ENABLING SET-BASED CONCURRENT ENGINEERING IN TRADITIONAL PRODUCT DEVELOPMENT

Dag Raudberget
School of Engineering, Jönköping University, Sweden

ABSTRACT
Set-Based Concurrent Engineering is described as an effective methodology for product development, but is also hard to implement in companies using traditional development processes. This paper suggests a new way to introduce Set-Based Concurrent Engineering by combining its three principles with a modified morphological chart. A structured process with design templates is proposed and incorporated in a computer tool. The approach is evaluated by using information from an industrial case study. The result indicates that the principles of Set-based Concurrent Engineering can be implemented in a traditional development process by the proposed process and computer tool.

Keywords: Set-based Concurrent Engineering, morphological chart, case study, design method, computer support

1 INTRODUCTION
The design phase has a major influence on the cost and quality of a product, and the improvement of design methods and processes therefore renders a high interest in industry and academy. One methodology that has received positive attention is Set-Based Concurrent Engineering [1]. The concept of Set-Based Concurrent Engineering (SBCE) is to develop multiple sets of solutions in parallel. As the design evolves, the sets of solutions are gradually narrowed down based on relevant information from customers, manufacturing departments, tests, and other sources. In end, only the optimal solution is left.
The Set-based process may seem inefficient, but some authors claim SBCE and other lean practices to be four times more productive than traditional phase-gate models [2, 3]. Unfortunately, SBCE is also described as the hardest part of Lean Product Development to implement [4]. One reason for this may be that the connection to existing development methodology is unclear. Another reason is that SBCE is not a concrete, prescriptive methodology. Instead it relies on three principles [5] that must be adapted to each firm. There is also a lack practical examples, and the published applications of SBCE in industry have so far been limited to three primary cases: The first case was performed at Toyota Motor Corporation [1, 6, 7] where the methodology was first developed. The other cases are studies of introduction of SBCE in Scandinavian companies [8] and at a Swiss company [9].
Given the barriers for designers to work Set-based, the objective of the work was to develop a structured process to introduce SBCE in a traditional development environment. The objective was also to investigate the usefulness of this process by using information from an industrial case study. To ease calculations, the process was incorporated in a computer tool. The research approach was based on studies of literature, and experiences from previous Set-based pilot projects.

2 FRAME OF REFERENCE
In my comprehension, SBCE is the practical engineering part of Lean Product Development. The other parts are concerned with management or the organisation of a company and are not explored in this paper. In spite of the importance of these practical engineering activities, there are relatively little published studies about the relationships between SBCE and traditional development practices. In literature, SBCE is described as a different paradigm of design [1, 3, 6] . It is also claimed that SBCE cannot be used on its own since it is integrated with Lean development [4]. However, previous research indicates that there are traditional design practices that can be adapted to capture the essence of SBCE in an engineering process [8, 10].

2.1 Set-Based Concurrent Engineering

The term "Set-Based Concurrent Engineering" was introduced by Ward et. al. [1] after studies of the development process at Toyota Motor Corporation. Originally, SBCE included most components of Lean Product development, but nowadays the term "set-based" has a different meaning. SBCE is often described as a process of developing multiple solutions [3, 4]. The Set-based approach represents a different way of organizing and executing development compared to the traditional development methodology, characterized by an early selection of one "best" specific solution for further development.

One distinction is that SBCE doesn't require an early establishment of a system level design; instead different sets of possibilities for each subsystem are explored. This allows decisions to be delayed and design options to remain open until sufficient knowledge of functional and manufacturing perspectives exists [11]. Another difference is that SBCE works better with initially loosely defined requirements, instead of a traditional need for a well defended specification [12]. This flexibility allows an extra degree of freedom to tune all parts of the system, rather than optimising individual parts under fixed constraints. Also, in practice, the decision process is different, based on the elimination of unfeasible parts, rather than the selection of a best alternative [10].

Figure 1 describes the three principles of SBCE as stated by Sobek et. al. [5]:

1. Map the design space:	Define feasible regions, explore trade-offs by designing multiple alternatives and communicate sets of possibilities.
2. Integrate by intersection:	Look for intersections of feasible sets, impose minimum constraint and seek conceptual robustness.
3. Establish feasibility before commitment:	Narrow sets gradually while increasing detail, stay within sets once committed and control by managing uncertainty at process gates.

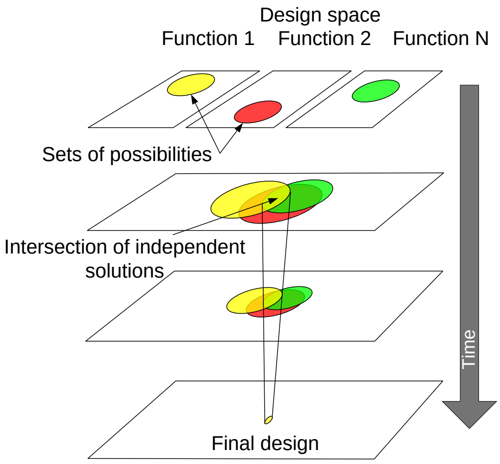


Figure 1. The principles of Set-Based Concurrent Engineering [8].

Although simple, these principles provided a useful guideline to introduce SBCE in an earlier research [8], and the applications will be discussed in more detail the next section.

2.2 Related examples of computer support for Set-based concurrent engineering

There are related examples of computer tools that support different aspects of SBCE. The first description of a Set-based process was made by Ward [13], based on his studies on constraint programming. Another application is the mathematical modelling of some aspects of product development, such as Set-based optimisations methods [14, 15]. Also simulations of the communication between engineers have been investigated, to compare the effectiveness of a set-based approach to traditional processes [16]. However, none of the prior studies has created computer support that includes all three SBCE principles.

3 ENABLING SET-BASED CONCURRENT ENGINEERING

The traditional design methods, represented by Pahl et. al. [17], require designers to select a best solution at an earlier stage of development than in SBCE. SBCE is thereby allowing better decisions, since the knowledge of the alternatives is more substantial [11]. In the author´s point of view, the most important difference between a Set-based approach and traditional methods is that the later do not reveal what is *possible* to design. Therefore, the key features that a process should enable are:

- Design of parallel alternatives and multiple sub-systems
- Elimination of the least suitable solutions instead of selection of the best solution.
- Promoting decisions to be based on a rational reason or facts, rather than opinions.
- Supporting multiple small decisions instead of single, important decisions.
- Supporting evolving designs with increased level of detail
- Supporting a step-vice evolution of requirements

To enable Set-based Concurrent Engineering, the above features must conform to its three principles.

3.1. Means to "Map the design space" (principle 1)

The first principle, "map the design space", starts with the definition of feasible regions. The regions are built up by designing multiple alternatives. Here, no new approaches are needed and there are sufficient creative methods and systematic approaches available. Traditionally the design space is pruned through the concept selection before any significant design work starts. However, prior studies found that it is possible to develop multiple alternatives one step further, provided that resources are available [8].

One barrier for introducing SBCE is that engineers are reluctant to share unfinished designs and therefore will not automatically communicate the sets of possibilities [8]. Another barrier is that SBCE is often described by the graphic illustration in Figure 2. This suggests that it is possible to use set theory to find the intersection between the sets, and that the convergence of the process is controlled by narrowing the circles. This description is simple and mathematically elegant, but has the disadvantage to be abstract and hard to implement.

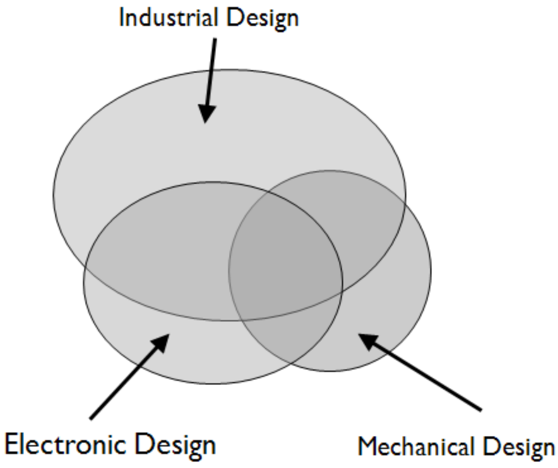


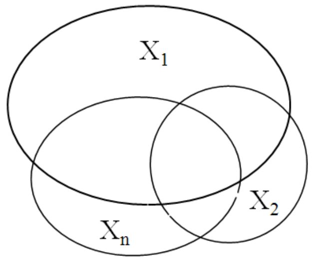
Figure 2. The set-theory view.

A well known tool that can be adapted to fulfil the first principle is the morphological chart [18]. In the morphological chart, two novelties are introduced:

- Each set is identified with a row in the morphological chart, and the sets are thereby becoming concrete, tangible objects.
- The chart is extended by the introduction of non-functional elements. A traditional morphological chart only contains solutions to product functions, but in the proposed process, also the

possible business cases, manufacturing processes, styling or other important aspects can be included.

To create the mapping of the design space, traditional creative methods are used to generate concepts for an extended morphological chart. Here, each function and its conceptual solutions are denoted in rows, representing the different sets (Figure 3).



Function					
X_1	The set	of	possible	industrial	designs
X_2	The set	of	possible	manufacturing	processes
....					
X_n	The set	of	possible	business	cases

Figure 3. The mapping of sets in a morphological chart.

The chart provides a visual overview of the possible combinations, thereby communicating the sets of possibilities. In order to explore trade-offs by designing multiple alternatives, the chart should be flexible and expanded with different solutions as the process proceeds.

3.2. Means to "Integrate by intersection" (principle 2)

The second principle, "Integrate by intersection", identifies intersections between the different sets by two different approaches. The first one is to find the region where members of the individual sets are compatible with each other. The other approach is to find the border lines where the current constraints are satisfied. For the second principle, the graphic illustration of Figure 2 suggests that it is possible to use set theory to find the intersection between the sets. Unfortunately, there is no physical representation connected to the figure, but the integration can be achieved by applying the principle of elimination to the sets and visualise the result in the morphological chart by introducing three novelties:

- The intersections of feasible sets can be found by an active search for the possible combinations where maximum compatibility between the alternatives exists. Traditionally, in the morphological chart, only one combination at a time is evaluated.
- The conceptual robustness comes from the search for the largest amount of compatible solutions. This can be achieved by strategic elimination of the right members of the set.
- Designs and functions that are incompatible with other sets are eliminated and the design reviews explicitly seek out these incompatibilities. If there is not enough information available for making a decision, the design remains the set and the decision is postponed until further investigations or tests are done.

3.4. Means to "Establish feasibility before commitment" (principle 3)

The essence of the third principle, "Establish feasibility before commitment", is to narrow the sets gradually at the same time as the amount of details of the remaining solutions is increased. In the proposed process, this is achieved by developing the remaining alternative one step further and then bring all the new information from simulations or tests together for evaluation. Information is used at the process gates to eliminate the least feasible alternatives by narrowing and adjusting the requirements, adding more requirements or by identifying incompatibilities. If there is not enough information to remove a solution confidently, more information on the different alternatives are sought.

Principle 3 also implies not to construct things that are incompatible with the other sets. If a new idea is introduced, it must fit into the system and it is not allowed to expand outside the specifications, such as committing to designs design where the constraints from the manufacturing process are unclear.

The morphological chart can still provide an overview of the different possibilities, but as the amount of information grows, the details must be stored in a different form.

4 A NOVEL PROCESS AND SUPPORTING COMPUTER TOOL

The suggested approach is intended to be used both at the system level design and at the detailed design level, and also to be flexible so designers can work with conceptual and detail level design at the same time. The second and third SBCE principles are intertwined in this process, and applied repeatedly. This follows the nature of design, where different parts are at a different state of development at a specific time ranging from well defined prototypes to imaginary parts. The process also supports the principle of “minimum constraints” [5], which implies to use the widest possible specifications at the current level of development. The form of the overview is not critical as long as it fulfils its purpose. Preferably it can be a company specific template that engineers can collaborate around.

The proposed design process can be summarised in the following activities, where the elimination of a solution is achieved by using information from tests, simulations, suppliers or other sources (Figure 4):

- Generate ideas
- Make conceptual designs/investigations
- Visualise solutions in an overview
- Identify sets of compatibility and eliminate unfeasible solutions
- Tighten or adjust constraints and eliminate unfeasible solutions
- Explore remaining possibilities by investigating/designing/testing one step further
- Visualise these new possibilities
- Eliminate unfeasible solutions and add or tighten constraints
- When constraints are unable to separate, use ranking methods
- Repeat the development/test/elimination process until the project is finished

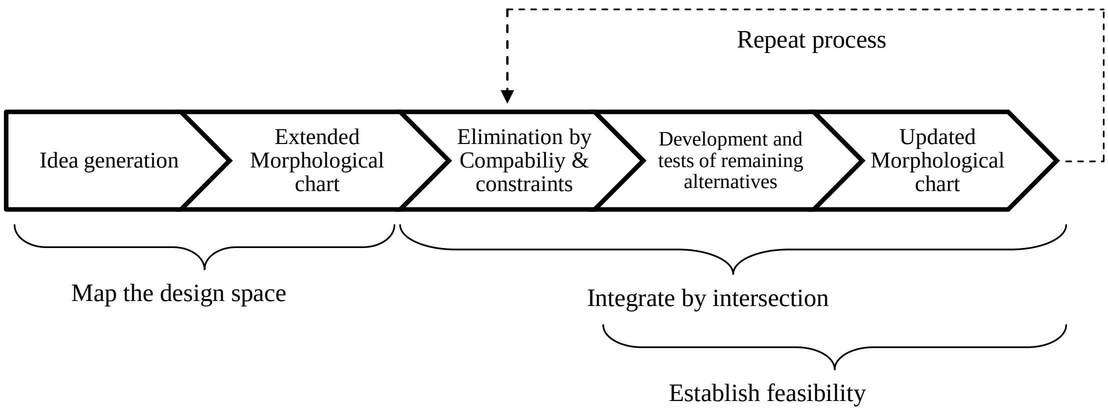


Figure 4. The proposed design process

4.1. Description of the computer tool

Even though the process could be implemented without computer support, manual handling of calculations and information is inconvenient when a larger amount of data is processed. Therefore a computer tool was developed. The computer tool provides a structured method to help designers work set-based, and it is not intended to replace existing product data management or requirement management systems.

The extended morphological chart was implemented in a spread sheet template with a pre-defined structure and hyper links providing the necessary basic functions. A spread-sheet concept was selected because of the ease of use and the speed of creating a prototype system. Also, necessary calculations can be done in the spread-sheet. The main function is to display information on different solutions, to link values and graphics and keep track of the requirement space. The chart can display multiple versions of a component or function, but the knowledge of the characteristics of those components can only be found out by design work and tests. This implies that there must be a structure that allows increased details. The tool is based on four templates (Figure 5):

- The “Morphological overview” is the master sheet, linking to additional information pages that provide deeper details of the specific solutions.
- The “Specification manager” that keeps track of the evolving specifications and the converging constraints.
- The “Technology manager” containing relevant information of the solutions.
- The “Morphological relations” template that stores the reasons for decisions.

The smallest part is the component, a concrete solution or possible solution described in the “Technology Manager”. The component is realised in a template with three pre-defined fields: Constraints, attributes/properties and a list of related sets as seen in Figure 5. Some properties are linked to values of the “Specification manager” that keeps track of the evolving specifications and the converging constraints. The “Morphological relations” template is dedicated to the reasons for decisions. It stores the design decisions and the facts that were used to make the decision. The purpose is to promote decisions based on knowledge and facts, instead of future expectations. It also has templates that evaluate compatibility between the different functional components of each set relative to *all* components of the other sets of solutions.

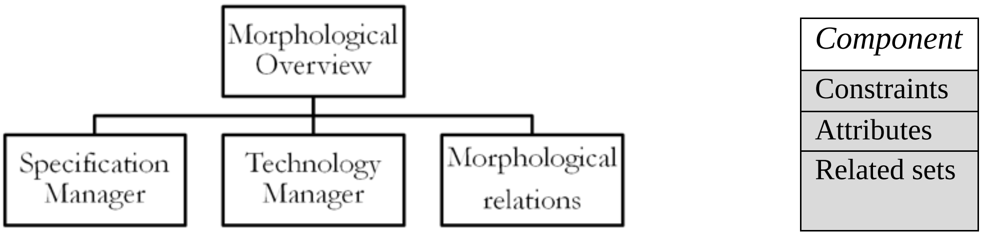


Figure 5.The structure of the proposed tool and a component

The templates are intended to be used from conceptual to detailed design, providing a flexible way to add a new level of detail or a new solution. To provide more details, components can be broken down into smaller parts and inserted on separate pages. Basic calculations can also be done, to keep track of cumulative properties such as weight, number of parts, and other facts. Once information is entered in the tool, it can be used for subsequent projects and hence serve as a design repository.

5 CASE STUDY

As an example of how this process is used, information collected from a finished development project is used. This project was a traditional point-based process that did not follow the principles of SBCE, and chosen because of the appropriate complexity allowing the different features of the computer tool to be evaluated. The original design process is first described and for reasons of confidentiality, some data are changed. Later, an application of this example using the new process is described.

5.1. Description of the development project

The product is a battery powered consumer product manufactured in large volumes. It has two basic functions: To clean and polish surfaces. The different modes of operations require different tools and different peripheral speed. The initial specifications were imprecise: Weight around 300 g, 30 min battery capacity, oscillating movement, service life 130 h. etc. At this stage of development, the goal was to lock down the requirements so that design work could proceed.
Initially, the design team found the design space shown in table 1:

Table 1. Initial design space

Set	*Solution*	
Styling	8 different designs	
Motor	Brushed DC	Stepper motor
Motor controller	PWM	Step-control
Battery	Rechargeable	

Speed control	Encoder measurement	Current measurement	Step-counting
Movement	Mechanism	Programmed	
Tool detection	Magnetic	Optic	Switch

In the first phase, broad investigations and conceptual design work were done. Also suitable stock parts were searched. It was decided to start the design work with the stepper motor concept, which offered a quiet operation, and an oscillating movement without the need of a mechanism. It also had the possibility to vary the stroke length and movement pattern of different tools. A preliminary design and a test rig with the capacity to run different types of motors were built to evaluate if the stepper solution could provide enough power. Generally, stepper motor systems are weak at high speed and this application required a large motor and a high voltage, implicating higher costs and a significantly larger and heavier product.

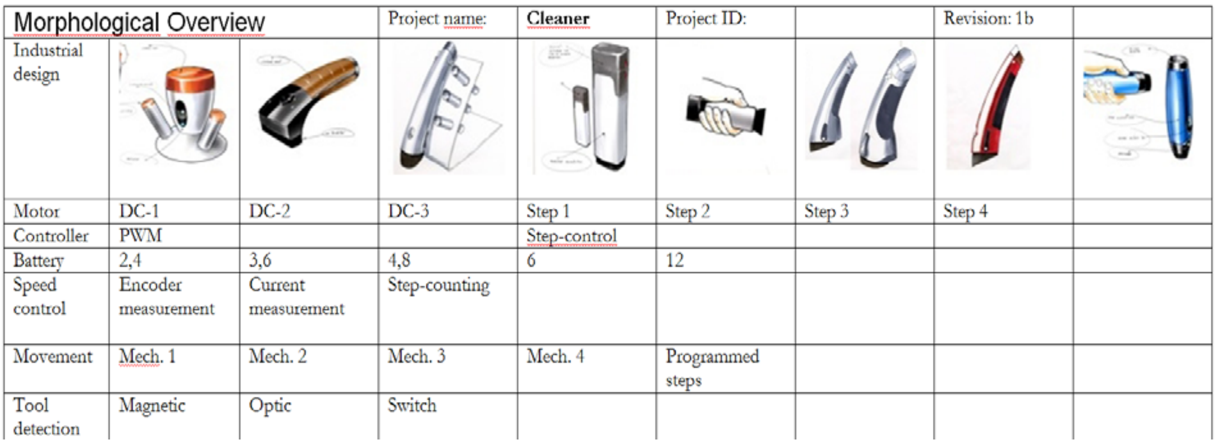
When the team discovered the shortcomings of this concept, it concentrated on investigating different DC motors and mechanisms. Initially, there were 3 different suitable stock motors and 4 mechanism concepts. The engineers selected one mechanism, and after design and tests, the conclusion was that the vibration levels were too high. After a redesign aiming at minimising the moving mass, another concept was chosen and the team started over again**.**

In parallel, the electronics team selected one DC motor as the target when developing the controller, and this task could be done relatively independent of the design of the mechanism. For the speed control of the DC motor, two suitable functions were identified. One uses an encoder to measure the shaft speed directly, the other measures the electric pulsations the motor emits to calculate the speed. Encoders require more hardware, so the electronics team quickly decided that the indirect measurement was a superior design. The result was an elegant but expensive controller, and a re-design by using less expensive parts was attempted. The results were poor and the design process had to be restarted with the encoder design. This required more physical space and the styling had to be changed, causing significant rework of the visual parts.

5.2. Implementation of Principle 1

By using information from the previous example, the initial design space may look as in Figure 6. The feasible regions of the design space are mapped and the multiple alternatives of each set are communicated, giving the designers an overview of the current possibilities and helping to keep track of compatibility. When a component in the chart is found unfeasible, it is marked.

The product can be described by different sets: Styling, electronics, mechanics etc. Worth noting is that, in the Figure 6, the sets "speed control" and "movement", are realised in different ways: Both as separate functions such in the encoder, and integrated as in the step motor drive. This indicates that the process supports different types of representation, from a strict functional decomposition to integrated function carriers and "organs" [19], fulfilling multiple functions solutions.



Morphological Overview		Project name:	Cleaner	Project ID:		Revision: 1b	
Industrial design							
Motor	DC-1	DC-2	DC-3	Step 1	Step 2	Step 3	Step 4
Controller	PWM			Step-control			
Battery	2,4	3,6	4,8	6	12		
Speed control	Encoder measurement	Current measurement	Step-counting				
Movement	Mech. 1	Mech. 2	Mech. 3	Mech. 4	Programmed steps		
Tool detection	Magnetic	Optic	Switch				

Figure 6. The initial Morphological overview

5.3. Implementation of Principle 2 and 3

The second and third principles are intertwined and the process now repeats principle 2 and 3 until only one solution remains. This follows the nature of design, where different parts simultaneously can be at a different state of development. In the present example, this is demonstrated by the speed control and tool detection both being conceptual solutions, while other systems have already gone through substantial development.

To find the intersections the first step is to increase the detail of different sets. All components created under Principle 1 are listed in their own row of a template and assigned to different sub-sets. The set number 5 has the subsets 51, 52 and 53, and new subsets can be added, see Figure 7.

A component can also be a member of many sets, an organ, fulfilling multiple functions. This is demonstrated by the component “Step-controller 1”, being a member of the sets “Controller”, “Speed control” and “Movement”. This is displayed by a green mark in corresponding column. It also has a yellow exclamation mark in the column 4 “Battery” that symbolises that it requires this set to work. The dependencies between sets and components are described only in one direction, exemplified by the motor requiring a controller, and the controller requiring a battery and a speed measurement. The reason for this is to make the description clear and avoid circular dependencies.

Morphological Relations

Project Name	Project ID	Revision
Cleaner	KA_1011	2010-09-18 DRT

Nr.	Instances	Active	Industrial design 1	Motors 2	Stepper motors 21	DC-motors 22	Controller 3	PWM 31	Step contol 32	Battery 4	Battery AA 41	Battery AAA 42	Battery Prismatic 43	Speed control 5	Encoder 51	Current 52	Step contol 53	Movement 6	Mechanism 61	Step Control 62	Tool detection 7	Magnetic 71	Optic 72	Switch 73
1	Stepper Motor A			✓	✓		!		!															
2	Stepper Motor B			✓	✓		!		!															
3	Stepper Motor C			✓	✓		!		!															
4	Stepper Motor D			✓	✓		!		!															
5	DC Motor A			✓		✓	!	!																
6	DC Motor B			✓		✓	!	!																
7	DC Motor C			✓		✓	!	!																
8	PWM-controller 1						✓	✓		!	!	!	!	!	!	!								
9	Step-controller 1						✓		✓	!	!	!	!	✓			✓	✓		✓				
10	Battery AA 2,4V_side									✓	✓													
11	Battery AA 3,6V_side									✓	✓													
12	Battery AA 4,8V_side									✓	✓													
13	Battery AA 6V_side									✓	✓													
14	Battery AA 2,4V_long									✓	✓													
15	Battery AA 3,6V_long									✓	✓													
16	Battery AA 4,8V_long_side									✓	✓													
17	Battery AAA 3,6V_side									✓		✓												
18	Battery AAA 4,8V_side									✓		✓												
19	Battery AAA 6V_side									✓		✓												
20	Battery AAA 12V_long_side								!	✓		✓												
21	Battery prismatic 12V								!	✓			✓											
22	Speed control_encoder 1													✓	✓									
23	Speed control_current 1													✓		✓								

Figure 7. The Morphological relations

Preferably, the process starts with “low hanging fruits”- readily available information that is easy to obtain. If a solution is proved inferior, by tests, simulations or other relevant information, its related components of other sets can be eliminated. This rapidly decreases the number of remaining alternatives. If set 32: “Step-control” is found unfeasible, all three stepper motors are also eliminated. The relative compatibility of each set compared to other sets can be investigated on other pages of the Morphological relations template. This is done by comparing every component to *all* components of the other sets of solutions. In the case of reducing the number of possible battery configurations, Figure 8, the batteries that can provide the correct voltage *and* storage capacity for a specific motor is marked, and some components are eliminated.

Battery compatibility														
Project Name Cleaner		Project ID KA_1011						Revision 2010-09-18 DRT						
		Active	Battery AA 2,4V_side	Battery AA 3,6V_side	Battery AA 4,8V_side	Battery AA 6V_side	Battery AA 2,4V_long	Battery AA 3,6V_long	Battery AA 4,8V_long_side	Battery AAA 3,6V_side	Battery AAA 4,8V_side	Battery AAA 6V_side	Battery AAA 12V_long_side	Battery prismatic 12V
			✗				✗							
Nr.	Instances													
1	Stepper Motor A												✓	✓
2	Stepper Motor B												✓	✓
3	Stepper Motor C												✓	✓
4	Stepper Motor D												✓	✓
5	DC Motor A			✓	✓	✓		✓	✓	✓	✓	✓		
6	DC Motor B				✓	✓			✓		✓	✓		
7	DC Motor C					✓						✓		

Figure 8. Compatibility template, a part of the Morphological relations template

Figure 8 is indicating that there is a watershed between two distinct conceptual solutions: The DC-motor and the stepper motor are not compatible. The design team selected the stepper concept for its initial design, but had to move back after some time due to the size and cost of the solution. In the proposed process, no premature choice of a best concept is done. The goal is to explore the concepts step-by-step to the point that it is possible to eliminate confidently one or a few inferior designs from the set of active solutions, not to get a complete knowledge of the design space.

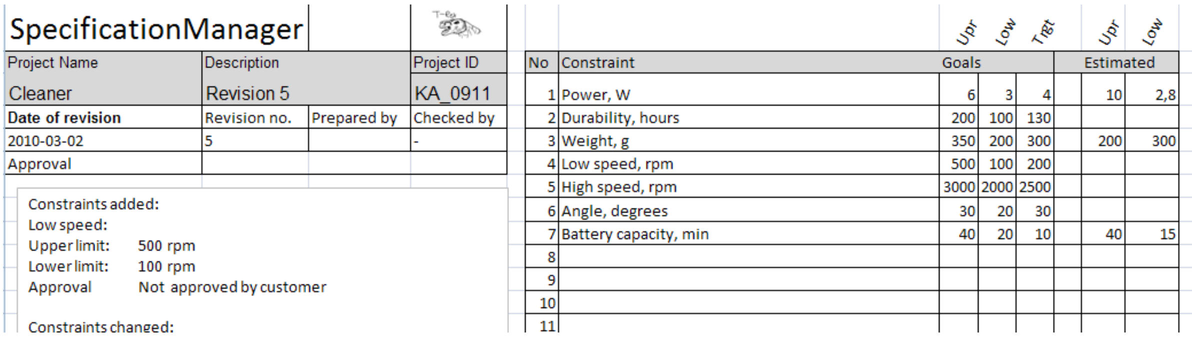
In this example, more information is needed to eliminate one of the motor technologies, and information can be found by continuing to investigate the compatibility between sets. The effort to investigate what battery configuration fits the different styling alternatives is significantly less than developing a solution and then realise that it is too large.

The next step is to apply more constraints. The current constraints and requirements, together with the previous revisions are stored in the "Specification manager" template. From specific requirements in the template, values can be linked to individual components if needed. The most important information on each component is put in a template of the "Technology manager", Figure 9.

TechnologyManager	2010-09-18			
Project Name Cleaner	Technology DC-motor		Project ID KA_0911	
DC-motor 1	Nichibo 5236_6V			
Constraint	Upper	Lower	Target	Actual
Power, W	6	3	4	3,4W
Durability, hours	200	100	130	100h
Low speed, rpm	500	100	200	200 rpm
High speed, rpm	3000	2000	2500	3000 rpm
Attributes	Upper	Lower	Target	Comment
Voltage, V	6	2,8	3,6	Datasheet: Lowest voltage to achieve power limit = 2,8V
Length, mm			52	
Diameter, mm			36	
Shaft diameter, mm			3,2	
Weight, g	200	50	140	
Cost, SEK	25	8	8	Batch size 5 000 or 40 000 units
Shaft diameter, mm			3,2	
Relations	Member	Requires		Comment
Motor	2			
DC-motor	22			

Figure 9. The Technology manager template

Here, attributes and relations to other sets are displayed, and important specifications are linked. Whenever possible, each requirement is an interval instead of a fixed number. By applying constraints, more solutions that fall outside the allowed range are eliminated and in this case the prismatic battery types could have been eliminated by the reason that they are twice as expensive as standard types. Basic calculations can be done in the templates of each component, and aggregate calculations, such as weight and cost, are done in the Specification manager (Figure 10) in the form of intervals.



SpecificationManager			
Project Name	Description		Project ID
Cleaner	Revision 5		KA_0911
Date of revision	Revision no.	Prepared by	Checked by
2010-03-02	5		-
Approval			

Constraints added:
Low speed:
Upper limit: 500 rpm
Lower limit: 100 rpm
Approval Not approved by customer

Constraints changed:

No	Constraint	Goals Upr	Goals Low	Goals Trgt	Estimated Upr	Estimated Low
1	Power, W	6	3	4	10	2,8
2	Durability, hours	200	100	130		
3	Weight, g	350	200	300	200	300
4	Low speed, rpm	500	100	200		
5	High speed, rpm	3000	2000	2500		
6	Angle, degrees	30	20	30		
7	Battery capacity, min	40	20	10	40	15
8						
9						
10						
11						

Figure 10. The Specification manager template

The process now proceeds with a repeated increase of details and elimination. When a set or component is eliminated, it is marked in the Morphological overview. At the same time, more details are created that needs to be added. An example of how the process could work is the development of the mechanism. Depending on the styling, the space available is distributed differently, and only some mechanisms and motors will fit a specific design. In the project, an evolving sequence of similar designs was developed, with the goal to arrive at a good compromise between the most favourable styling, motor, batteries and mechanism. The team arrived at a solution having high vibration levels. After a redesign to reduce vibrations, the team started over with another mechanism concept.

In the proposed approach, instead of starting a re-design, the other mechanism concepts should be developed to a level where it is possible to compare it to the failing mechanism. Also information from these failed designs could be used to evaluate the compatibility between specific motors, designs, batteries and motors and thereby reducing the amount of remaining solutions. In the case of the expensive controller, this problem could have been avoided by the proposed process. Since no selection of a best alternative is done, it requires both solutions to be explored one step further. Conceptual designs and a subsequent preliminary cost analysis of the alternatives are possible since both of the alternatives are based on commercial hardware with known prices.

At this stage of development, the Morphological overview could look like Figure 11, where the eliminated members are marked by red graphics.

Figure 11. The updated Morphological overview. Red graphics indicates eliminated solutions

The process continues in the same way for the remaining sets by generating solutions for the “Tool detection” and eliminating unfeasible components. A designer can be confident that any combination of batteries, motors and styling will be compatible, and there is no need to rush decisions on what specific details to use, until the constraints from incomplete components are understood.

6 CONCLUSION

The objective of the work was to develop a structured process to introduce Set-based Concurrent Engineering in traditional companies and to investigate the usefulness of the approach. The process was supported by a series of design templates in a computer tool. To evaluate the process, information from an earlier industrial case study was used. The result indicates that the proposed process can implement all three principles of SBCE successfully in a traditional development process by the proposed process and computer tool.

The scientific contribution is the combination of an adapted morphological chart and a structured application of the SBCE principles. Each set is represented by a row in the morphological chart and the sets are thereby becoming concrete, tangible objects. The morphological chart is extended also to represent non- functional qualities such as styling, production requirements, or different business cases. Another novelty is that the use of the morphological chart is reversed: instead of exploring one solution at a time, the intersections of feasible sets are found by focusing on the compatibility of multiple solutions. The other solutions are eliminated and the number of remaining combinations in the chart decreases.

The presented work needs further improvements, and the templates of the computer tool are filled with information that could be reused in subsequent development projects. Properly organised, this could be used as a design repository. Also, the principle of elimination must be supplemented by other methods, since it may be hard to separate solutions by requirements only; in the later stages of development, the remaining members of a set are more similar and may all fulfil the requirements.

The information used in the example was collected from a finished project, and this may have affected the outcome, since it's easy to be wise after the event. However, the author believes that the process can improve product development by introducing a Set-based working process and by promoting decisions based on knowledge and facts, instead of expectations. The best way to evaluate the usefulness of the process, is to test it with professional designers in a development project, which is the next step in the research process.

ACKNOWLEDGMENTS

The author would like to express his gratitude towards the participating company and the Swedish Governmental Agency for Innovation Systems (VINNOVA) under the program “Innovativ Produktutveckling” grant number 2005-02880.

REFERENCES

[1] Ward, A.C., Liker, J.K., Sobek, D.K. and Cristiano, J.J., Set-based concurrent engineering and Toyota. *ASME Design Theory and Methodology*, 1994, pp79–90.

[2] Ward, A.C., *Lean product and process development*. (Lean Enterprise Institute, Cambridge, MA, 2007).

[3] Morgan, J.M. and Liker, J.K., *The Toyota product development system: integrating people, process, and technology*. (Productivity Press, New York, NY, 2006).

[4] Kennedy, M., Harmon, K. and Minnock, E., *Ready, Set, Dominate: Implement Toyota's Set-Based Learning for Developing Products and Nobody Can Catch You*. (Oaklea Press, Richmond, Va, 2008).

[5] Sobek, D.K., Ward, A. and Liker, J., Toyota's Principles of Set-Based Concurrent Engineering. *Sloan Management Review*, 1999, 40(2), pp67-83.

[6] Sobek, D., K., Principles that shape product development systems,-a Toyota-Chrysler comparison. PhD thesis in Industrial and Operations Engineering. (University of Michigan, Ann Arbor, MI. 1997).

[7] Morgan, J.M., High performance product development: A systems approach to a lean product development process. PhD thesis in Industrial and Operations Engineering. (University of Michigan, Ann Arbor, MI. 2002).

[8] Raudberget, D., Practical Applications of Set-Based Concurrent Engineering in Industry. *Journal of Mechanical Engineering- Strojniški vestnik*, 2010, 56(11), pp685-695.

[9] Schmitt, P., Schulze, A. and Boutellier, R., How Set-based design fosters flexibility- Parallel design, delayed commitment and communication and coordination. (ETH Working paper, Chair for Technology and Innovation Management, Zurich, 2011).

[10] Raudberget, D., The decision process in Set-based Concurrent Engineering- An industrial case study. In *11th International Design Conference - DESIGN 2010*. Dubrovnik - Croatia. (Design Society)
[11] Ward, A., Liker, J.K., Cristiano, J.J. and Sobek, D.K., The second Toyota paradox: How delaying decisions can make better cars faster. *Sloan Management Review*, 1995, 36, pp43-61.
[12] Sobek, II, Durward, K. and Liker, J.K., Another look at how Toyota integrates product development. *Harvard business review*, 1998, 76(4), pp36.
[13] Ward, A.C., A theory of quantitative inference for artifact sets, applied to a mechanical design compiler. PhD thesis. (Massachusetts Institute of Technology 1989).
[14] Finch, W.W. and Ward, A.C., A set-based system for eliminating infeasible designs in engineering problems dominated by uncertainty. In *ASME Design Engineering Technical Conferences.* Sacramento, CA.
[15] Rao, S.S. and Lingtao, C., Optimum Design of Mechanical Systems Involving Interval Parameters. *Journal of Mechanical Design*, 2002, 124(3), pp465-472.
[16] Shahan, D., Madhavan, K. and Seepersad, C.C., The Implications of Alternative Multiscale Design Methods for Design Process Management. *International Conference on Engineering Design,* Paris, France, 2007.
[17] Pahl, G. and Beitz, W., *Engineering Design: A Systematic Approach*. (Springer, London, 2007).
[18] Zwicky, F., *The morphological approach to discovery, invention, research, and construction. In New Methods of Thought and Procedure*. (Springer- Verlag, New York, 1967).
[19] Hubka, V. and Eder, W.E., *General Procedural Model of Engineering Design*. (Heurista, Zürich, 1992).

Contact: Dag Raudberget
School of Engineering, Jönköping University
Department of Mechanical design
Box 1026
SE- 554 46 Jönköping
Sweden
+46 36 101676
+46 36 125331
dag.raudberget@jth.hj.se

Dag Raudberget is a senior lecturer at the Department of Mechanical design and a part- time PhD student focusing on Set-based Concurrent Engineering and Lean product development methods in industry.