p corresponds to function-component map for the specific product feature p .

$\mathbf{y}_{p,k}^{Eng}$: the shared variable/component value associated with product variant k providing product feature p .

k : the k th candidate aerodynamic particle separator product variant

Q : The total number of products attempting to share design variables/components

$\mathbf{y}_{p,k}^{Eng}$.

ϵ_y : deviation tolerance between linking variables that is minimized in the objective function.

Optimization Level 2: Aerodynamic Particle Separator Variants

The engineering design model for the aerodynamic particle separator can be mathematically represented as:

k th Aerodynamic Particle Separator

Minimize:

$$F(\mathbf{x})_{variant(k)} = Cost_k - \xi_k + \left\| \mathbf{y}_p^U - \mathbf{y}_{p,k} \right\| \quad (6.22)$$

where

$$\zeta(\mathbf{x}, d_{pi}) = 1 - \exp\left(\frac{\rho_p d_{pi}^2 C_c Q \tan(\alpha) L_s}{9\eta(r_2^2 - r_1^2)}\right) \cdot \exp\left(\frac{\rho_p d_{pi}^2 C_c (V_t^2 G_t(x) + V_z^2 G_r(x))}{\eta V_z}\right) \quad (6.23)$$

$$\xi_k = \sum_{i=1}^N \zeta(\mathbf{x}, d_{pi}) \cdot F(d_{pi}) \quad (6.24)$$

Here,

ξ_k : efficiency of aerodynamic particle separator variant k

\mathbf{y}_p^U : linking variable target value cascaded down to the *Level 2* from *Level 1*; a constant value at each iteration that is subsequently updated with each successful iteration

$\mathbf{y}_{p,k}$: linking variable at *Level 2* that attempts to match the target linking variable value \mathbf{y}_p^U used to achieve the product feature p

k : the k th candidate product variant that has been identified for component sharing

C_c : Cunningham slip correction factor

d_{pi} : diameter of particle i , μm

$F(d_p)$: particle size distribution

$G_t(x)$: efficiency model geometric relationship between design variables, tangential acceleration

$G_r(x)$: efficiency model geometric relationship between design variables, radial acceleration

ρ_p : particle density, kg/m^3

η : air viscosity, $\text{Pa}\cdot\text{s}$ or $\text{kg}\cdot\text{m}/\text{s}$

Q : air flow rate, m^3/s

V_t : tangential velocity of particle mixture

V_z : axial velocity of particle mixture

r_1 : inner tube radius

r_2 : inner tube radius

α : vane discharge angle

L_s : maximum pressure drop

Subject to:

$$\mathbf{g}_k(\mathbf{x}_k, \mathbf{y}_{p,k}) \leq \mathbf{0} \quad (6.25)$$

$$\mathbf{h}_k(\mathbf{x}_k, \mathbf{y}_{p,k}) = \mathbf{0} \quad (6.26)$$

6.5 Results and Discussion

6.5.1 Level 1: Temporal Market-Driven Preferences

Figure 6.11 presents the Data Mining Predictive Model based on the temporal market-driven preferences relating to the aerodynamic particle separator. The results in Fig. 6.11 can be interpreted by traversing down each individual branch in the tree until a class variable (Efficiency) is reached (rectangular box). The ovals in Fig. 6.11 represent the product feature that is deemed *relevant* to predicting the market preferences for aerodynamic particle separator efficiency.

Four unique paths can be attained based on the results in Fig. 6.11:

- $L_{max} > 0.49$ then efficiency $> 85\text{--}90\%$
- $L_{max} < 0.49$ and $Q > 3$ and $\Delta_{pmax} > 1560$ then efficiency $> 95\%$
- $L_{max} < 0.49$ and $Q > 3$ and $\Delta_{pmax} < 1560$ then efficiency $> 90\text{--}95\%$
- $L_{max} < 0.49$ and $Q \leq 3$ then efficiency $> 85\text{--}90\%$

Figure 6.11 also provides designers with the appropriate product feature classification (*Standard, Nonstandard, and Obsolete*) of all product features existing in the market space. The next step is to quantify the relationship between product features and product function so that designers can understand how evolving market-driven preferences guide next-generation product platform and product family design decisions.

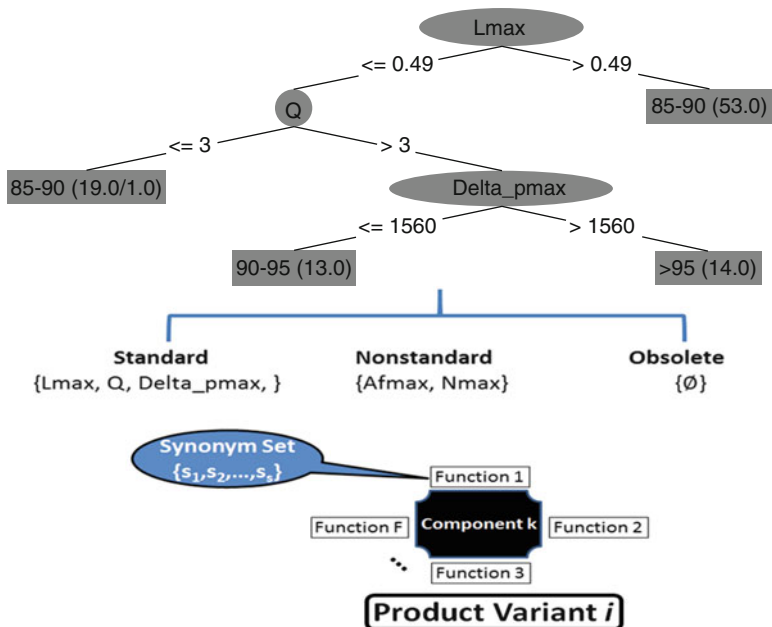


Fig. 6.11 Results from the data mining predictive model [attained using Weka 3.6.6 (Frank et al. 2010)]

Table 6.4 Mapping product features of the aerodynamic particle separator to the engineering design space

Product features	Aerodynamic particle separator design space		
	Vane section	Straight region	Converging region
Q	0.76	0.71	0.72
Δp_{max}	0.48	0.5	0.47
L_{max}	0.45	0.46	0.43
Af_{max}	0.86	0.82	0.78
N_{max}	0.49	0.55	0.52
$F(d_p)$	0.35	0.5	0.47
ρ_p	0.35	0.5	0.45
T_{air}	0.55	0.52	0.57
P_{air}	0.63	0.59	0.62

6.5.2 Level 2: Engineering Design Optimization

Mapping Product Feature Space to Engineering Design Space: Table 6.4 presents the result, employing Latent Semantic Analysis to quantify the relationship between the market-driven product feature space and the product family design space. As described in Sect. 6.2.2, the textual description of each product feature is compared with the functional description of each product module/component providing this function. Table 6.4 represents the *similarity* of a product feature to a component function as measured on a 0–1 scale, where values closer to 1 indicate a stronger relationship to the functionality of the product component/module, while 0 represents a weaker relationship. As can be seen from Table 6.4, the product features are strongly coupled across the entire product architecture. Such insight will help designers understand the market effects of adding, removing, or replacing specific functionality relating to a product. For an *Obsolete* product feature classification, designers would need to ensure that the removal of a particular component/module does not have negative market demand implications. The results in Table 6.4 are consistent with the absence of an *Obsolete* product feature classification from the Data Mining model in Fig. 6.11. The *Standard* and *Nonstandard* product classifications from Fig. 6.11 support the findings from Table 6.4, indicating that all product features are *relevant* to market success at this time. The challenge arises when designers are trying to optimize product family sharing and platforming decisions, which will now be guided by the product feature preference trends in the market space.

Figure 6.12 presents the results from the Data Mining-Driven Product Design methodology. The resulting product feature classifications from Fig. 6.11 helps guide the product family design process by first quantifying the functional relationships between product features and product design variables (Table 6.4) and then suggesting candidate modules/components for sharing decisions. In Fig. 6.12, the vane component is considered a candidate for commonality across product variants due to its relation to the product features deemed relevant by the

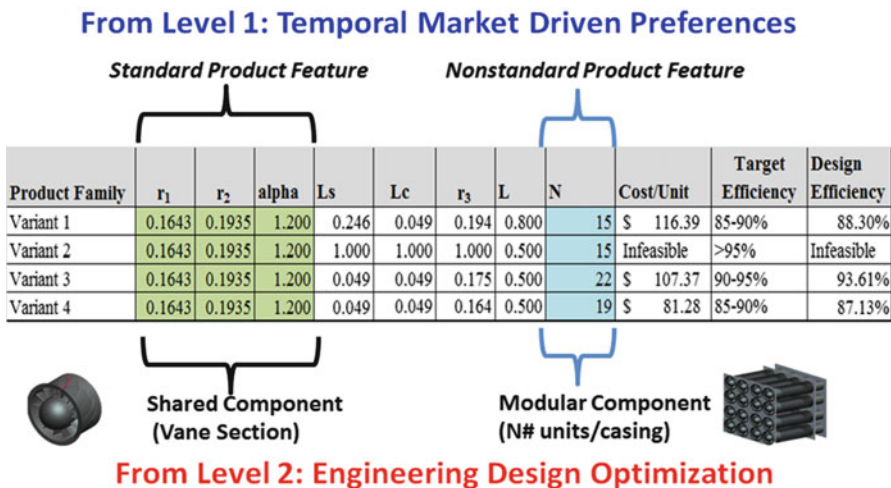


Fig. 6.12 Results from data mining-driven product family design

Data Mining Predictive model in Fig. 6.11. The *Nonstandard* product feature suggests modularity in the design of product variants by including multiple individual products that are housed in a design case. This modularity approach will enable enterprise decision makers to quickly address market needs by increasing/decreasing the number of units housed in a casing in an attempt to meet emerging customer preferences. The *Nonstandard* product feature classification means that there is volatility in the market space, wherein a product may have to undergo modifications (modular or scalable) in the future. Figure 6.12 reveals that of the 4 product segments suggested by the market space, the designers can only satisfy the performance requirements of 3 of those markets, as Product Variant 2's design (Target efficiency > 95 %) is infeasible at the engineering design level. Such design insights enable design teams to focus on developing a product portfolio that both capitalizes on product standardization through *guided* commonality decisions, and at the same time, providing customized solutions to the market that meet customer expectations (efficiency requirements).

6.6 Conclusions

This chapter introduces a market-driven, product family design framework based on product feature classification framework as it relates to engineering component selection and product family design. Product features are classified as *Standard*, *Nonstandard*, or *Obsolete*, with a given classification having different implications in the product family design process. A component-function matrix is presented to quantify the relationships and interactions among components as it relates to

product specific features. By employing Natural Language Processing techniques, the product feature space can be mapped to the engineering design space for optimal product platform decisions that incorporate market-driven objective. The methodology aims to aid design teams in the efficient modeling of customer preferences and designing of subsequent product portfolios.

References

- Agard B, Kusiak A (2004) Data-mining-based methodology for the design of product families. *Int J Prod Res* 42:2955–2969
- Alizon F, Shooter S, Simpson T (2009) Assessing and improving commonality and diversity within a product family. *Res Eng Des* 20:241–253. doi:[10.1007/s00163-009-0066-5](https://doi.org/10.1007/s00163-009-0066-5)
- Aoyama Y, Izushi H (2003) Hardware gimmick or cultural innovation? Technological, cultural, and social foundations of the Japanese video game industry. *Res Pol* 32:423–444. doi:[10.1016/S0048-7333\(02\)00016-1](https://doi.org/10.1016/S0048-7333(02)00016-1)
- Barker D (2008) Development of an optimization design platform for aerodynamic particle separators. MS, University of Illinois at Urbana-Champaign
- Boas RC (2008) Commonality in complex product families: implications of divergence and lifecycle offsets. Thesis, Massachusetts Institute of Technology
- Bouchereau V, Rowlands H (2000) Methods and techniques to help quality function deployment (QFD). *Benchmark Int J* 7:8–20. doi:[10.1108/14635770010314891](https://doi.org/10.1108/14635770010314891)
- Braha D (2001) Data mining for design and manufacturing. Kluwer Academic Publishers, The Netherlands, P.O. Box 17, 3300 AA Dordrecht
- Brookey RA (2007) The format wars. *Convergence* 13:199–211. doi:[10.1177/1354856507075245](https://doi.org/10.1177/1354856507075245)
- Browning TR (2001) Applying the design structure matrix to system decomposition and integration problems: a review and new directions. *IEEE Trans Eng Manag* 48:292–306. doi:[10.1109/17.946528](https://doi.org/10.1109/17.946528)
- Buneman P, Davidson S, Hillebrand G, Suciu D (1996) A query language and optimization techniques for unstructured data. *SIGMOD Rec* 25:505–516. doi: <http://doi.acm.org/10.1145/235968.233368>
- Chen PP (1976) The entity-relationship model – toward a unified view of data. *ACM Trans Database Syst* 1:9–36
- Collier D (1981) The measurement and operating benefits of component part commonality. *Decis Sci* 12(1):85–96
- Daidj N, Grazia C, Hammoudi A (2010) Introduction to the non-cooperative approach to coalition formation: the case of the Blu-Ray/HD-DVD standards’ war. *J Media Econ* 23:192–215. doi:[10.1080/08997764.2010.527206](https://doi.org/10.1080/08997764.2010.527206)
- Deerwester S, Dumais ST, Furnas GW et al (1990) Indexing by latent semantic analysis. *J Am Soc Inform Sci* 41:391–407
- Dell Inc. (2012) DELL 60 WHR 6-Cell Lithium-Ion Battery for Dell Inspiron 14z (1470)/15z (1570) Laptops: Laptop Accessories | Dell. <http://http://accessories.us.dell.com/sna/productdetail.aspx?c=us&l=en&s=bsd&cs=04&sku=312-0823>. Accessed 1 Feb 2012
- Dougherty J, Kohavi R, Sahami M (1995) Supervised and unsupervised discretization of continuous features. In: *Proceedings of the 12th international conference on machine learning (ICML-1995)*, San Francisco, CA
- Farkas BG (2009) *The Xbox 360 pocket guide*. Pearson Education, Upper Saddle River, NJ
- Frank E, Hall M, Holmes G et al (2010) Weka-a machine learning workbench for data mining. In: Maimon O, Rokach L (eds) *Data mining and knowledge discovery handbook*. Springer, Berlin, pp 1269–1277
- Han J, Kamber M, Pei J (2011) *Data mining: concepts and techniques*. Elsevier, Amsterdam

- Jiao J, Tseng MM (2000) Understanding product family for mass customization by developing commonality indices. *J Eng Des* 11:225–243
- Kendall M, Gibbons JD (1990) Rank correlation methods, 5th edn. A Charles Griffin Title, London
- Kim HM, Michelena NF, Papalambros PY (2003) Target cascading in optimal system design. *Trans ASME J Mech Des* 125:474–480
- Kota S, Sethuraman K, Miller R (2000) A metric for evaluating design commonality in product families. *ASME J Mech Des* 122(4):403–410
- Kukich K (1992) Techniques for automatically correcting words in text. *ACM Comput Surv* 24:377–439. doi:[10.1145/146370.146380](https://doi.org/10.1145/146370.146380)
- Kwong CK, Bai H (2003) Determining the importance weights for the customer requirements in QFD using a fuzzy AHP with an extent analysis approach. *IIE Trans* 35:619–626. doi:[10.1080/07408170304355](https://doi.org/10.1080/07408170304355)
- Martin M, Ishii K (1996) Design for variety: a methodology for understanding the costs of product proliferation. In: ASME design engineering technical conferences and computers in engineering conference – design theory and methodology, ASME, Irvine, CA, 18–22 Aug. Paper no. 96-DETC/DTM-1610
- Martin MV, Ishii K (2002) Design for variety: developing standardized and modularized product platform architectures. *Res Eng Des* 13(4):213–235
- Maimon O, Rokach L (2005) Data mining and knowledge discovery handbook, 1st edn. Springer, Berlin
- Martin MV, Ishii K (1997) Design for variety: development of complexity indices and design charts. *Advances in design automation*, ASME, Paper No. DETC97/DFM-4359
- Meyer MH, Lehnerd AP (1997) The power of product platforms. Free Press, Boston
- Microsoft Inc. (2007) Microsoft unveils xbox 360 elite. <http://www.microsoft.com/en-us/news/press/2007/mar07/03-27xbox360elitepr.aspx>. Accessed 21 Jul 2012
- Moon SK, Kumara SRT, Simpson TW (2006) Data mining and fuzzy clustering to support product family design. In: Proceedings of the ASME design automation conference
- Paice CD (1994) An evaluation method for stemming algorithms. In: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval. Springer, New York, NY, pp 42–50
- Parker PM (2006) The 2007–2012 world outlook for dust collection and other air purification equipment and parts for cleaning incoming air. ICON Group International, San Diego, CA
- Pimpler TU, Eppinger SD (1994) Integration analysis of product decompositions. <http://dspace.mit.edu/handle/1721.1/2514>. Accessed 29 Jan 2012
- Pullmana ME, Mooreb WL, Wardellb DG (2002) A comparison of quality function deployment and conjoint analysis in new product design. *J Prod Innov Manag* 19:354–364
- Quinlan JR (1992) C4.5: programs for machine learning, 1st edn. Morgan Kaufmann, San Francisco
- Siddique Z, Rosen DW, Wang N (1998) on the applicability of product variety design concepts to automotive platform commonality. In: Design theory and methodology, ASME, Atlanta, GA, Sept 1998. Pap. No DETC98DTM-5661
- Simpson T, Maier J, Mistree F (2001) Product platform design: method and application. *Res Eng Des* 13:2–22. doi:[10.1007/s001630100002](https://doi.org/10.1007/s001630100002)
- Simpson TW, Siddique Z, Jiao J (2006) Product platform and product family design: methods and applications. Birkhäuser, New York, NY
- Simpson T, Bobuk A, Slingerland L et al (2012) From user requirements to commonality specifications: an integrated approach to product family design. *Res Eng Des* 23:141–153. doi:[10.1007/s00163-011-0119-4](https://doi.org/10.1007/s00163-011-0119-4)
- Thevenot HJ, Simpson TW (2006) Commonality indices for product family design: a detailed comparison. *J Eng Des* 17(2):99–119
- Tosserams S, Etman LFP, Rooda JE (2006) An augmented Lagrangian decomposition method for quasi-separable problems in MDO. *Struct Multidisc Optim* 34(3):211–227

- Tucker C, Kang S (2012) A bisociative design framework for knowledge discovery across seemingly unrelated product domains. In: 2012 ASME international design engineering technical conferences
- Tucker CS, Kim HM (2008) Optimal product portfolio formulation by merging predictive data mining with multilevel optimization. *Trans ASME J Mech Des* 130:991–1000
- Tucker CS, Kim HM (2009) Data-driven decision tree classification for product portfolio design optimization. *J Comput Inform Sci Eng* 9:041004 doi: [10.1115/1.3243634](https://doi.org/10.1115/1.3243634)
- Tucker C, Kim H (2011a) Predicting emerging product design trend by mining publicly available customer review data. In: Proceedings of the 18th international conference on engineering design (ICED11), vol. 6. pp 43–52
- Tucker CS, Kim HM (2011b) Trend mining for predictive product design. *ASME J Mech Des* 133
- Tucker CS, Kim HM, Barker DE, Zhang Y (2010) A ReliefF attribute weighting and X-means clustering methodology for top-down product family optimization. *Eng Optim* 42:593–616
- US EPA O (2012) Particulate matter | air & radiation | US EPA. <http://www.epa.gov/pm/>. Accessed 22 Jul 2012
- Wacker JG, Trelevan M (1986) Component part standardization: an analysis of commonality sources and indices. *J Oper Manag* 6(2):219–244
- Wang J (2008) Encyclopedia of data warehousing and mining, 2nd edn. IGI Global, Hershey, PA
- Weka (2012) Weka 3 – data mining with open source machine learning software in java. <http://www.cs.waikato.ac.nz/ml/weka/>. Accessed 24 Jul 2012
- Zhang Y (2005) Indoor air quality engineering. CRC, Boca Raton, FL