NordDesign 2006
August 16 – 18, 2006
Reykjavik, Iceland

Dynamic Product Structure Configuration Specification Management

Niko Ville Salonen

Rolls-Royce Oy Ab
niko.salonen@rolls-royce.com

Introduction

This study uses a framework of a research program to understand product structure configuration management requirements based on findings over recent years in fourteen different commercial marine machinery product companies. The competition and innovations in the mature market require understanding products in depth, which require companies to evolve and enhance product data structuring capabilities in different product life-cycle data management systems. Capturing product structuring knowledge and re-using information more efficiently is more question of survival than creating competitive advantage.

Some of the requirements have already been identified in the current stage of the research program. One of these is flexible change management to enable dynamic changes of documents linked to parts during continues design changes and evolution. This requirement is one of the most important needs of structure configurations versus current industry de-facto description of document to part relation. The enabler of the flexible change management is form, fit and function based version management of all product structure objects.

This study specifies how dynamic configuration specification works as part of product structure options and variants definition. Study identifies what are requirements of dynamic configuration specification against change management and product life-cycle phases. It describes the approach where part is no more identified as static revision based definition as such, but references to more universal rule of form, fit and function definition and dynamic state as design always is.

The aim of the study is to identify issues, which should be part of the overall definition of dynamic product structure configuration model and what should discussed in appropriate academic, industry and standardization forums and changed to de-facto definition within commercial product data management systems.

Approach

The model used in this study is the four classes of product structures by Andreasen M.M, Hansen C.T & Mortensen N.H. Design life-cycle views [1] and product life-cycle phases are here defined as corresponding to the domain theory and the four classes of product structure model. Configuration management has to satisfy at least configuration management II (CMII) definitions by the Institute of Configuration Management where applicable. This study is based on case studies, benchmarking and implementation projects of 2D and 3D CAD tools and different commercial product data management (PDM) systems during the last five years. This study takes as its basis modular commercial marine products and overall situation in these companies.

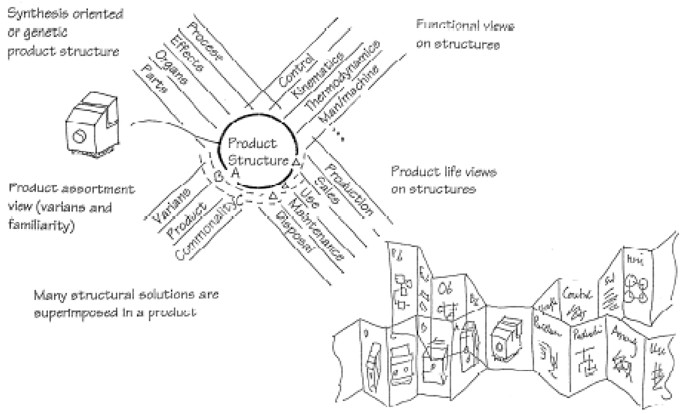

Figure 1. Four classes of product structure [1]

Genetic product structures

Generally industrial product structuring is based on part structures. Part structure represents assembly content-based relation structures. Relation objects are the glue of part structures. The most common part structure is based on assembly content structure also called as bill of materials (BOM). Assembly content part structure identifies parts and part quantities, effectivities of parts, different part baselines such as versions and states, occurrences of parts and possibly options and variants that are included in product structure definition. This study concentrates to options and variants, also called as superimposed part structures.

Industrial product structures include main life-cycle phase part structures, but so far many companies are concentrated to most essential structure definition using mainly manufacturing part structures especially for project-oriented products. Products that have long design phase or have complex product assortment use normally engineering part structures and manufacturing part structures. However, currently implemented product structure models do not define assembly connections between parts, assembly sequence information of parts or design content between parts. Structures do not usually support part structure specific relation documentation; this would require relation objects to support internal documentation. Process, functional and physical behaviour relations are also neglected. This means that current implemented industrial product data management systems do not support analysis of change impact.

Above described part structures are based on embodiment and detail design phases including the functional views, but there exists earlier phases and even further views of genetic product structures and life-cycle views as defined in study by Andreasen [1]. All genetic part structures and life-cycle views can be captured, viewed and configured to next life-cycle phase by dynamic product structure configuration specification, which basics are introduced in the next chapter.

Some of these views and relations between parts can be build comprehensively in current data management systems, but usually complexity of creating, representing and understanding of numerous parallel, consecutive and looped connections become a too complicated environment for engineers. Engineers are not able to work efficiently, because commercial systems do not support visualization and advice to understand the impact of change in systematic way, defined by simple and powerful principal life-cycle rules. This means that product data management systems are customized to support complex reporting to capture needed connections.

These problems create new needs for relation data model objects and systems supporting the use of this data. The model and tool together should include visual viewing tools, flexible change and configuration management. Configuration management should be managed by a set of simple engineering based rules. These would enable different behaviour definitions in design and product life-cycle phases and support engineers in understanding change impact.

Dynamic product structure configuration specification management

Dynamic configuration specification approach takes advantage of dynamic, systematic, comprehensive and concurrent life-cycle status management over product structuring. Dynamic means the principle thought that product structures evolve continuously in time, which requires simple and systematic definitions to be used for engineering staff that do not have or need deeper understanding of data models, database structures and class hierarchies.

Dynamic configuration specification identifies universal rule defined in earlier research study by Salonen [4], which applies same configuration specification versioning rules as to part structures, also to part internal definition structure as defined normally at least by documents. This rule defines same form, fit and function rule to all product structure objects, which influences to internal version management of a part object. Earlier study concentrated to part structuring, but this study includes more comprehensive view to options and variants structure configuration management.

Dynamic configuration specification is a viewing tool to product structure that filters and re-arranges product structure based on configured specification. Configuration specification is currently understood to carry structural filters that set structures to specific state, baseline or effectivity type views. The main area of dynamic configuration specification functionality is in variants and options filtering. Genetic structure definition is already defined by different studies. Behaviour modelling itself is till in an immature state of evolution to understand all filtering requirements. There is also need to enhance baseline and effectivity management, which this study takes a closer look. Dynamic configuration specifies also Boolean views, which can combine earlier mentioned views in a sequence.

Variants and Options view

Variants and options filtering identify certain types of option and variant features. These features are related to filtering and manipulating all product structure objects. Processes, functions, organs, parts, CAD documents, generic documents and document content, especially CAD files. The difference between variant and option is defined as: “variant is mandatory feature of structure” and “option is optional feature of structure”. Option can be treated as variant including null variant when option is not configured. Variants and options are structural features but are also internal features of structure objects as parts and documents are.

Most important system definition requirement is to understand that variants and options is part of design process and should be generated during design process as part of product structuring process from CAD design to BOM creation. Currently these features are generated in commercial PDM systems with different tools or tools that support more the selling perspective of variants and options. Engineering configuration and sales configuration are different things and have different rules to apply and also behave in different ways. By separating the engineering variants and options configuration and sales configuration, more simple sales configurator systems can be built.

During the study three (3) different variant types were recognized to support different product structuring requirements in modular products.

Static variants

Static variant is data definition of the relation object. It can be part of part structure relations or part and document relations or even document and content file relations or it can be mixture of these structure relation object definitions. It is information of structure, not part of data model of principal objects (part, document, file) themselves. Static variant can use any known algorithm to specify is configuration specification. Most used are predefined values or codes, mathematical formulas or Boolean rules.

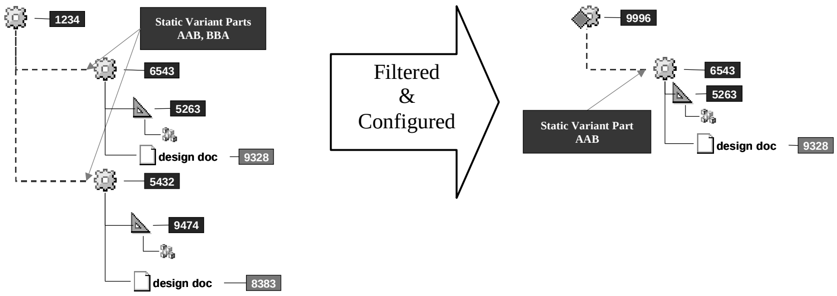

Figure 2. Static variant filter configuration

Dynamic variants

Dynamic variant is data of the principal object. There are algorithms that define the behaviour of the object. Algorithms define the rule of configuration, how object changes, but also how object behaves in the engineering change process flow. Objects can be influenced by algorithm driven through parent to child relations shown in figures 3 and 4. Dynamic variants use more efficiently the universal version management definition to generate unique product structures from generic to specific, because relation object models require consistent data uniqueness. This is especially important for different documents as shown in figure 4. Certain rules can be defined which product structure objects are set dynamically to change when design is changed or configured from state to an other, but there are engineering best practices that influence to these rules, especially documentation of generic design documentation use practices that reduce documentation dramatically. When talking about creation of new objects by configuration one of the key indicators is how many parts and documents are maintained and what are just stored as historical data. This is very important factor to be measured against when building a new product structure model for life-cycle phases.

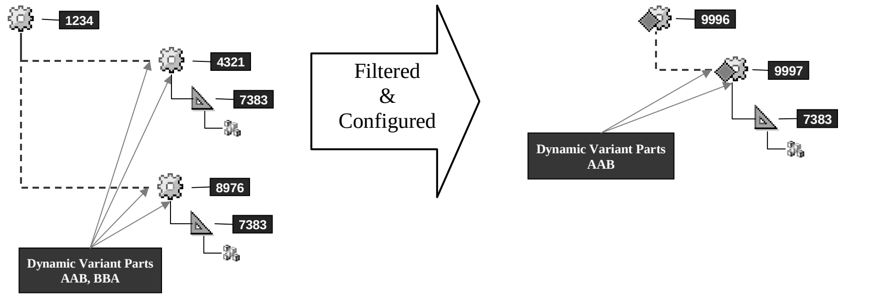

Figure 3. Dynamic part variant filter configuration

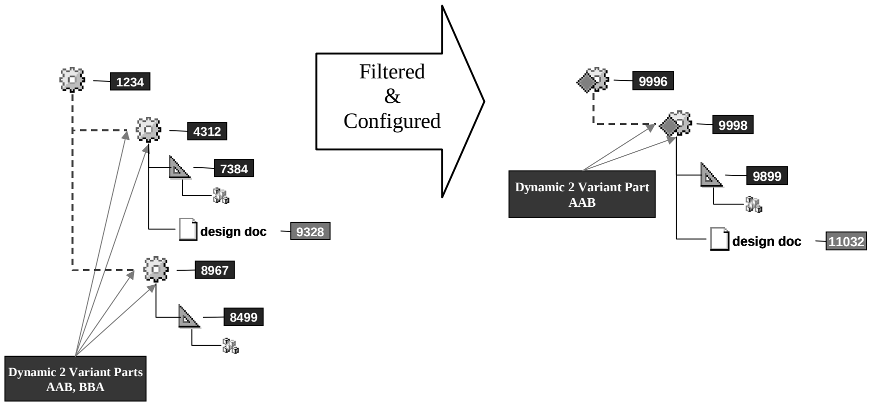

Figure 4. Dynamic part and document(s) variant filter configuration

Library selected variants

Library selected variant is a sub feature of dynamic variant but it refers to unknown rules that design intent is not able to predict. Library selected variant in engineering product structure is a phantom in procurement means. It is a place holder of principal object that is replaced by real one when configured to next life-cycle state. Library name refers to a library of objects that represent a large group of possible solutions as shown in figure 5 or related to a process, which can create a solution option.

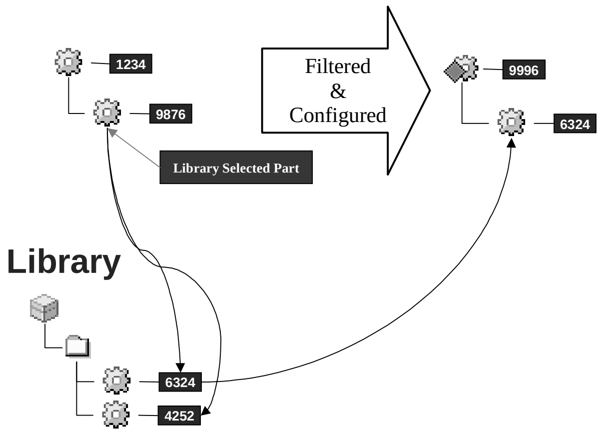

Figure 5. Library selected variant filter configuration

Genetic structure view
This study does not go into detail definition of genetic structure filtering, because this area has already being well defined by several researches based on Systems Theory by Hubka [3]. Genetic structure filtering identifies views between different design evolution states process, function, organ and part structures are identified in the Chromosome Model by Andreasen [2].

Behaviour model view
Behaviour modelling is large area of design analysis. One area is documentation related to analysing design, which is currently done as part of product structuring process. Other area is advanced computer based design analysis including structural strength, kinematics, thermodynamics, hydrodynamics and aerodynamics or other physical theories that are modelled to support design. Third area is control design and man/machine interaction analysis. All these areas are tightly linked in real life.

Currently behaviour analysis is done as separate engineering tasks by specialists and most likely including less documentation than to understand the actual impact to design. Study showed that there is no actual product structure linking done currently to relate behaviour and product structure objects. Behaviour modelling filter should tight up different behaviour models and give engineers feedback what phenomena interact with product structure objects. This could be then extended to re-usable behaviour model structures that would enable change management to a new level of understanding change impact.

This study found behaviour model filtering extremely complex case, because behaviour is consequence of designed solution, not so much re-usable structure of design currently. There exist some simply behaviour model examples, spring or bolt connections that can be applied to support design, but these are done inside CAD tools or with customized tools. This type of design behaviour modelling automation is currently part of Knowledge Based Engineering (KBE) research by many CAD system providers, universities and institutes. When this area continues to evolve there should be universal rules how behaviour models are connected to other product structure objects. Principle rule of filtering should be based on different connection and behaviour types. Behaviour connections in CAD design area where identified in the study by Salonen [4].

Baseline view
Baseline is a stored snap shot of product structure. Baseline requires a configuration specification to be set before baseline is stored, so it is a Boolean type filter. However a variants and options filtering can also be set to already defined baseline which is again a Boolean of existing Boolean filter. Baselines merge the life-cycle phases to systematic way of managing life-cycle stages, design process phases, stages and gates as shown in figure 6.

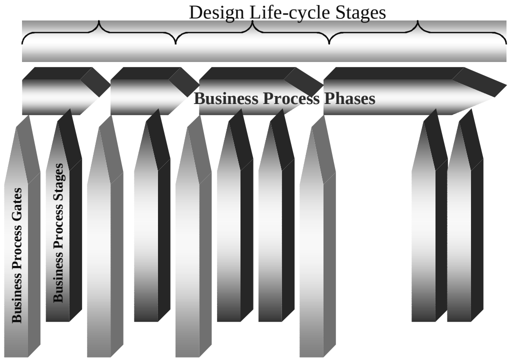

Figure 6. Baseline stages, phases and gates

Baseline is used to support systematically product life-cycle management. This study found it very flexible for different type of products and different life-cycle requirements. Especially change management perspective can be joined to baseline definition seamlessly. PDM systems still do not have baseline objects supported as well as part objects. Especially change management requirements including version management, creation and change support features are not currently supported.

Effectivity view filtering

Effectivity is not described as a class of product structure, but it is raised to more important feature in this study, which joins some of the rules of product structure life-cycle views. Effectivity is a rule against product structure object specifying when specific object version becomes effective to be used in product structure. Effectivity is a business rule based filtering feature. Effectivity is understood currently to date type effectivity, but there are other types of effectivities. There are combination effectivities that specify object relation effectivities. There is logistical effectivity, which emphasises is product structure object ready to be configured to next life-cycle phases. There are also disposition effectivities that define old version usage of the object.

Conclusions

Dynamic configuration specification is viewing tool of product structure. The configuration management of product structure should include all product structure views as described in the study approach model. This study showed how dynamic configuration specification works with variants and options. The study identified change management requirements and issues to handle product structures through life-cycle phases in a systematic way.

The research program is continuing by identifying product structure cases which dynamic configuration specification management can have an effect on. Tests will be conducted to determine how dynamic configuration specification will change requirements of genetic product structure modelling, baselines and effectivities. The model for dynamic configuration specification is still under development process.

References

[1] Andreasen, M.M., Hansen, C.T. & Mortensen, N.H.: The structuring of Products and Product Programmes, Proceeding of the 2'nd WDK Workshop on Product Structuring, June 3-4 1996, Delft University of Technology, Delft, The Netherlands, 1996.

[2] Andreasen, M.M.: Machine Design Methods Based on a Systematic Approach – Contribution to a Design Theory, Diss., [In Danish], Department of Machine Design, Lund Institute of Technology, Sweden, 1980.

[3] Hubka, V. & Eder, W.E.: Theory of Technical Systems, Springer-Verlag. Berlin, 1988.

[4] Salonen, N.V.: Complexity of product structure configurations, Proceeding of the Design 2006, May 15-18, Dubrovnik, Croatia, 2006.