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

PREPARING FOR A PRODUCT PLATFORM - PRODUCT FAMILY HIERARCHY PROCEDURE -

Niels Henrik Mortensen, Lone Munk, Ole Fiil-Nielsen

Keywords: Product Family, Family Architecture, Product platform, Generic Product.

1 Introduction

The strengths of utilizing platforms and family architectures in product development are widely recognized. We perceive *platforms* as the scheme by which companies consciously aim to introduce one or more families of products aiming for different market segments, while utilizing the commonalities within these families for mass-production. *Family architectures* are the documented structure of product families describing the composition of the product variants from a common perspective pin-pointing the commonalities and differences between them. Platforms and architectures are applied in various contexts for various purposes, the creation and implementation of platforms and architectures is however not simple.

The first task of setting up a platform is always a selection of included products. This includes updates of present products and future undeveloped products alike. Some companies simply seek to incorporate all of their products in the platform, while others act on an intuitive feeling of which products could benefit from a greater commonalty or variety. These unconscious approaches however often lead to unsuccessful platforms and very complex family architectures, which fail at addressing products to costumer demands.

The ultimate goal of platform projects is to offer a product diversity that mirrors market demands while keeping the rest of the product common. Hence the primary challenge of designing family architectures is the transformation of beneficial commonality and diversity within a product family into *modules* for reuse or/and for isolation of variance, and this modeling and structuring of diversity and commonality is a key-aspect in the creation of product development platforms and family architectures.

To reduce the complexity of the task and to improve the basis on which decisions concerning modules are made, we need a way of visualizing the relations of the product family in terms of commonality and variety. This visualization method can then be used as a tool for creating the product families, which platforms and family architectures depend upon. Additionally a classification of product variants in a hierarchy of relations could form the basis for creating modular family architectures.

In this article we introduce a sub-procedure, ‘The Product Family Hierarchy Procedure’ covering the early phases in an overall procedure for platform and architecture development. The procedure aids product developers in selecting products for a platform and setting up a draft for the modular structure.

2 Procedure requirements

A product family is established by selecting a number of present products, which could be updated and included in the family and by selecting a number of undeveloped (possibly very vague) future product ideas, which could be designed to fit the product family.

When selecting these products (i.e. both present and future) for a product family at least two issues must be dealt with:

- Deciding which products to include and exclude in a product family sounds trivial but is important for the success of the product platform. If a non-beneficial product is included in the product family, the creation of common family architectures is likely to be hard and the success of the entire platform project is likely to fail.
- The consequences of having several product families compared to one are not always obvious, but as different organizing of the products into product families are bound to have different effects on market, this organization of the product range into one or more product families is also important.

A procedure, which can offer guidelines for determining how to transform a traditional range of products into a number of product families taking the above issues into consideration, would greatly increase the chance of success for any platform project.

When the product family has been established, a family architecture describing reusable or exchangeable modules should be synthesized. The primary issue of architecture synthesis is the establishment of a modular structure. A number of procedures and tools for optimizing modular structures have been created and presented in state-of-the-art literature during the last decades ([2][5][11] and many more). Procedures for creating a preliminary or rough modular layout has however not been offered the same attention, and as a result product developers might lack a good starting point for designing the final modules.

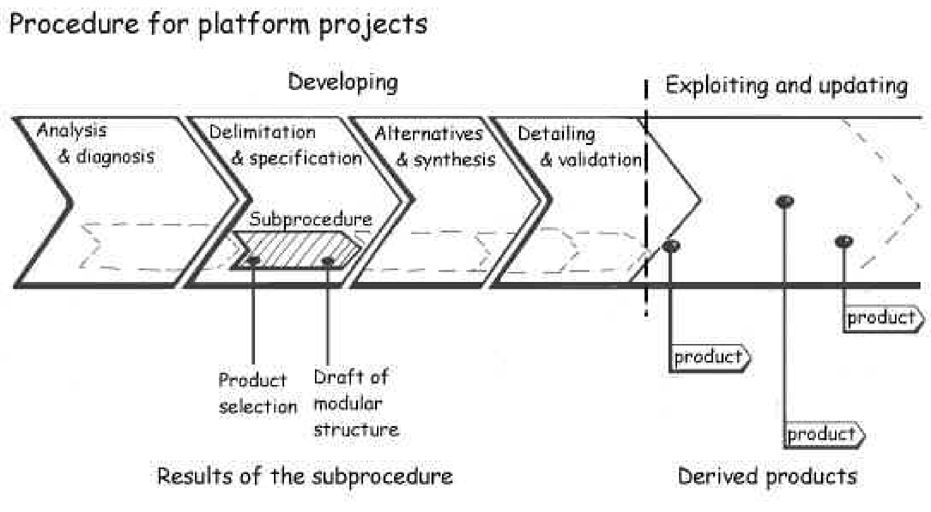

Figure 1. The figure shows a procedure for platform projects consisting of four development phases followed by a continuing process of exploitation and updating. The sub-procedure presented in this paper is applicable in the second of the four phases, which focuses on delimitation of the platform scope and specifies the demands for a solution. The product selection is a necessity in this phase and a rough modular structure is the starting point for the third phase of synthesis and alternatives.

We generally describe a procedure for platform and architecture development as shown on the figure below. The sub-procedure of selecting products and drafting a modular structure is mapped in the second phase.

We generally divide platform projects in four phases listed in the illustration below (Figure 1), and this paper presents a sub-procedure for phase two.

Finally as platform projects are primarily teamwork and requires the full commitment of the entire company any procedures must also offer a way of easy visualizing and communicating the generic structure and the relations between modules and product variants.

3 State-of-the-art in platform and architecture literature

State-of-the-art literature on product development platforms and architectures tend to overlook the importance of choosing the right product candidates for a product family. Selection of products are either based on intuitive commonality or not considered at all, and therefore a major disposition of the platform project is neglected.

The reason for this approach is a general tendency to use small-scale examples, which give an easier overview and visualization of the specific procedure, but fail to address the complexity of platform projects and product families in general. While small-scale examples do give an easier overview it is however important that the procedures are applicable in full-scale cases.

Several authors (including [9]) describe market demands in matrices and relate these to product variants. These tools have been used for inspiration for the procedure presented in this paper; they do not however describe a systematic approach for selecting possible products, instead a pre-selected product family is presented as a point of departure.

Instead state-of-the-art literature on product development platforms and architectures focuses mainly on procedures for generating improved product structures, and here the emphasis is on *improving* the modularity of the products rather than the *creation* of the modular structure itself.

Some literature focuses on modularization of a single product [8][10], and as a result the amount of reuse is restricted to generational reuse of subsystems through future updates.

Other literature focuses on a group of product variants with differences, which are not fundamental to the product, and for which similar functional diagrams can be sketched. The generic product structure is either based on a functional decomposition [2][5][11] or the traditional schematics of the product [7][9]. Variance based on the demands of targeted market segments is included and presented in tables like the differentiation- and the commonality-plan [7][9] or using the QFD-method [2][5] .

The differentiation- and commonality-plan and the QFD-method are powerful tools, but still the creation of the modular structure itself is not emphasized, and instead the emphasis is on improvement and optimization of the modular structure.

Recently publicised work [1][3] focuses on multiple products, which are not similar. These works are also founded on a functional decomposition basis, and the generic structure is characterized by a common diagram of functions. Dendrograms [3] are used to visualize the similarities between subsystems from a functional point of view. The procedure can be used to show similarity and diversity of corresponding subsystems, but cannot show several corresponding subsystems simultaneously. Therefore whole products or subsystem combinations cannot be evaluated using this method. The Modularity Matrix [1] offers a full visualization of product variants and subsystem combinations. The matrix-modules are however generated using functional decomposition and are therefore not related to customer demands in a direct way.

Our research of state-of-the-art literature has identified a potential for procedures dealing with early stage family architecture development, as these stages are only partially covered by existing literature. This includes the transformation of a traditional product range (including future products) into one or more product families and the creation of a rough preliminary modular layout for the family architecture on this basis. As the variance of a product family ideally is directly related to customer demands such procedure should be based on a market or customer approach.

4 Scientific method

Tools that meet (or partly meet) the requirements listed above have been developed as part of a cooperative project (Sep. 2003-May. 2004) between the Section of Engineering Design and Product Development at the Technical University of Denmark and the product development and marketing sections in the Danish company Martin Professional A/S (Martin). Martin is a world-wide developer and producer of intelligent lighting.

The approach used in the project is based on the Theory of Technical Systems (TTS) [4] and is partly inspired by object-based programming.

Because using Martins products, which consist of several hundred different parts, would require a lengthy technical introduction and a scale of diagrams, which would be too comprehensive for the length of this paper, a simpler example has been used for illustration. This example is the product portfolio of Vipp, a Danish company designing containers for different purposes: mainly dustbins, but also laundry and other bathroom applications (Se Figure 2). As the Vipp example is a relatively simple case the strength of the procedure when dealing with large product portfolios consisting of complex products does not directly show in this paper. Bear in mind however that the procedure has also been utilized on Martin's extensive and complex product portfolio.

Figure 2. Examples of the Vipp product portfolio: (from left) toilet brush and various examples of dust bins. To the right a sketch of the inside of a dust bin with the inner container and the lid-lifting mechanism with shock absorber.

5 Product Family Hierarchy Procedure

This procedure aims to meet the requirements listed previously. It is a tool for creating a product family from a loose range of products, and it makes it possible to establish variance for this product family and represents it in an unambiguous way.

The procedure is applicable for companies, which are planning to launch or re-launch a larger number of different products, while utilizing the commonalities between these products for reuse. The set of products or product ideas (i.e. loose product specifications) form the starting point of the procedure. Note however that all of these product ideas not necessarily are realized.

The procedure has four steps:

1. Identify *variance dimensions* and *generic products* by organizing the product ideas from a market or customer viewpoint in a *Product Family Hierarchy diagram*.
2. Establish one or more product families based upon the diagram and sound judgment.
3. Identify key components and attach them to generic products in the diagram.
4. Expand or reduce use of key components and group them into *draft modules* as a starting point for a modular structure.

These steps are explained in detail in the following subsections.

The result of the procedure is one or more structured product families with a draft modular structure based on market demands. The natural next step of a company having used this procedure is the design of the final modules, which is not covered in this paper.

5.1 Identifying variance dimensions and generic products

The Product Family Hierarchy Procedure is based on the perception that product variants forming a product family are derivatives of a common generic product. To isolate differences or benefice from commonalities, this generic product must be divided into modules. The modules are used for isolation or reuse depending on the point of view. As a result family architectures correspond to common generic products in a way where a family architecture describes the composition of the common generic product and how product variants are derived from it.

In an effort to describe the full variance of a product family we divide the common generic product into *variance dimensions* (e.g. height, width, color, and price). Within each of these dimensions several *1-dimensional generic products* can be derived from the common generic product. Each dimension represents a variable, which covers a number of alternative values – one for each generic product – the product variants can take on. In this case a value can be a specific value, an interval, a property or an overall application.

The Vipp containers have either oval or circular cross-sections. Hence the shape dimension has two 1-dimensional generic products (i.e. the oval generic container and the circular generic container) each with a specific value (i.e. oval and circular). All of Vipp's containers are either derived from the oval generic container or the circular generic container inheriting the value (i.e. oval or circular) of the shape dimension-variable from their respective generic product.

As the product variant inherits its values from one specific generic product in each variance dimension, the physical product variant is actually a derived combination of these generic products.

A Vipp container with a 'dustbin' application, a 'oval' cross-section, a 'small' size and a 'curved foot' style is a derivative of the common generic container, as well as a derived combination of the 'dustbin' generic product, the 'oval' generic container, the 'small' generic container, and the 'curved foot' generic container.

Using this mindset, we perceive the products in a number of different ways (i.e. one for each dimension) and the commonality and variety between the product variants are greatly affected by this (i.e. the pattern of generic products and derivations differs greatly between dimensions). This gives us an opportunity to revise each product variant, first however we will introduce a diagram for illustration.

The common generic product and the associated hierarchy of derived specified generic products can be visualized in a Product Family Hierarchy diagram (PFH diagram). A PFH diagram is created by classifying the products of a product portfolio according to their commonalities or similarities and relating them to each other through generic products. This approach leads to identification of the variance dimensions and a complete hierarchy of generic products.

There are of course many ways to classify products after their commonalities and similarities in general, but as the number of products variants and the justification of each product variant is founded in the market, products in product families should always be classified from a market or customer viewpoint.

Figure 3 shows the template of a PFH Diagram, showing the common generic product with a branch for each of the dimensions of variance, shown with the dotted lines. Within each of the dimensions there is further branching, one for each relevant generic product within the dimension. A detailed description of the creation is given below.

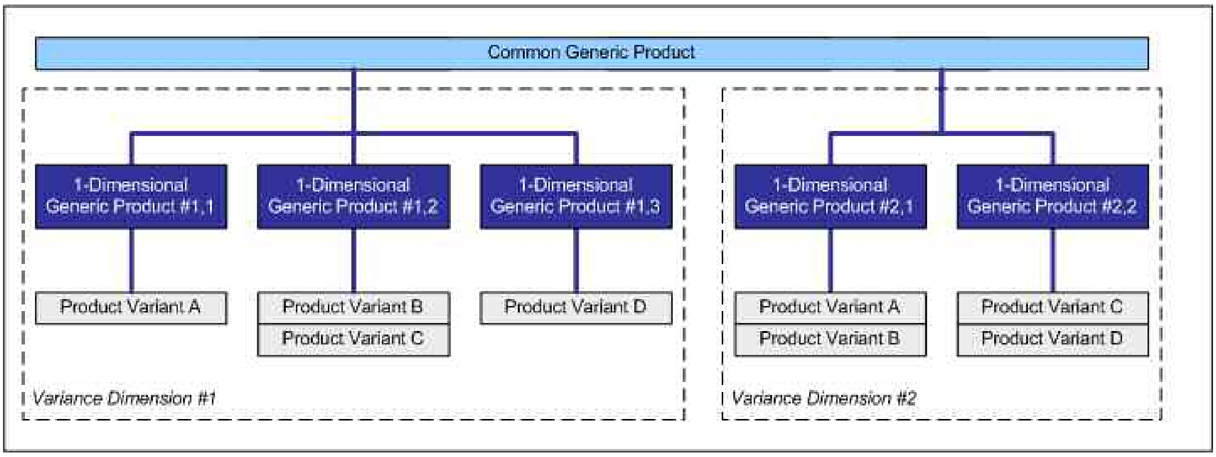

Figure 3. Template of the PFH diagram. Notice how all product variants appear in each variance dimension, because product variants are derived combinations of generic products.

The PFH diagram for the Vipp products is shown on Figure 4.

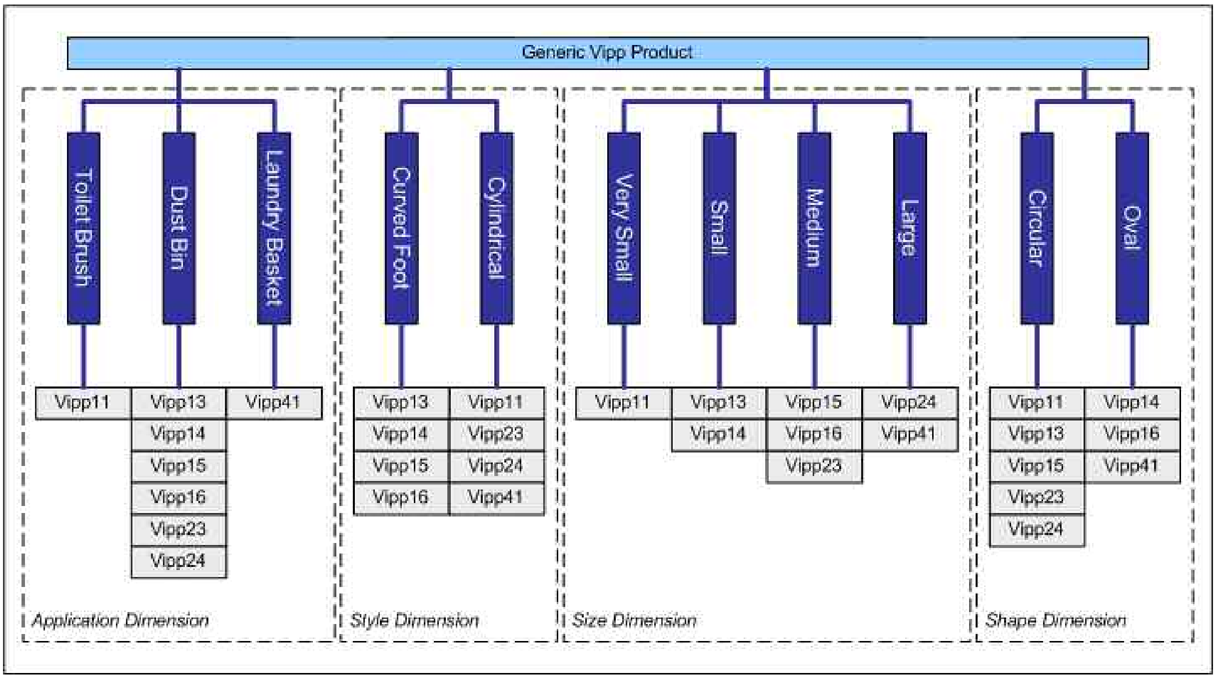

Figure 4. The PFH diagram for Vipp products. Four variance dimensions and eleven 1-dimensional generic products are sufficient in describing the complete product range.

The dimensions and the generic products can be identified from the customer demands for each product. The challenge is to 'translate' customer demands into a fixed set of values. Based on product descriptions five dimensions of product variance were identified for the Vipp portfolio: application, style, size, shape, and color (color is omitted in the diagrams to reduce complexity).

There are three guidelines for building a PFH diagram:

- The dimensions must be independent, so a products value in one dimension does not affect its values in any of the other dimensions. If a height variance dimension was included in the Vipp diagram, many invalid combinations would occur like a 'high' product with a 'very small' size. Note that there is a difference between invalid combinations and non-occurring combinations.
- The products each represent a unique combination of values. If this is not the case additional variance dimensions or generic products should be added (unless the products are too much alike and one of them should be terminated).
- Minimize the number of both dimensions and values (preferably the number of values) to reduce the number of invalid combinations.

As can be seen from the above guidelines, number two guideline may result in an increase of the diagram, while number three may result in a decrease. Hence it will often be an iterative process, where a diagram structure is proposed, compared with the products to evaluate it, a new structure is proposed and so on.

5.2 Establishing product families

Using the PFH diagram it is possible to identify a product by determining its value in each of the dimensions of variance. Product A is represented on Figure 4 by adding it to one generic product in each dimension. Showing all products in this way gives a clear picture of the possible product family.

The PFH diagram can be used to aid in the decision on which products to include in a product family:

- If any products share the same combination of generic products, they cover the same customer need, and one of them might be superfluous.
- Alternative combinations for new products can appear by systematic examination of the diagram.
- A product or a set of products are not beneficial for the architecture, if they expand the number of dimensions of variance or values in a radical way which offers a multitude of new theoretical combinations but few realizable and saleable ones.
- If a single product is the only reason why a specific generic product exists, then it must be considered if this product should be excluded from the architecture. First however it should be examined if this generic product has potential for any future product variants, and could be a basis for later expansion of the product family.

The products, which are excluded from the product family may still be realized, produced and sold – only not as a part of the created product families.

The importance of the dimensions may vary; some dimensions will provide far better commonalities than others. A product which is the only one with values in several (less important) dimensions may still be beneficial to include, if it shares values with a large group of products in important dimensions. Likewise a product can seem beneficial at first, if it shares values in many dimensions, but if it has its own value in a crucial dimension, it may be best to exclude.

*In the PFH diagram (*Figure 4*) for the Vipp products it is clear that the toilet brush is the only reason for two generic products (i.e. the generic toilet brush and the generic very small container) and hence it creates many invalid product combinations. This indicates that it should not be included in the same architecture as the rest of the products. The laundry basket is the only reason for one generic product, and is therefore also removed from the diagram. When removing the laundry basket (i.e. Vipp41) the largest dustbin becomes the only reason for the 'large' generic product in the diagram. The largest dustbin (Vipp24) however shares all of its other values with another product (Vipp23) and is therefore kept in the diagram. As a result we propose a product family for Vipp consisting of only dustbins, and that they design and manufacture toilet brushes and laundry baskets independently.*

5.3 Identifying key components and adding them to the PFH diagram

Family architectures partition generic products into modules. This partitioning is by no means simple, but a beneficial starting point can be created extending the PFH diagram by mapping key components in it and from their configuration derive *draft modules* for an architecture. These draft modules will then form a starting point or additional specification for the final modular structure.

Key components and modules can easily be described by mapping them in a PFH diagram by their respective belonging within the dimensions. To do this it is however necessary to detail the PFH diagram further. The diagram described up until this point describes a key component or module as either common for the entire product family (i.e. by attaching it to the common generic product) or belonging to a specified generic product in one of the dimensions of variance (i.e. by attaching it to a 1-dimensional generic product) – like the product representation. Since the key components and modules can be shared for a number of generic products in one dimension but not all, extra levels are put in between the specified generic products and the common generic product as show on Figure 5.

The key components can be identified by asking the question: 'What components make these products different?' This means that the key components that will be mapped in the diagram will not describe the full structure of the products – only the components that create valuable variety for the customers!

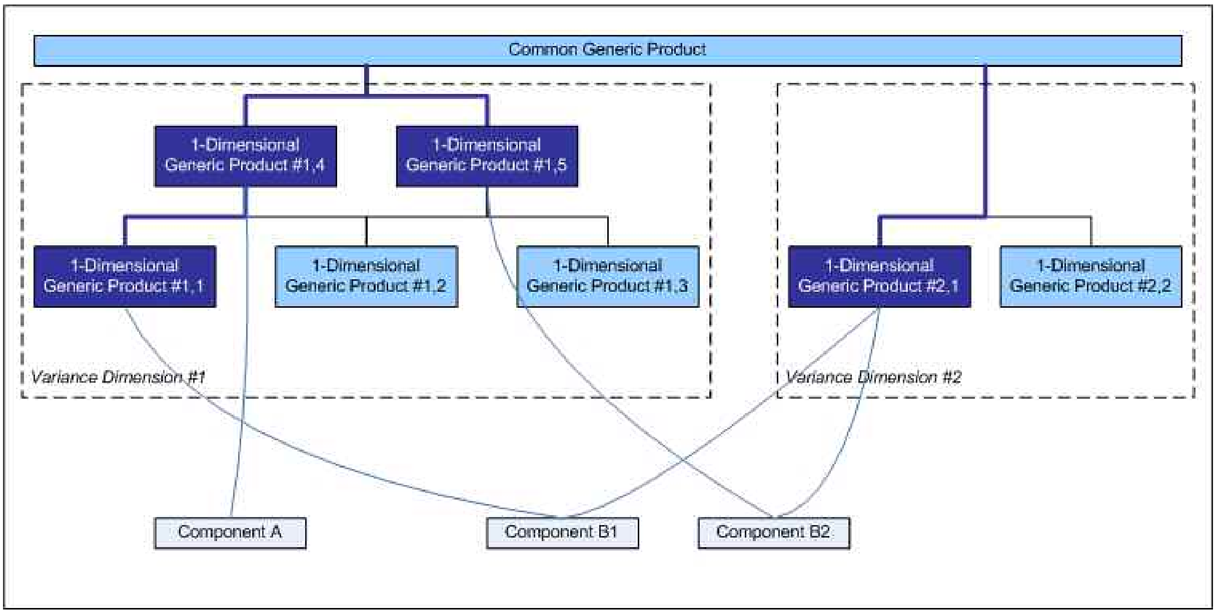

Figure 5. Template for the extended PFH diagram. Two kinds of components (A and B) are mapped into the diagram. Component type A only occurs in the products which can be derived from the generic product #1,4. This means that products derived from the generic product #1,3 do not contain component A – but all other products do contain it. Component type B covers two physical components. Component B1 occurs in products derived from the generic product #1,1 but only if these products are also derived from the generic product #2,1. Component B2 occurs in all other products derived from the generic product #2,1.

Each variance dimension should be investigated separately. For each generic product one should list the key components that make the derived products different from products derived from other generic products within the variance dimension and map their dependencies with other dimensions.

For the Vipp dustbins an example would be asking why dustbins of a small and medium size necessarily are different. The answer might be that they require different lids, bodies and bottoms – and that these elements are key components for creating the difference in size, that is valuable for the customers.

A selection of key components from the Vipp products is shown on Figure 6 through Figure 8. The selected key components are the bottom, the handles, and the pedal. They show different kinds of components and are described in details in the diagram captions.

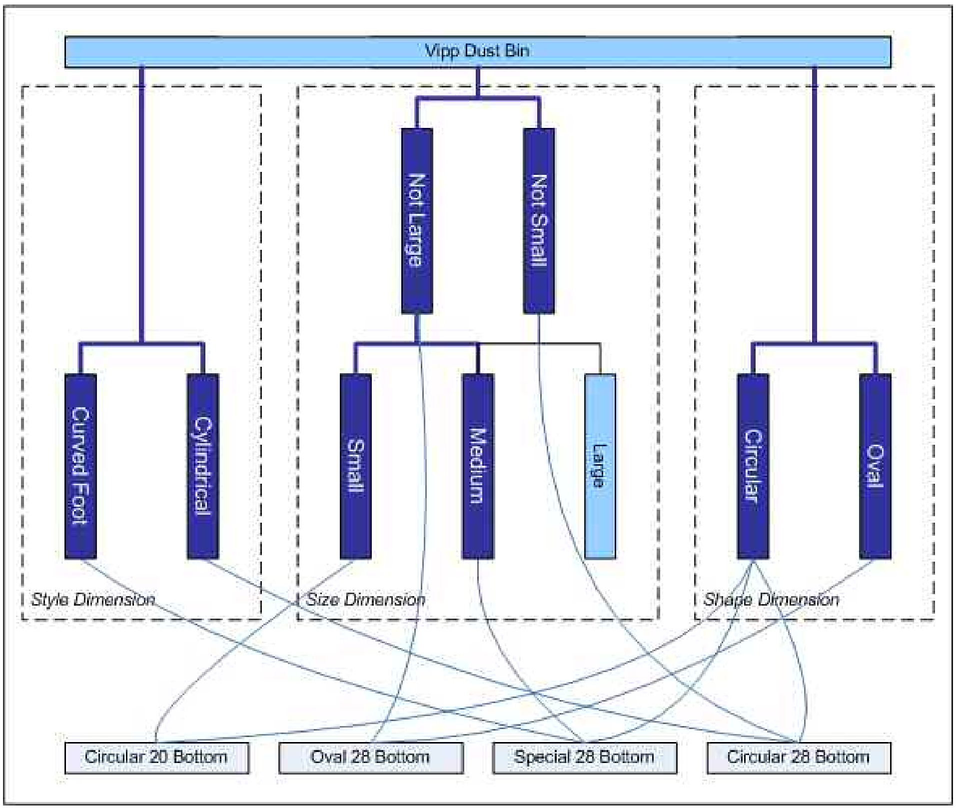

Figure 6. Mapping of the four different bottom components, which are needed in the Vipp dustbins. As can be seen from the diagram small circular bins have the 'Circular 20 Bottom' medium ‘curved foot’ bins have a ‘Special 28 Bottom’, and medium and large cylindrical circular bins have a 'Circular 28 Bottom'. Small and medium oval bins have a ‘Oval 28 Bottom’. Large oval bins do not have a bottom component assigned, but this is only because no large oval bins exist.

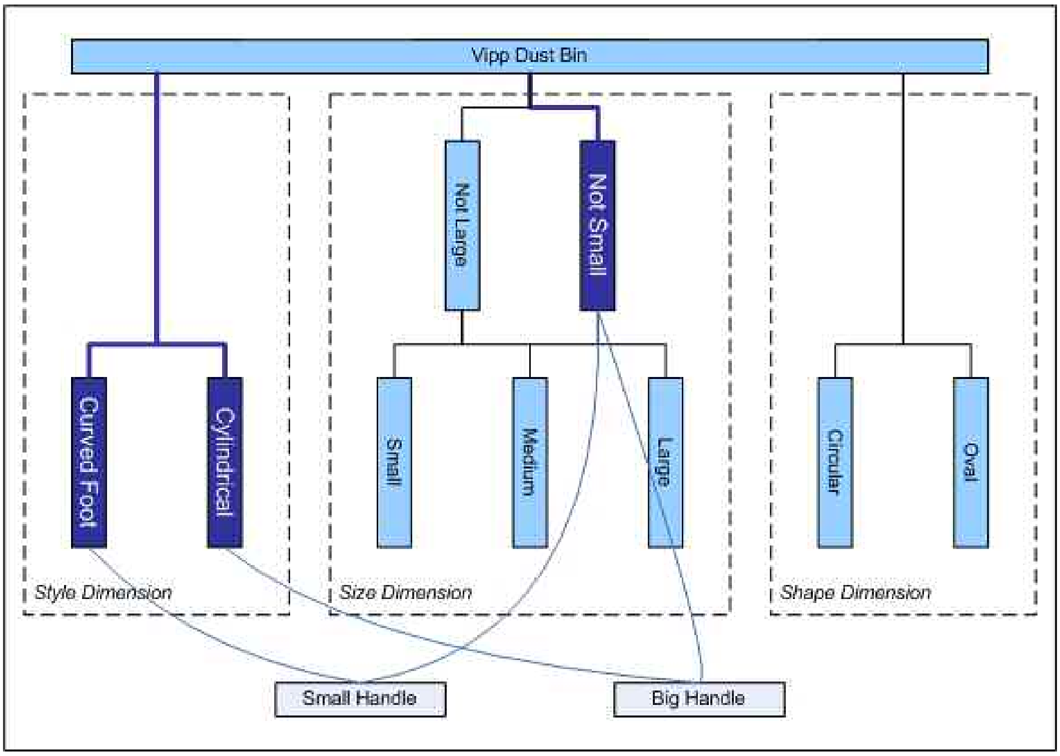

Figure 7. Mapping of the handle components in the PFH diagram. Small handles occur in 'Curved Foot' bins whereas big handles occur in 'Cylindrical' bins. No small bins have handles. The shape (i.e. oval or circular) has no influence on the choice of handles.

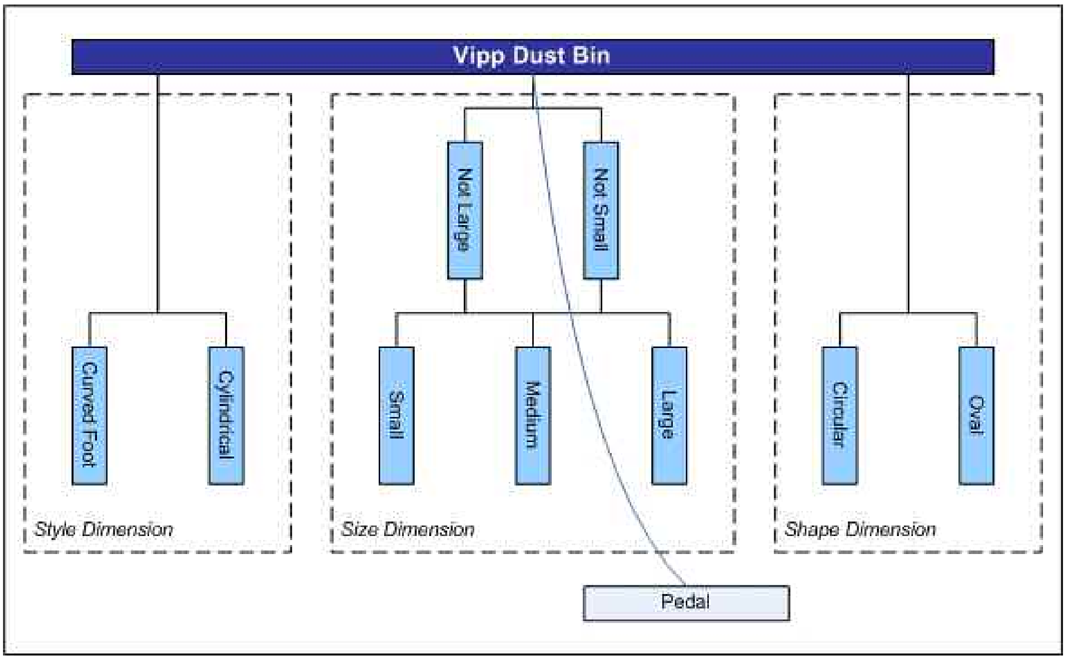

Figure 8. Mapping of the pedal component in the PFH diagram. The pedal occurs in all Vipp's dustbins therefore it is attached to the common generic product in the diagram.

The pedal and the handles are easily described in the Vipp PFH diagram, and this makes this kind of reuse easy to see-through. The bottoms however are an example of components that are reused in product variants, but with dependency of many dimensions of variance. This makes the representation of reuse look quite complex, and makes it more difficult to identify potential reuse that crosses the dimensions of variance.

It will be possible to describe several key components like the pedal or handles in the same diagram, but some components will make the diagram look very complex and not provide any overview. Hence it is recommended that each kind of component is described in its own diagram as it is seen in the example.

5.4 Improving key component structure and creating draft modules

When all necessary components have been mapped into PFH diagrams, these should be examined for similarities and other patterns. It could prove beneficial to move one or more components in the diagrams, so that they are reused through other generic products. Mostly this will involve using components on a higher level possibly replacing several actual components with one common one.

From the Vipp PFH diagram we identify two potential improvements for the dustbins:

- It should be examined if the handles need to vary with the style dimension.
- It should be examined if the ‘circular’ shaped ‘cylindrical’ and ‘curved foot’ dustbins could share the same bottom component. This would render the bottoms independent of the style dimension and reduce the number of alternative bottoms to three.

Draft modules for an architecture are established when the final attachment of key components has been reached. For each combination of generic products that has components attached a draft module is created. If several components share the same combination of generic products (i.e. occur in exactly the same products) these are integrated into the draft module. If only one component has a specific combination this becomes a draft module on its own.

If the draft module attached to the common generic product consists of no or only a few components and two or more independent subgroups of generic products can be identified then it may be an advantage to split up the architecture. The final modules will only relate to one of the architectures, and this creates a more transparent representation and architecture. This is not the issue in the Vipp example, but in the application case at Martin professional it was beneficial to split up the architecture into four separate architectures.

5.5 Procedure results and further module partitioning

The result of this procedure is the PFH diagram, the selection of products for the product family, and the mapping of the variance creating key components, which are combined into draft modules for a family architecture. These draft modules are by no means the final modules and are just a starting point (i.e. a specification of which key-components to include which modules) for the creation of the reference architecture. The final modules in family architectures cannot be justified from a marked or customer viewpoint alone. Upgrading, outsourcing, serviceability, assembly etc. might justify further partitioning into smaller modules. The procedure presented in the above offers no way to create these final modules. However the created draft modules form a valuable starting point for a modular structure and the dimensions of variance can be used as a fix point for the rest of the process of setting up a family architecture.

6 Application at Martin Professional

The procedure described in this article was tested on Martin's portfolio. This was done in cooperation with key-employees at Martin in a series of interviews and reviews. When milestone decisions or final evaluation of alternatives were reached a larger group of the product development and project manager staff and executives were consulted.

The outcome of applying the procedure at the product portfolio of Martin was the above mentioned resulting diagrams and a complete draft of a future product development platform in Martin, this draft was developed into a family architecture and a platform, which are now in the early implementation process.

This draft was received with great enthusiasm by company executives and project managers, as it gave them an overview of their planned products and their possible relations, which they had not had and had needed badly. Furthermore it acted as a basis for discussions of the final modular structure of the products. One project leader claimed that the overview could additionally be used in training sessions for new employees.

The procedure supported the selection of 35 products to be included in the architecture out of 39 considered. It excluded the more exotic products, which would be tempting to include, because the decisions could be based on systematic arguments and not just random and personal ideas and preferences. The representation gave overview and made discussion possible with product managers, engineers from different technical areas of the product development and the production.

Using this procedure caused the decision of making more than one architecture and the resulting diagrams were used in the rest of the process as well, both as check point material and to provide inspiration. This indicates that the resulting diagrams have advantages that go beyond the early phases of architecture and platform development.

7 Further research

There are three areas of further research: Appliance of other tools in combination with the procedure, reparing on the drawbacks of the procedure, and further development of the mindset and visualization methods that have been identified. The first and second area may be overlapping.

In the application case at Martin the procedure was used successfully in cooperation with tools like Product Family Master Plan [6] and Module drivers [2]. It may also prove useful with other tools.

Reuse is only shown indirectly in the PFH diagram, tools that visualize this like commonality plans may be a useful supplement. Furthermore it may be possible to set up an even more precise way to identify the dimensions of variance and the key components.

During the work with the procedure and the PFH diagram three aspects of improvement have appeared:

- The definition of a value is vague and could be related to well-defined terms, e.g. functionalities, properties, characteristics and features.
- Although variance dimension are treated alike in this paper a difference between primary and secondary dimensions of variance in sense of their importance and influence has been identified. This difference and its effects should be further investigated.
- A tool or method for measuring the effectiveness of a product family or a modular structure this early stage would be very useful.

8 Conclusion

A procedure of early stage investigation of different product families for architectures has been set up. The procedure provides identification of variance dimensions based on customer demands. The PFH diagram visualizes these dimensions, and makes it possible to argument rationally when it comes to limiting the product family. The procedure relates key components of the product solutions to the dimensions. From this it is possible to identify draft modules as a starting point for the final modular structure. In total the procedure offers a way to handle early decisions and aids the designer in making the right choices in the early platform synthesis.

The procedure has been successfully applied in industry and proved useful as decision base for key aspects in the development of an architecture and product platforms. Product developers and executives received it with great enthusiasm because of the overview, structure and clarification it brought to a traditionally loose and undocumented process.

References

[1] Dahmus, J., Gonzalez-Zugasti, J., and Otto, K., “Modular product architecture”, Elsevier Science Ltd, 2001.

[2] Erixon, G., “Modular Function Deployment – A Method for Product Modularization”, The Royal Institute of Technology, 1998.

[3] Hölttä, K., Tang, V., and Seering, W., “Modularizing Product Architectures Using Dendrograms”, International Conference on Engineering Design in Stockholm, Stockholm 2003.

[4] Hubka, V. and Eder, W.,”Theory of Technical Systems”, Springer-Verlag, New York 1984.

[5] Martin, M. and Ishii, K., “Design for variety: developing standardized and modularized product platform architectures”, Research in Engineering Design 13, Springer-Verlag 2002, S. 213-235.

[6] Mortensen, N., “Design Modelling in a Designer’s workbench – Contribution to a Design Language”, Ph.D. Thesis, Technical University of Denmark, 1999.

[7] Robertson, D. and Ulrich, K., “Planning for Product Platforms”, Sloan Management Review, Summer 1998, 1998, S. 19-31.

[8] Stone, R., Wood, K., and Crawford, R., “A heuristic method for identifying modules for product architectures”, Elsevier Science Ltd, 2000.

[9] Ulrich, K. and Eppinger, S., “Product Design and Development”, McGraw-Hill Higher Education, 2000.

[10] Van Wie, M., Rajan, P., Campbell, M., Stone, R., and Wood, K., “Representing Product Architecture”, Proceedings of DETC’03 ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference in Chicago, Chicago 2003.

[11] Zamirowski, E. and Otto, K., ”Identifying Product Portfolio Architecture Modularity Using Function and Variety Heuristics”, Proceeding of the 11th International Conference on Design Theory and Methodology 1999 ASME Design Engineering Technical Conferences in Las Vegas, Las Vegas 1999

Ole Fiil-Nielsen
Technical University of Denmark
Department of Mechanical Engineering – Engineering Design Section
Nils Koppels Allé – Bygn. 404, DK-2800 Kgs. Lyngby
Denmark
Phone: +45 45 25 56 41
Fax: +45 93 15 77
E-mail: ofn@mek.dtu.dk