INTERNATIONAL DESIGN CONFERENCE - DESIGN 2006
Dubrovnik - Croatia, May 15 - 18, 2006.

DESIGN 2006

COMPLEXITY OF PRODUCT STRUCTURE CONFIGURATIONS

N. V. Salonen

Keywords: product structure, configuration management and flexible change management

1. Introduction

This study sets a framework for a research program to understand product structuring requirements based on findings over recent years in fourteen different commercial marine machinery product companies. The study prepares an outline for definition of flexible change management for parts and documents in product life-cycle phases. The study presents computer aided design views by introducing different design relations between CAD document and file configurations.
Some of the requirements have already been identified in the current stage of the study. One of these is flexible change management to enable dynamic changes to documents linked to parts. This requirement is one of the most important needs of structure configurations versus current industry de-facto description of document to part relation. This study specifies this issue and firstly sets required change to enable flexible change management. The study concentrates on CAD documents, which are identified in this industry sector as the principal document of defining parts. The aim of the study is to identify issues, which should be discussed in appropriate academic, industry and standardization forums and changed to de-facto definition within product data management systems.

2. Approach

This study reviews three product structure models and compares industrial cases against them. Models are the domain theory by Andreasen M.M, the chromosome model by Mortensen N.H and the four classes of product structures by Andreasen M.M, Hansen C.T. & Mortensen N.H.
The chromosome model represents both constitutive and behavioural models for product structuring taking into account different classes of product structures. The chromosome model defines the map of progressive design identifying the requirement structures evolving to technological design domain and further to actual result of behaviour domain. The chromosome model includes the domain theory and technical system theories, which are not applied systematically in companies to consistently manage design intent and history of design evolution.
The domain theory is recognized to find the nature of causal design characteristics or as introduced in this study under name design life-cycle views. Four classes of product structures tell about real complexity of engineering design and how many levels, views and in-time depth it can have. This study uses these models and theories as the basis for more detailed benchmarking against change management issues found in industry document management cases.
Design life-cycle views and product life-cycle phases are here defined as corresponding to the domain theory and the four classes of product structures 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

3D CAD tool and different commercial product data management (PDM) systems during the last five years. This study takes as its basis modular commercial marine products.

3. Product structuring

Product structure can be defined with a model, which includes at least three objects. Parent part, child part and relation object between these two objects. This is a normal relation database definition of a product structure. There are also other definitions in use, but this framework concentrates on the relation object data models.
Product structure identifies the documents linked to part or relation objects. This study takes an approach to current product data management practices in which the documents are part of the same data model.

3.1 Part definition

Part as described in figure 1 has at least a master notation of identifying unique number and name. Part can have other metadata such as revision, date of creation, status, system architectural attributes and metadata, which defines the nature of the object. The most important task for engineering is definition of form, fit and function (FFF) for the design elements. This definition is not only for part itself but also for the child relation objects the part is linked with to other elements.

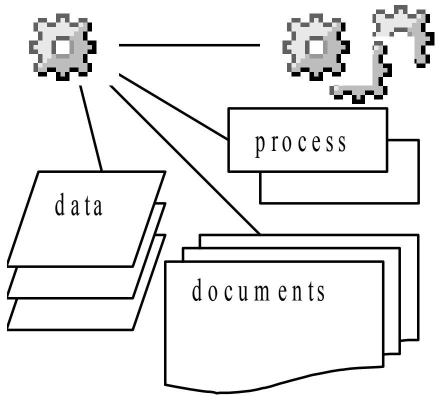

Figure 1. Part definition

The definition of form, fit and function used in the industry is necessity for revision management. A change of revision does not comply unless it is evaluated against the FFF rule. The FFF rule converts the change of part-to-part definition by the fact that adding new part(s), replacing part(s) or deleting part(s) is not possible without revision change of the parent part. A part revision can still include changes, which take place in revision level of its child parts. An engineering task is to decide whether parent part should be revised or not based on the FFF rule. To understand whether a parent part should be revised or not, the rules of function, fit and form have to be defined. Those definitions are set out below.

- Function property includes man-machine interaction and technical behaviour in designed conditions. If the original definition of functionality changes, it cannot be revised and is therefore a new part. Function is the main requirement of the FFF rule.
- Fit property defines how a part interfaces other parts. Later revisions of a part must fit to all earlier parent parts (part 001 revision A uses part 111 revision B and part 111 revision C has to fit part 001 revision A). Fit is the second main requirement of the FFF rule.
- Form is the child property of function and fit properties. It defines the geometrical, physical and chemical form of the part during life-cycle processes.

Form has to comply with both primary properties before form is considered. Then it can be treated as its own property. Form property has external property factor which is the documented (especially tolerances) and visible form of part. Form also includes microscopic internal form of the part, which normally identifies the physical and the chemical properties of the part. Sometimes external property is divided into separate part definitions of the part, which then changes management of form property. If these properties however are defined in one part then both of these have to comply to defined life-cycle processes with changes that can be handled within revision policy. Both of these properties can also be called “form to fit functions of life-cycle process phases”.
An example of when form becomes unrevisable in this context is:

> A part form change is minor – it has no effect on function or fit. The machining of part can be handled with revision change to CNC code. It still fits the box in which it is to be transported. Weight is not changed so much that it requires different lifting equipment. It can also be assembled with the same tools as previous revision of the part. In as-maintained service life-cycle phase a service engineer requires totally different hand tools to disassemble part from the assembly – which means that form change does not comply with all life-cycle phases form to fit rule – therefore part cannot be revised.

This is a strict interpretation of form with respect to life-cycle phases. Life-cycle management of parts is required and this is normally not fully defined in current systems, which means that ignoring life-cycle phase processes can create problems to revised parts.
Like the domain theory, the chromosome model and the four classes of product structure can, and should, contain other views of product structures. These should consider different views of technologies, product use, organisations, functions, organs, processes and actual information models of design content for part definition. However, these theories and models do not show the real life complexity of how to manage change of this data through life-cycle.

3.1.1 Part structures

Generally, part structure represents relation structure that identifies assembly content-based relation structure. Relation object is the glue of product structures. It makes part structures rigid or flexible depending on the solution architecture of the data management system. As discussed earlier the most common part structure is based on assembly content structure also called as bill of materials (BOM). However this structure does not define assembly connections between parts, assembly sequence information of parts or design content between parts. Design content’s primary carrier is CAD document as defined in chapter 3.2.1. Bill of material identifies parts and part quantities, options and variants that are included in the assembly. Functional and physical behaviour relations, which could be called “as virtual reality relations”, should also be described between parts and found in product data management systems to support analysis of impact of change.
Some of these relations can be defined comprehensively in current systems, but usually complexity of creating, representing and understanding of numerous parallel, consecutive and looped connections become a too complicated environment so that engineers are not able to work efficiently because coThese problems create a need that relation objects should include flexible configuration to manage a set of attributes and rules. These enable different behaviour definitions in design and product life-cycle phases and support engineers in understanding change impact.

3.2 CAD file structures

CAD files are identified in this study to include all different formats that yield information from design geometry to visual format - from scanned sketched paper to high-end 3D CAD models. A CAD file structure is more the basic feature of 3D CAD systems in figure 2. 2D CAD systems less often use CAD file structures. Some exceptions exist especially in electrical and control design systems, where intelligent objects are used to create relations and structures. In addition, embedded files are sometimes used in CAD files.

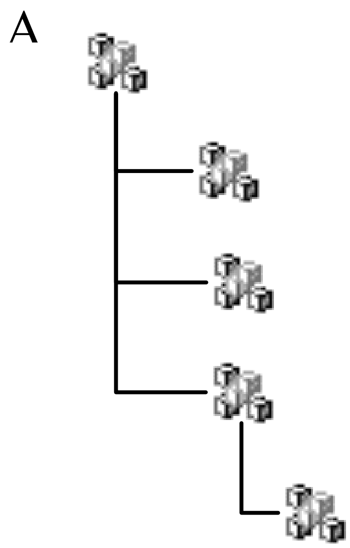

Figure 2. CAD file structure

3*.2.1 CAD file relations*

Today 3D CAD file structure models have more complex relation models than traditional part structure models have. A CAD file structure defines assembly content like part structure, but can also include assembly connection relations. Many CAD systems cannot identify real physical connections between CAD models, which makes realistic virtual definition almost impossible. Many CAD systems can model and represent assembly sequences, different positions, and orientation views in an assembly context. Some high-end CAD systems can also describe behaviour models, including kinematics and dynamics by integrating CAE (FEM, CFD) models to CAD model structures.

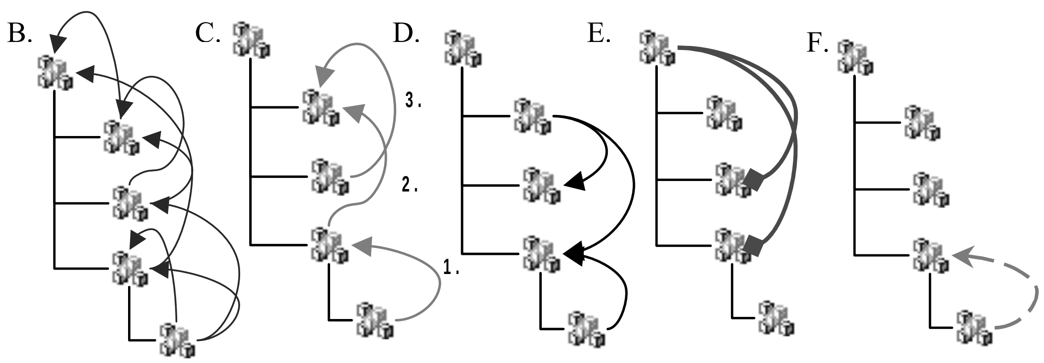

Figure 3. CAD file relations

CAD file structure relation types are shown in figure 2 and 3:
A. content relations (figure 2)
B. mating relations
C. sequence relations
D. geometry definition relations
E. attribute and expression definition relations
F. behaviour model relations
G. mixture of A, B, C, D, E and F

3.3 CAD document structures

CAD document contains metadata and is the carrier of CAD files. CAD documents are generated between CAD files and parts. Some systems allow direct reference from CAD files to parts but this is not the usual case. CAD document can contain one or many CAD files. A CAD document can also include representative files. These files are called as attachment files that are driven from actual CAD files or support CAD file information model definition. Usually one CAD file is related as the master for one CAD document, but the same CAD file can be related to several CAD documents by slave relation. CAD documents can be related to one or many parts. This is why CAD documents should be separated from part definition.

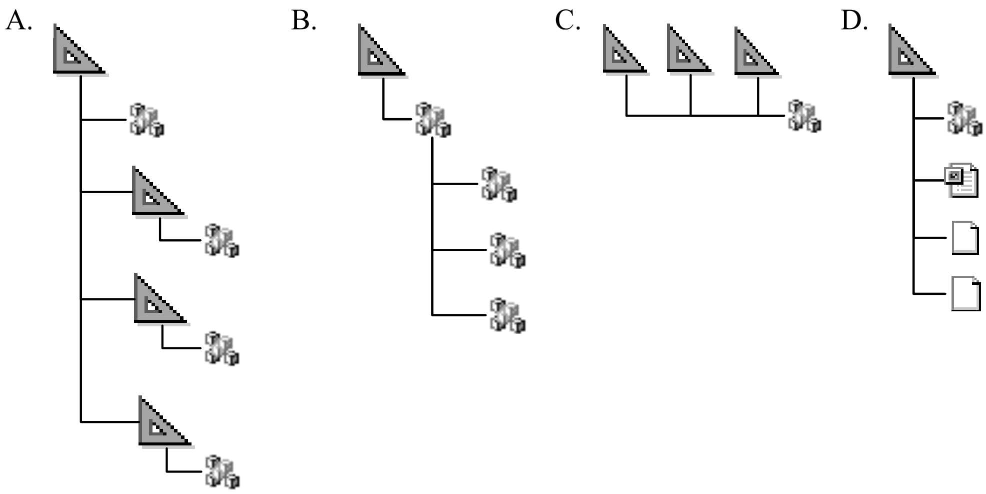

Figure 4. Different CAD document definitions

CAD document structure types as shown in figure 4:

A. one CAD document to one CAD file type relation
B. one CAD document to package of CAD files type relation
C. many CAD documents to one CAD file type relation. This requires primary CAD document definition
D. one CAD document to many different CAD file types relation. Same feature is used for one CAD document to many same type of CAD files when different representations is needed
E. mixture of A, B, C and D

3.3.1 CAD document relations

The CAD document relations primary purpose is to manage file structure. A secondary purpose also important is managing CAD file internal relation information as described in chapter 3.2.1 in figure 3. These features however are still in an immature state in many product data management (PDM) systems.

3.4 Configuration management of documents

Configuration management has been part of part structure functionality in many product data management systems for many years. Especially baseline and configuration types that relate to effectivity date, status, time or status factor of parts, have been the essence of configuration management in product data management systems. Variants and options have been the latest introduction to commercial product data management systems. A baseline is essentially a snapshot of a collection of product data at a specific moment in time. Once a baseline is created, parts and documents can be added to the baseline. A baseline can contain any number of parts and documents, in addition a part or document can be in any number of baselines. Effectivity date is a time period where a part is introduced as effective to use.

This study creates a new requirement for configuration management and specially revision management of documents, which is a part of status configuration definition. Revision status definition has at least two states, work-in-progress and released states. There can be many other options which companies use, such as preliminary, pre-released, cancelled and in-review states. Status defines what is the current phase of objects in the design process. Revision status can also hold identification of major and minor revisions.

As discussed earlier, the revision management of part structures is based on the form, fit and function rule. As known, changes to detail data of part occur dynamically all the time. Small changes and corrections to documents are needed without interfering with the FFF definition, especially in 3D CAD systems where assembly is updated automatically when child parts are changed. This is one of the disadvantages of 3D CAD tools, where the system do not understand engineering rules like form, fit and function, because it is not mathematically clearly defined. However, this problem can be solved with configuration management for documents.

In industry, the most widely used de-facto standard in commercial PDM systems follows the definition that part is the primary object of design, where internal definition, metadata and linked documents are under parts revision control. Documents related to part can be revised to current part revision only when part is not in released state. This means that modifications done to documents do not have the same rule as the FFF rule. They are not self-contained against the FFF rule. The document definition is frozen to specific part revision unless the company has identified this as the bottle-neck of design changes and has customized or configured their PDM systems to work in a more harmonized way using the FFF rule across product structure definition.

The above argument that standard commercial PDM systems do not follow the FFF rule fully is actually based on historical consequence of data management. Enterprise resource planning (ERP) systems were the data management systems before the PDM era. ERP systems used item naming instead of part. Items were the carriers of data in those systems and still are today. Items carry the cost values and logistical data, which is relevant for business to freeze the definition to item level. However, a major part of this definition is done in PDM systems by documents defining different form, fit and function properties. This issue makes implementing document configuration management hard for industry, because these systems have to talk to each other.

Configuration management of documents should apply the same definition as part structures have to simplify interpretation of the FFF rules in design. It would harmonize system behaviour and increase the power of configuration management to include documents into it. In this scope documents should have at least the baseline, effectivity date and version status in which configuration rules are applied.

4. Change management

Product structuring is a set of engineering activities done in parallel and in sequence. These activities are actually based on change management. Change management includes main activities such as create, update and delete. Change management however is, more about understanding reasons why to change and how to change, by analysing impacts of change. Configuration management is part of this change if configurations are stored as baselines or views and used for change to create new structures or are exported to other systems to be used in these configurations. In this perspective, configuration management should be part of change management.

Change management is the key element of product structuring. As described earlier, change management should include all information about objects themselves, but above all information about configurations of objects, which means managing the relations of objects. This means that requirements of change management features become complex, when all views of product structures and life-cycle phases are managed.

Change management is commonly used across the industry sector, following the Configuration Management II (CMII) principles [Institute of Configuration Management 1988, Guess 2002]. This is of course not the case in all companies and especially not in small or medium size companies. It is the most common near de-facto standard of change management.

4.1 Flexible change management

CMII or company specific change management processes do not meet flexibility and efficiency requirements unless change process is adapted to their business needs. CMII however is capable of introducing basic requirements and main process flows that immediate increase quality of engineering change management processes. The area of change management considered in this study is document version management.

4.1.1 Revision and iteration management – version management

Version management is found to introduce severe undesirable impacts unless system architecture does not support engineering change definitions. Form, fit and function rule is one of most important rules and should be defined clearly for the engineering processes. Unless change management is not understood, no tool can provide security against errors.

Parts and CAD documents version management is laborious, because 3D CAD systems change is not based on engineering rules. The rules of IT systems architecture are sensitive to any change in objects, which can lead to escalation of change from child to parent without any real engineering need for it. This behaviour creates extra need for flexibility for change management process architecture. To accommodate secure change and it also effects configuration rule definitions. This requirement is not understood by tool providers.

5. Conclusions

Product structure change management can have an holistic view, but the reality of complex product information models require flexible change management process architecture and tools which support evaluation of impacts. This is not the situation with the current commercial systems, and this study introduces aspects for tool providers to emphasise configuration management in all product structuring.

The models identified in chapter 3 can be achieved in large engineering organisations but it still requires extensive training of engineers to understand change and impacts of change. The study also shows that products using these kinds of models are not possible to structure and manage without supporting IT tools. Simple models are more achievable in this state and the sum of efficiency and quality is more easily gained from them.

The configuration management of documents introduced in chapter 4 is the outcome of this specific study. Identified flexible change management requires that objects are self-contained against the form, fit and function rule throughout the product definition to enable dynamic changes. The outcome of this study relates to other product life-cycle phases, which require other such as systems like enterprise resource planning systems to recognize change of document such as parts. This however is not true today in current systems.

The research program is continuing by identifying part structuring cases which document configuration management can have an effect on. Tests will be conducted to determine how document configuration would change requirements of baselines and effectivity dates. Tests will identify how other product life-cycle phases are affected when documents have same functionality of configuration management as parts have.

References

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.

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.

Guess V.V.: CMII for Business Process Infrastructure, ISBN 0-9720582-0-6, Holly Publishing – Scottsdale, Arizona – United States 2002.

Mortensen, N.H.: Function concepts for machine parts – contribution to a theory of detailed design, Proceedings of ICED 99, Munich, TU-Munich 1999.

Mr Niko Ville Salonen[1, 2]
[1]Systems Manager / Product Management
[1]Rolls-Royce Oy Ab
[2]Post-graduate student
[2]Tampere University of Technology (TUT)
Po Box 220, Suojantie 5, Rauma, Finland
Tel.: +358 2 8379 4911
Fax.: +358 2 8379 4736
Email: niko.salonen@rolls-royce.com, niko.salonen@tut.fi
URL: http://www.rolls-royce.com/marine