INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 05 MELBOURNE, AUGUST 15-18, 2005

ROBUST PRODUCT CONCEPT GENERATION

Yoram Reich   Amir Ziv Av

Keywords: optimal concept, product platform, product family, standard, modular

1 Introduction

It is accustomed to call robust design a design that is resilient to noise. A product could be designed to be robust by methods such as Taguchi's (1987). The idea is to manipulate the design parameters that could be controlled by designers to minimize the effect of the noise on the planned behavior in the designated environment. We are concerned with a broader perspective of robustness, one that arises from many environmental uncertainties including those related to technical knowledge, customers, and market conditions. In such interpretations, the product behavior includes physical behavior as well as customer satisfaction, cost, as well as any parameter that is related to the technical and market success of the product. In this context, we define a product as robust if a large variety of potential environmental uncertainties have little impact on its behavior.

While the broad perspective of robustness applies to all design stages, we are in particular interested in the conceptual design stage that is considered to be the most critical step in product development. In this stage, an abstract description of the product is created that serves as the basis for subsequent design stages and decisions. To a large extent, the quality of the product concept determines the fate of the product. In (Ziv Av and Reich, 2005) we presented a method – SOS (subjective objective system) – for generating optimal concepts in diverse disciplines. In this work, we extend SOS to generate robust product concepts.

In the context of SOS, robustness is defined as the stability of the optimal concept or configuration generated by SOS with respect to (1) variations in designers' subjective judgment, (2) variations in available technology, (3) variations in organization context, and (4) variation in customers' preferences. All these could have an impact on the results obtained by SOS. In order to assess the robustness, we run different tests with simulated changes and analyzed the results. For example, robustness with respect to designers' judgment was tested by varying such judgment and checking the stability of the solution to such variations. Robustness with respect to customer preferences was calculated by sampling different preferences and finding their related optimal concepts. This data was subsequently analyzed to find robust concepts as well as risky concepts. For each preference we also analyzed the local robustness of the solution; that is, how much can we change the customer preferences from the available estimation and maintain the same solution. If these variations are large, our confidence in the solution increases. The robustness with respect to other variations is analyzed similarly.

We first review studies related to robust concepts and configurations and SOS, the system that serves as the basis for the new study. Subsequently, we present the problem of finding robust concepts and an algorithmic solution. We conclude with a case study and its conclusions.

2 Review of robust concepts and configurations

The product of a design process progresses from general and abstract concept to concrete and detailed configuration during the design process. It is recognized that the initial stages of design have the most influence on the quality and success of the product. Yet, there is little support for design methods to assist designers in the initial stages of design. In contrast, detailed design enjoys enormous variety of tools that it even becomes an issue to select among them. An important set of the tools for detailed design are robust design methods.

By and large, these methods are used for fixing the parameters of systems so that their behavior variation is minimal with respect to external noise. Rephrased simply, these methods seek to "give designers some control over uncontrolled inputs." While this is highly useful in the detailed design stages, it bears no relevance to the critical initial design stages. Our goal is to develop a method that will help designer in creating design concepts that are robust to the potential variations in the context of the design. Such external variations include change in customer preferences, emergence of new technologies, organization evolution, and also subjective judgment of designers. By developing such method designers would be able to create robust design concepts or if they choose different concepts, they would be able to understand the impact of this decision on the risk of their project.

There has been work on robustness of design decisions (e.g., Ullman, 2001). A robust decision is one that is immune to various types and levels of noise or errors. We have also developed a new method for improving the robustness of classical decision-making problems (i.e., selection among available alternatives) given available resources (Schor and Reich, 2003). Such an approach is useful in a concept selection phase as a replacement for other methods such as Pugh's (1981) concept selection. When generating a product concept, the generation needs to be robust with respect to information quality used in the process and the nature of the process itself (e.g., various subjective ratings employs, calculation performed). These methods however, do not support concept generation; consequently, they alone cannot address the generation of robust product concepts.

Ford and Barkan (1995) and Andersson (1997) discussed the differences between concept design and other design stages and the need to introduce robustness into the concept design stage instead of waiting to introduce robustness into detailed design. Nevertheless, their treatment of the robustness was qualitative – introduce robustness as an objective that shapes the concept generation or quantitative – deal with existing design concepts that could be analyzed quantitatively. Others that studied robust concepts also referred to concepts whose detail allows analysis by quantitative means (e.g., Simpson *et al.*, 1999). In contrast, we would like to address the concept generation stage where information is scares, qualitative, subjective, prone to error, and evolves in time.

The strength of our robust concept generation is its foundation – SOS – a method that generates optimal concepts in a stage where very rough understanding of the design exist and the analysis is only qualitative.

3 An overview of SOS

SOS is a method for optimally generating product concepts (Ziv-Av and Reich, 2005). It creates the best concept that maximizes the satisfaction of a set of goals and adheres to specified constraints.

3.1 Definitions

We use the following definitions throughout the paper:

Customer characteristics are product properties that are specified by the customer or the product users.

Implementation characteristics are product properties derived from the context of the manufacturing organization including its capabilities, for example, a capability to mass-produce a product, which determines the product production cost.

Engineering environment is the setting in which the design takes place. In particular, it determines the product *building blocks*: product descriptors such as components, parameters, or technologies, used by designers to describe the design solution.

Constraints are dependencies and limitations placed on the use of various combinations of building blocks when creating the product concept.

An optimal objective product concept is the best product concept that can be found to maximize attaining the customer characteristics but without taking into account any resource, organization, or issues such as investment, risk, knowledge, etc.

An optimal subjective product concept is the best product concept that is independent of functionality, but addresses all implementation characteristics such as manufacturability, simplicity, cost, risk, investment, know-how, etc.

A decision layer is a part of the objective or subjective concept formulation that organizes the relevant information in relation to the layer topic (e.g., product simplicity). The layer topic is either a customer or implementation characteristic.

A *decision layer weight* is the relative importance placed by the customer and/or designer on each of the decision layers.

An optimal concept is formed by combining the formulations of the objective and subjective product concepts and solving it.

SOS is formulated as a maximization problem, where a function that includes all contributions, objective and subjective is expressed and maximized subject to constraints.

3.2 The layers organization

Figure 1 shows the general arrangement of m layers. There is a layer for each customer requirement or implementation characteristics. The vector D of length n denotes the engineering environment components and is the same for all layers. We assume the availability of a database of existing components or parameters, or that the design team that uses the method can provide building blocks for potential inclusion into the product concept.

In many areas, these building blocks are well identified (e.g., aircraft design); nonetheless, conceiving a product concept is a challenging task (e.g., defining the concept of a new aircraft).

The elements of D take 1 or 0 values depending on whether they are incorporated or not in the design concept.

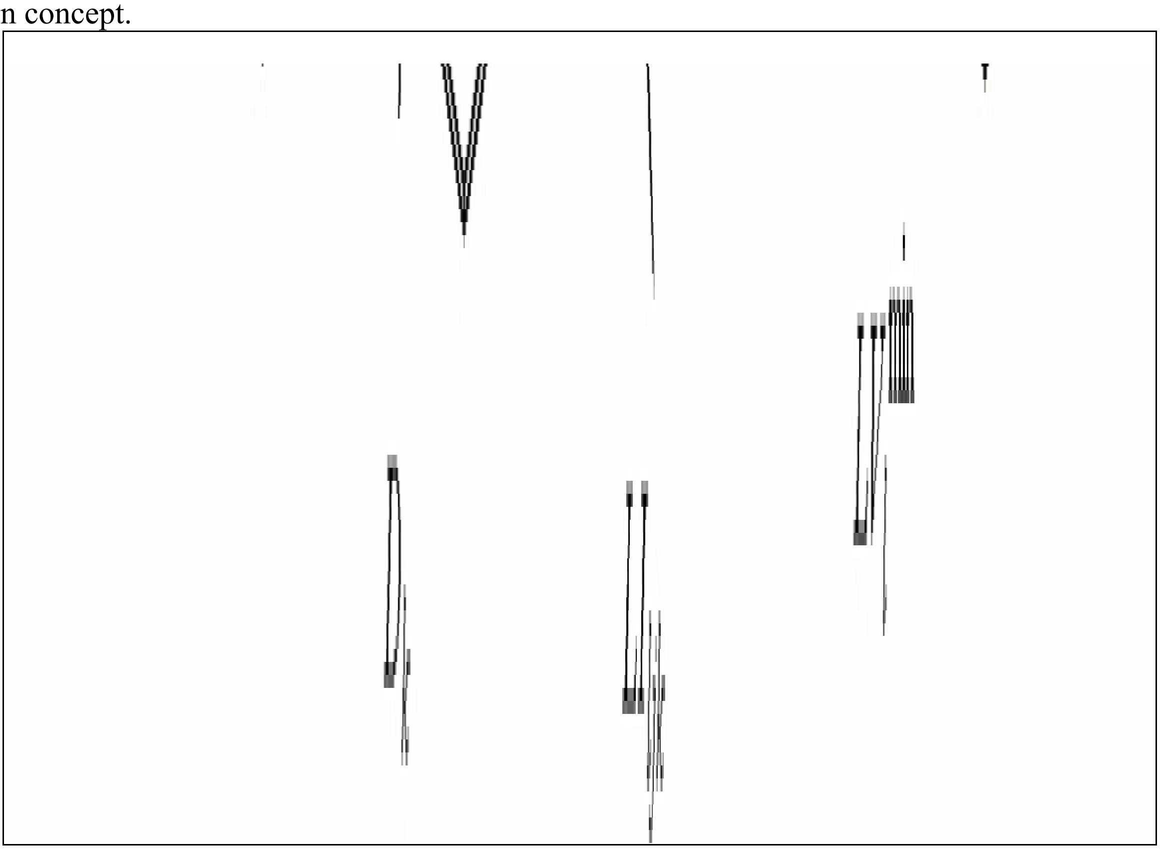

Figure 1. The arrangement of SOS decision layers

There are constraints between the candidate design building blocks. They are modeled by:

$$g_k\left(D_1,\ldots,D_n\right)\overset{\le}{\underset{\ge}{=}}b_k, k=1,\ldots,R \tag{1}$$

In SOS, the constraints g_k are linear functions of the independent variables D. This allows solving the problem as a regular optimization without resorting to combinatorial enumeration. This modeling can account for diverse constrains such as:

- *Mutual exclusiveness*: If three components D_1, D_2, D_3 compete to be incorporated in the product and only one could be selected then the constraint $D_1 + D_2 + D_3 = 1, D_j = 0,1, j = 1,2,3$, makes sure that only one would be selected for the design concept.
- *Functional necessity*: When component D_1 must be selected if component D_2 is selected we get $D_1 - D_2 \ge 0$. This works since if D_2 is set to 1, D_1 must be set to 1 also in order to satisfy the equation. If D_2 is set to 0 (not selected), D_1 can assume any value to satisfy the equation.

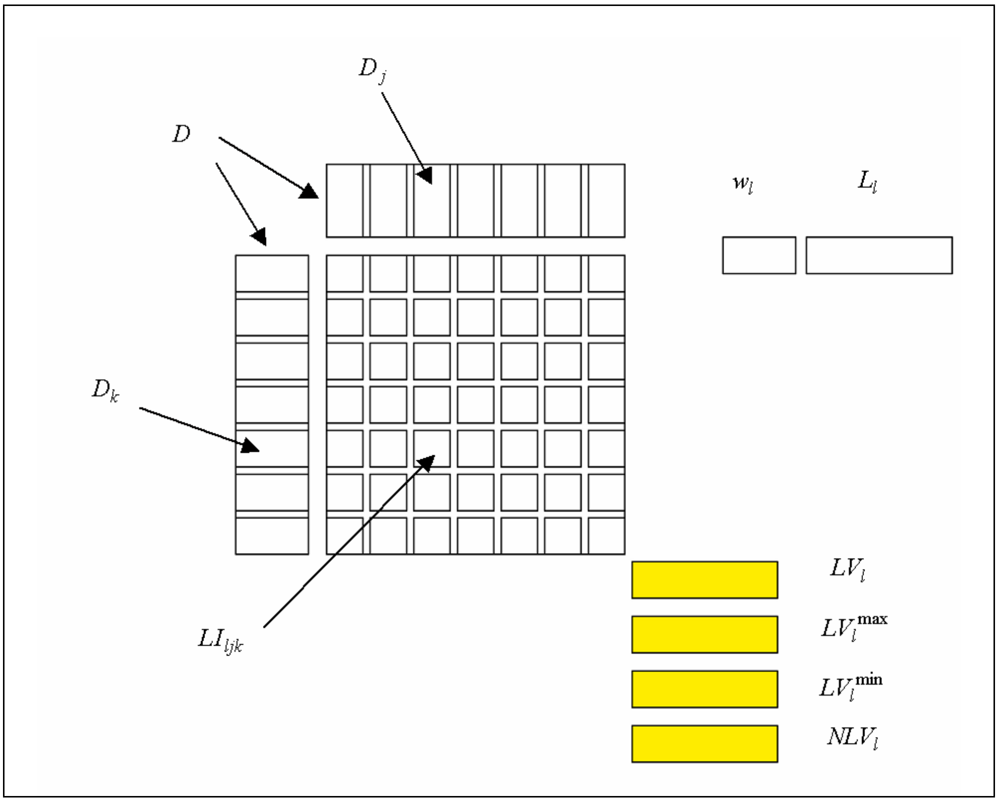

Figure 2. The layout of a single decision layer

Figure 2 shows the arrangement of information in the l^{th} layer. The matrix LI_l denotes the influence of the engineering environment components D on attaining the customer or implementation characteristic L_l. Each entry $LI_{ljk}, LI_{ljj} \in \{-1,0,1\}$, in the matrix specifies how much the incorporation of the two design building blocks D_j and D_k is assisting in attaining the overall value of the layer. The diagonal elements LI_{ljj} simply specify the contribution of D_j towards LV_l.

$$LV_l = \sum_{j=1}^{n} D_j \cdot \sum_{k=1}^{n} LI_{ljk} \cdot D_k \tag{2}$$

If we solve the following optimization problem:

$$\begin{gathered} \max LV_l \\ \text{subject to:} \\ g_k(D_1,\ldots,D_n) \underset{\geq}{\overset{\leq}{=}} b_k, k = 1,\ldots,R \\ D_j = 0,1,\ j = 1,\ldots,n \end{gathered} \tag{3}$$

We would get the best combination of *D* that maximizes the value of the layer and satisfies the constraints. We denote this value by $LV_l^{\max}$. Similarly, we denote by $LV_l^{\min}$ the minimal value of the layer obtained by the worst combination of *D*.

The normalized layer value is given by:

$$NLV_l = \frac{LV_l - LV_l^{\min}}{LV_l^{\max} - LV_l^{\min}} \tag{4}$$

The optimal solution takes into account the contribution of all layers. In the formulation, each characteristic *l* (whether customer or implementation) is assigned a weight w_l. Consequently, the problem becomes:

$$\begin{gathered} \max Q = \max \sum_{l=1}^{m} w_l \cdot NLV_l \\ \text{subject to:} \\ g_k(D_1, \ldots, D_n) \overset{\le}{\underset{\ge}{=}} b_k, k = 1, \ldots, R \\ \sum_{l=1}^{m} w_l = 1, \quad 0 \le w_l \\ D_j = 0{,}1, j = 1, \ldots, n \end{gathered} \tag{5}$$

Note that $Q \in [0,1]$. This formulation is an integer quadratic programming with linear constraints, which is easily solved by a variety of numerical techniques (Grossmann, 2002).

The mathematical formulation does not differentiate between different layers. Nevertheless, conceptually, we subdivide them into the aforementioned objective and subjective types. The objective layers represent the contribution of the customer characteristics or requirements and the subjective represent the implementation characteristics. Therefore, the optimal objective solution is derived by only considering the objective layers and the subjective by taking into account the subjective layers. The objective solution becomes the target for attainment since it best addresses the customer requirements without constraining the solution by any context related aspect. Examples of using SOS and its potential could be found in (Reich and Ziv-Av, 2003; Ziv-Av and Reich, 2003, 2005).

4 Robust product concepts

In the context of SOS, we define two types of robust concepts. In general, *robust concept* is a product concept that remains stable as different evaluations in SOS varies due to different circumstances. For example, the values of layer weights w_l or matrix entries LI_{ljk} might change. The w_l values determine robustness to different customer markets while the LI_{ljk} values determine robustness with respect to engineering knowledge.

The first type of robustness – *global robust concept* – is defined as the concept that is most prevalent if we let SOS input values vary randomly in their allowable range. This is an operational definition because it specifies the method to find that concept:

1. Draw randomly a set of w_l values. Normalize each set to have a sum of 1.
2. For each value vector w_l, create the product concept.
3. Enumerate the number of each concept type discovered.

The second type of robustness – *local robust concept* – is defined as a concept designed for a particular set of inputs and that is remain intact even if these input values change significantly from their present values. This is also an operational definition:

4. Vary the values of w_l by gradually changing the values of each separately.
5. For each change, maintain the sum of weights as 1.
6. Create the product concept.
7. Increment to the next w_l once the product concept is different than before and record the previous value. Return to step 1.
8. Output the extreme values in which the product concept is still unchanged.

Since SOS automatically generates the concept from its inputs, we can run simulations with different input values and obtain the results that allow assessing the global and local robustness of a concept. In this section, we exemplify these definitions in the context of the layer weights w_l values and the design of a car. Similar studies could be performed for the LI_{ljk} values.

4.1 Global robust concept

Table 1 provides the description of the concept design problem: designing a car with the given building blocks to satisfy the product requirements. For a particular market segment, each product requirements has its own specified weight. For a real problem, these weights might be distributions obtained from customers and manufacturer surveys. In order to assess the global robustness of car product concepts, we varied the weights randomly and independently in the range of [0,1] by maintaining their sum as 1.

Table 1. Car concept design: Building blocks and requirements

Building Blocks	**Product Requirements**
1. Front Engine	1. Handling & stability
2. Rear engine	2. Comfort (Min. jostling)
3. Driver location before front wheels	3. Accident safety
4. Driver location behind front wheels	4. Off road maneuverability
5. Front drive	5. Minimal turning radius
6. Rear drive	6. Easy maintenance
7. 4 wheel drive	7. Min. investments
8. Central transmission	8. Min. production cost
9. Rigid axels	
10. Independent suspension	

The results for 100 random weights are shown in Table 2. The table lists for each concept the number of times it appeared in the 100 cases, and its building blocks. The most frequent concept appeared almost half of the times. This concept appears in cars such as Toyota's Previa and others. The second frequent car is based on simplifying and reducing cost of the most frequent concept; it appears in most of the American automakers commercial vehicles. These and the third concepts are quite robust. They are optimal for a large part of the market. Therefore, changes in customer perception and other factors, would not lead to significant reduction in their customer base.

Table 2. Robust and risky concepts

	Number of occurrences					

1	46	Front engine	Driver behind	Rear drive	Independent suspension	
2	26	Front engine	Driver behind	Rear drive	Rigid axels	
3	20	Front engine	Driver behind	Front drive	Independent suspension	
4	4	Front engine	Driver before	Rear drive	Rigid axels	
5	1	Front engine	Driver before	Rear drive	Independent suspension	
6	1	Front engine	Driver behind	4 wheel drive	Independent suspension	Central transfer
7	1	Rear engine	Driver behind	Rear drive	Independent suspension	
8	0	Rear engine	Driver behind	Rear drive	Rigid axels	
9	0	Rear engine	Driver behind	4 wheel drive	Independent suspension	Central transfer
10	0	Rear engine	Driver before	4 wheel drive	Rigid suspension	Central transfer

The concept that appears 4 times is the cheapest and simplest car and appears in some distribution cars produced in Japan (Kia) and Korea (Hyundai). The concepts that appeared once or even four times are risky. They are optimal only for a small or even negligible part of the market. If these concepts are not identified exactly, they could easily fail. A small evolution in customer perception might render these concepts suboptimal for their intended customers. There are also three feasible solutions, i.e., that satisfy the constraints, but did not show up in the 100 examples. They are certainly risky because even now they do not seem to have a market.

4.2 Local robust concept

In order to assess the local robustness of a concept, we took the most frequent concept and a set of weight values and varied one weight at a time in the positive and negative direction. At the same time, we maintained the total sum of 1 of the weights. We recorded the maximum variation in % and the negative variation that would still lead to obtaining the same concept design. If these variations are high, the design is robust.

Table 3 provides the local robustness data of the most frequent concept. It is clear that within the vicinity of the nominal weights, large deviations of the weights that might reflect errors in customer surveys or shifts in customer preferences, do not change the optimality of the concept. Consequently, we could consider this product to be robust.

It is interesting to check what might be the penalty in concept optimality if we select to implement not the optimal but the second or third concepts. A product whose optimality level is significantly beyond its competitors it a robust concept. In contrast, a product that is second to others with small deviation from the optimal, but behaves such in all the space, is also robust. Choosing it is slightly worse than the optimal but in all the market. Since SOS could generate all feasible solutions and not just the optimal, we could easily implement this exercise.

Table 3. Product local robustness

	Requirement	Initial	Max positive weight	Max negative weight

		requirement weight	deviation %	deviation %
1	Handling & stability	0.128	31	-94
2	Comfort (Min. jostling)	0.184	174	-87
3	Accident safety	0.209	153	-96
4	Off road maneuverability	0.164	73	-97
5	Minimal turning radius	0.105	229	-76
6	Easy maintenance	0.114	141	-35
7	Min. investment & risk	0.061	65	-65
8	Min. production cost	0.036	450	0

Table 4 shows the requirements and four sets of relative weights. These weights typify the following car manufacturers:

1. Conventional cars such as Volkswagen/ Ford
2. Expensive performance cars such as BMW
3. Transporter cars such as Hyundai or Kia
4. Cars such as Peugeot

Table 4. Competing concepts

	Requirement	**Relative weight of requirements**			
		1	2	3	4
1	Handling & stability	2	3	0	3
2	Comfort (Min. jostling)	2	3	0	3
3	Accident safety	2	3	0	2
4	Off road maneuverability	1	3	0	1
5	Minimal turning radius	1	1	3	1
6	Easy maintenance	2	1	3	2
7	Min. investment & risk	2	0	3	2
	Concept (Table 2)	**Normalized quality**			
1	1	100%	92%		
2	2	96%			86%
3	3	92%	96%		100%
4	4			100%	
5	5			88%	
6	6		100%		
7	7			76%	72%
8	8		0%		
9	9			0%	
10	10	0%			0%

For each column weights we list the concepts from Table 2 that maximize the requirements (100%), those that come second and third, and the worst concept (0% normalized quality). We observe that the first three concepts are also good concepts for requirements in which they are not optimal and that sometimes, selecting the second concept leads to negligible quality penalty. This information is useful if we get a constraint or a decision external to this analysis that force abandoning the best concept. In order to make judgment about a particular market segment, it is worthwhile to conduct the local robustness analysis and this analysis.

5 Product family as robust concept

Product family has been the subject of numerous studies (e.g., Dobrescu and Reich, 2003; Gonzalez-Zugasti *et al*., 1999; Simpson *et al*., 1999). In this paper, we use the concept of a product family to extend the concept of robust design. In doing so, we depart from the traditional work on product family and platform that mainly deals with complete or detailed designs. In contrast, we deal with product family and platform at the concept generation stage. In the context of SOS, instead of finding a design concept that is prevalent across the space of SOS input values as we did in the global robustness analysis, we define *robust platform concept* to be a product concept family that addresses several markets and whose common platform is almost completely specified. Consequently, implementing the platform concept in multiple markets involves minimal customization.

Figure 3 visually compares between the three definitions of robustness. It depicts three different cases of a concept defined by three requirements whose weights are (w_1, w_2, w_3). For each case, we collect four sets (w_1, w_2, w_3) of values.

1. Diamond ♦ – weights are in close proximity. The concept platform is completely defined and corresponds to the local robust concept.
2. Square ■ – weights are distributed across the full range, platform is partially specified by talking the shared building blocks of D_1, D_3, and D_4 concepts.
3. Circle ● – weights are distributed across a medium size range, platform is specified completely and is the same as the global robust concept D_3.

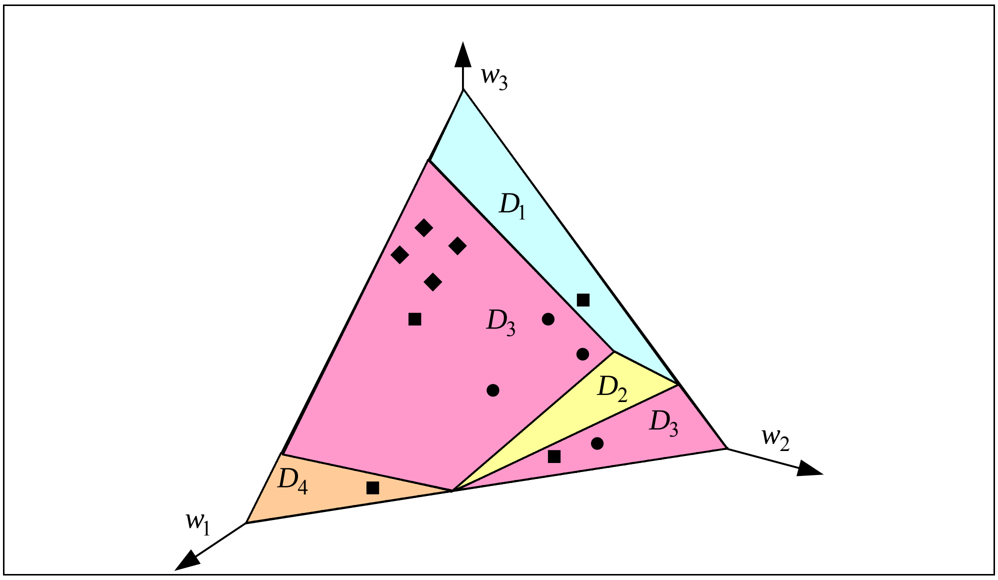

Figure 3. Comparison between different robustness definitions

The following notations are used in the product family model:

- The *weight of the market segment s* among the *d* markets is specified by the value WW_s that satisfies: $WW_s \geq 0, \quad \sum_{s=1}^{d} WW_s = 1, \quad s = 1,...,d$.
- A *concept vector for a market segment s* is defined by D_s.
- *Customization* is defined as the level of optimality of a member of a product family with respect to its intended market segment. A complete customization means that each market segment receives a concept optimized for its requirements weights. In this case, the concept platform is simply the building bocks shared by all specialized concepts.
- *Standardization* is the level of similarity of the building blocks between different members of a product family. A complete standardization means that all market segments receive the same concept which consequently might be suboptimal for all of them.
- *Customization weight WC* is the relative importance assigned to the customization with respect to the standardization, $0 \leq WC \leq 1$. Similarly, *standardization weight* is the relative importance of standardization with respect to customization 1-*WC*.
- The *weight of a requirement l* in a market segment *s* is W_{sl}, $s = 1,...,d, \; l = 1,...,m$.
- The *importance* assigned to having a building block *i* in the product concept platform is WP_i.
- The *similarity* between platform members in building block *i* is $P_i \in \{0,1\} \quad i = 1,...,n$. If $P_i = 1$, all concepts have to incorporate building block *i*, i.e., $D_{si} = 1, \; s = 1,...,d$.

The level of attaining requirement *l* in market segment *s* using the concept vector D_S is VL_{sl}, calculated by

$$LV_{sl} = \sum_{i=1}^{n-1} D_{si} \sum_{j=i}^{n} LI_{lij} D_{sj} \; . \tag{6}$$

The normalized score of VL_{sl} is NVL_{sl} and is calculated similarly to Eq. 4 by

$$NLV_{sl} = \frac{LV_{sl} - LV_l^{\min}}{LV_l^{\max} - LV_l^{\min}} \; . \tag{7}$$

The objective function would be

$$\max Q = \max \left[(1 - WC) \sum_{i=1}^{n} WP_i \cdot P_i + WC \sum_{s=1}^{d} WW_s \sum_{l=1}^{m} W_{sl} \cdot NLV_{sl} \right]$$

subject to:

$$g_k(D_{s1},\ldots,D_{sn}) \begin{array}{c} \geq \\ = \\ \leq \end{array} b_k, \; k = 1,\ldots,R, \; s = 1,\ldots,d \tag{8}$$

$$D_{si} \geq P_i, \; s = 1,...,d, \; i = 1,...,n$$

$$P_i \in \{0,1\}, \; i = 1,...,n$$

$$D_{si} \in \{0,1\},\ s = 1,\ldots,d,\ i = 1,\ldots,n\,.$$

In order to demonstrate this formulation, consider a case where:

1. $WP = (3,3,3,3,1,1,1,0,2,2)$. This means that the first 4 building blocks have high importance of being in the platform compared to low importance of the last two. The other building blocks could be omitted from the platform.
2. $WC = 0.25$, i.e., standardization is three times more important than customization.
3. $WW = (3,1,1,2)$. Therefore, the first market segment is most important, the last one has medium importance and the other two are least important.

The solution of the maximization problem leads to obtaining four different concepts with a concept platform consisting of: 'Front engine' and 'Driver behind front wheels.' If the importance of standardization increases, the platform becomes a complete concept with the addition of 'Rear drive' and 'Independent suspension.' The more standardization we demand, the more the design satisfies diverse market segments but the less it satisfies each particular segment. It is therefore a means to trade the amount of robustness or risk reduction with customer satisfaction. As such, it is a common product development tradeoff.

6 Conclusion

Based on our concept generation method SOS, we developed a method for the generation of robust product concepts. The method improves the confidence of designers in their solution or allows them to trade their confidence with some other aspects of the design. The method has start being used in the design practice of the 2^{nd} author and we foresee its practical benefits. In the future we intend to further extend it so that product robustness is treated by SOS just as another design issue. In this paper, we compared with, and distinguished our approach from other studies on product robustness.

We implemented robustness estimation methods and tested them on several design problems. In one problem – the design of a particular car - we find that our robust solutions correspond to the most used concepts in the car industry, whereas risky concepts are seldom used or were used in the past by various automakers and were replaced by more robust concepts. Other variations that we checked led to the anticipated results, i.e., a major variation in subjective judgment led to a change in the design concept in the way we anticipated from the variation. Similar results were obtained by analyzing several design projects.

We also shown how to formulate the problem of creating product concept platform and related it to the robustness of concepts. We provided an example that shows how it could be used. In the future, we intend to extend the robustness analysis with our previously developed decision-making robustness method (Schor and Reich, 2003).

References

[1] Andersson P., On robust design in the conceptual design phase: A qualitative approach, *Journal of Engineering Design*, 8(1):75-89, 1997.

[2] Dobrescu G. and Reich Y., Progressive sharing of modules among product variants, *Computer-Aided Design*, 35(9):791–806, 2003.

[3] Ford R. B. and Barkan P., Beyond parameter design – A methodology addressing product robustness at the concept formation stage, *Proceedings of the National Design Engineering Conference*, pp. 1-7, 1995.

[4] Gonzalez-Zugasti J., P., Finch W. and Otto K., Set-based product platform requirement definition, *International Conference on Engineering Design, ICED 99* Munich, 1999.

[5] Grossmann I. E., Review of nonlinear mixed-integer and disjunctive programming techniques, *Optimization and Engineering*, 3:227-252, 2002.

[6] Pugh S., Concept selection, a method that works, *Proceedings of the International conference on Engineering Design*, (ICED 81), 1981.

[7] Reich Y. and Ziv-Av A., A comprehensive optimal product concept generation framework, in *Proceedings of the 15th International Conference on Design Theory and Methodology (DTM)*, (New York, NY), ASME, 2003.

[8] Simpson T. W., Chen W., Allen J.K. and Mistree F., Use of the robust concept exploration method to facilitate the design of a family of products, in *Simultaneous Engineering: Methodologies and Applications*, pp. 247-278, (U. Roy, J.M. Usher and H.R. Parsaei, Eds.), Chapman-Hall, New York, NY, 1999.

[9] Schor A. and Reich Y., Improving the robustness of multicriteria decision making, in *CD-ROM Proceedings of the 14th International Conference on Engineering Design (ICED)*, The Design Society, 2003.

[10] Taguchi G., *System of Experimental Design*, vols. 1 and 2. UNIPUB/Krauss International, White Plains, NY, 1987.

[11] Ullman D. G., Robust decision-making for engineering design, *Journal of Engineering Design*, 12(1):3-13, 2001.

[12] Ziv-Av A. and Reich Y., A comprehensive optimal product concept generation framework, in *CD-ROM Proceedings of the 14th International Conference on Engineering Design (ICED)*, The Design Society, 2003.

[13] Ziv-Av A. and Reich Y., SOS – Subjective objective system for generating optimal product concepts, *Design Studies*, in press, 2005.

Yoram Reich
Center for Design Research, Stanford University
424 Panama Mall
Stanford, CA 94305-2232
USA
Phone: 650-723-7911
Fax: 650-725-8475
E-mail: yreich@stanford.edu / yoram@eng.tau.ac.il
(On sabbatical from The School of Mechanical Engineering, Tel Aviv University)