INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 03 STOCKHOLM, AUGUST 19-21, 2003

APPLICATION SCENARIOS FOR CLASS-SYSTEMS IN PRODUCT LIFECYCLE MANAGEMENT

Werner Puri
Stefan Sander
Harald Meerkamm

Abstract

The philosophy and scope of classical Product Data Management (PDM) systems have changed strongly in recent years. The first systems were appropriate for drawing administration only, the enhancement with functions for parts list (Bill of Material - BOM) management, product structuring and classification of components were added in the course of time. A further development step was the support of processes and workflows such as release and change management. However the focus of all these extensions was the technical order transaction. The next development step within the PDM area was Collaborative Product Definition Management (cPDM), which places to the fore the process chain and the interplant of b2b co-operation. The transition from classical Product Data Management (PDM) to Product Lifecycle Management (PLM) asks for enhanced functionality of the employed IT systems as well as data models. Following the development stated above, the bottom line conclusion is: A standardized format for component data is needed in order to ensure smooth, efficient and economic data exchange between cooperating enterprises. The advantages of organizing data with class systems are given evidence on the basis of examples from different product life cycle phases.

Keywords: product data management, functional modelling, classification, life cycle

1. Management of component data in product development

Data management is determined nowadays in most enterprises by *numbering systems and master data records*. The description (specification) depth achieved with these methods is not fulfilling the requests of PLM. Take for example the support of the flexible configuration rules or prescriptions within complex products. A *attribute/characteristic*-based approach for component management is indispensable. Following this practicable scheme leads to efficient hierarchically structured and interlaced class systems.

A class system combines the benefits of a hierarchical classification system and the describing features of a class list of characteristics. Similar objects with the same describing features that only distinguish themselves by different property values are bundled into classes. These classes are described by those properties and structured hierarchically. The parent-child relationship within the hierarchy can be semantically described as a “is_a”-relationship. Every class defines a space within which the properties for the description of objects are defined uniquely. This means that the semantic meaning of a specific property is the same for all objects within the same class and its subclasses. This is not the case for class lists of characteristics. A class system follows the paradigm of object orientation, meaning that all parameters of a parent class are inherited in all child classes. Classes that have other child classes of their own account are called generic classes.

These generic classes can contain generic objects respectively parts. In this case, these parts or products are just an abstraction of real parts respectively products. These generic parts are useful in case of a methodical design approach in which a designer defines the use of a generic part "rolling bearing" within a bill of material at an early stage of the design when no concrete part can be specified. This generic "rolling bearing" can then be replaced by the designer with a concrete bearing (e.g. drawn cup needle roller bearing or cylindrical roller bearing) at a later point in time depending on the individual requirements of the current design.

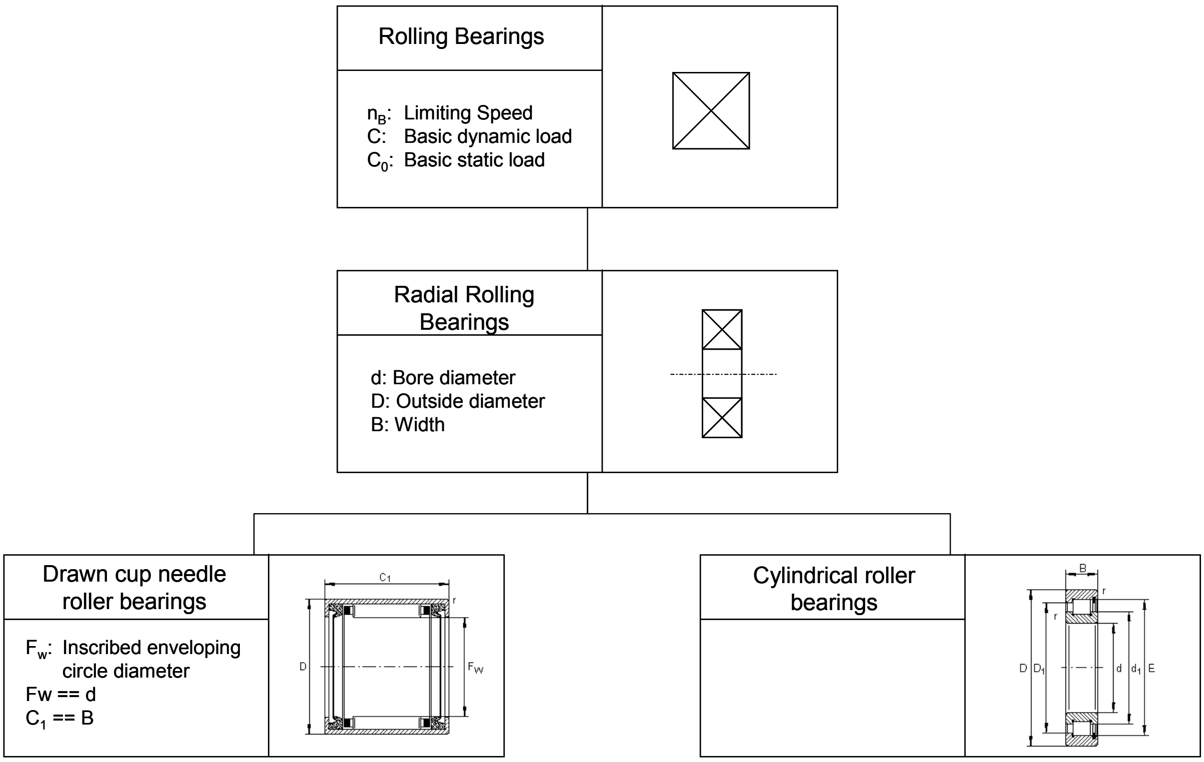


Figure 1: Example of a class system

The usage of a class system for the management of data offers a lot of benefits within the product development and the total product lifecycle. The hierarchical structure between the classes offers a better overview and supplies some order to the contained data. This leads to a faster and more methodical search and enables the inheritance of properties. Inheritance provides a means of uniquely applying properties within a hierarchy. It saves time during the definition of the properties since not all the properties have to be defined for every class and it avoids redundancies. Another benefit of the inheritance of properties is the improved property-based search capability within the hierarchy.

The ISO 13584 PLIB (parts library) is a series of International Standards for the computer sensible representation and exchange of part library data [3]. The objective is to provide a mechanism capable of transferring parts library data, independent of any application which is using a parts library data system. The nature of this description makes it suitable not only for the exchange of files containing parts, but also as a basis for implementing and sharing databases of parts library data. Part 10 of ISO 13584 [2] stipulates the creation of class systems that meet all demands of PLM. It strikes us as reasonable to master component administration (management) based on class systems consistent with the ISO 13584.

2. Application scenarios for class-system in product lifecycle management

2.1 Supporting the early design phases

Methodical design is easing the design within the phases but at the same time it creates problems at the transition between phases. Stepping to another phase is in the opinion of Roth [8] at the choice of the designer. The early phases of product design are characterized by the modeling with partial mostly qualitative information. Elements of the conceptual phase are gradually enhanced and completed with quantitative data to permit the evaluation and choice of solutions [7]. It is impossible to make accurate predictions on costs, manufacturing or environmental impact in the planning and concept phase of product development, even if forecasts in this phase would open possibilities for great improvement. O'Shea [5] is proposing an enhancement of product models as a solution to this problem. The most experts in the field of expertise are speaking about a integrative, phase-overlapping product model as a prerequisite for PLM.

The early phases have a great impact on quality and profitableness of the designed artifacts.. Even if the theoretical base of the methodical design was founded 30 years ago, there is no extensive usage outside the science world. Several reason for not employing the method are counted: to universal, no software support, not known, unwieldy etc.. This is the starting point of today's research: geometry and functions in CAD, standardized schema for representing functions, unified vocabulary for naming functions of mechanical devices [10]. In our eyes one of the most important task is the formal description of functions. What and what for?:

- Unequivocal terminology
- Reduction of misunderstanding, ease of communication of functions during modeling
- Improvement of expressiveness and compare ness of functional models
- Systematical search after analogies in existing functional models for new designs as well as reengineering
- Better software [7], [8], [10]

The need for a taxonomy of mechanical engineering functions is known for several years. The standard literature shows quite a bit of approaches [1], [7]. Actual research tries to combine the existing based on similarities [6]. The authors propose a function library for supporting the product life cycle management. This library should build on the ISO 13584 – Parts Library standard. Comparing parts is possible only through relating to a reference library. In analogy to the ISO 13584, a functional reference library would provide a way to relate parts and products based on their main function. The domain specific description of functions could be translated and used in other domains.

Relating functions and sub functions to the structure ("Baustruktur") of existing products should be possible without great methodical problems. The functional description would open new ways of cooperation for component suppliers in the early phases of product development. Following tasks and problems arise from this approach:

- a clear without ambiguity correlation between parts, assemblies and functions is not always possible, partly due to the discretisation of the function structure

- the consistent formulation of technical functions is not given, need for a reference hierarchy
- formulation of function integration.

Operation an flow formulations after Stone [10] could be used as the starting point in building up a reference hierarchy for engineering functions. The benefits generated by using a linked function - part library, implicitly connecting the function structure to the product model are:

- ease of product maintenance,
- ease of reengineering
- ease of search for equivalent solutions [9]
- ease of optimization etc.

Fig. 2 shows the possible linking of product data with catalog data. By creating the relation between a function and a part a component supplier can be implicated in the conceptual phase of product design.

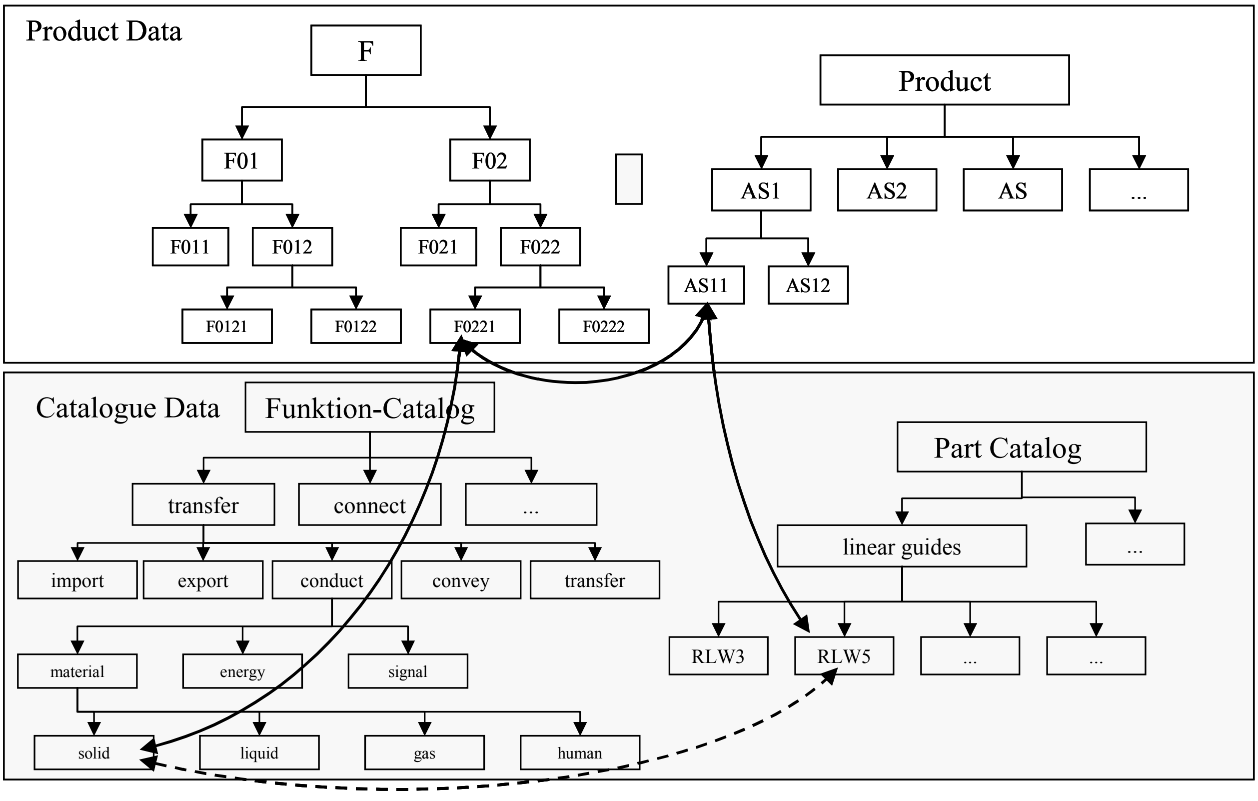


Figure 2: Linking Product Data to Catalog Data

Further benefits of product life accompanying function models are the connection to business relevant data. Function based estimations as starting point for economic decisions can be carried out efficiently over the entire product life cycle (i.e. value analysis). Changes in the physical design as for changing the component supplier can be evaluated trouble-free. Active assistance of product lifecycle management is given in further domains:

- change management
- reduction of part number (price, logistics etc) through function integration
- platform strategies
- modular design
- control of product function and function price

2.2 Efficient search for replicate parts

Most part management systems structure their content using a hierarchical classification systems. Due to a huge amount of data, this structure can grow immensely complex and confusing. Another drawback is that the structure is always created from the same viewing angle (also called the discriminator. An example of a discriminator is for instance the manufacturing method for a part (e.g. welding, casting, etc.). The searcher has to know the structure of the hierarchy in order to search successfully. This is increasingly difficult for great numbers of parts and different kinds of structures. Some parts are not classified at all making them almost impossible to find. This is one reason for the time and time again designing of existing parts and increasing costs. Searching parts delivers only narrow results. Alternative parts in other branches of the hierarchy are not found and shown.

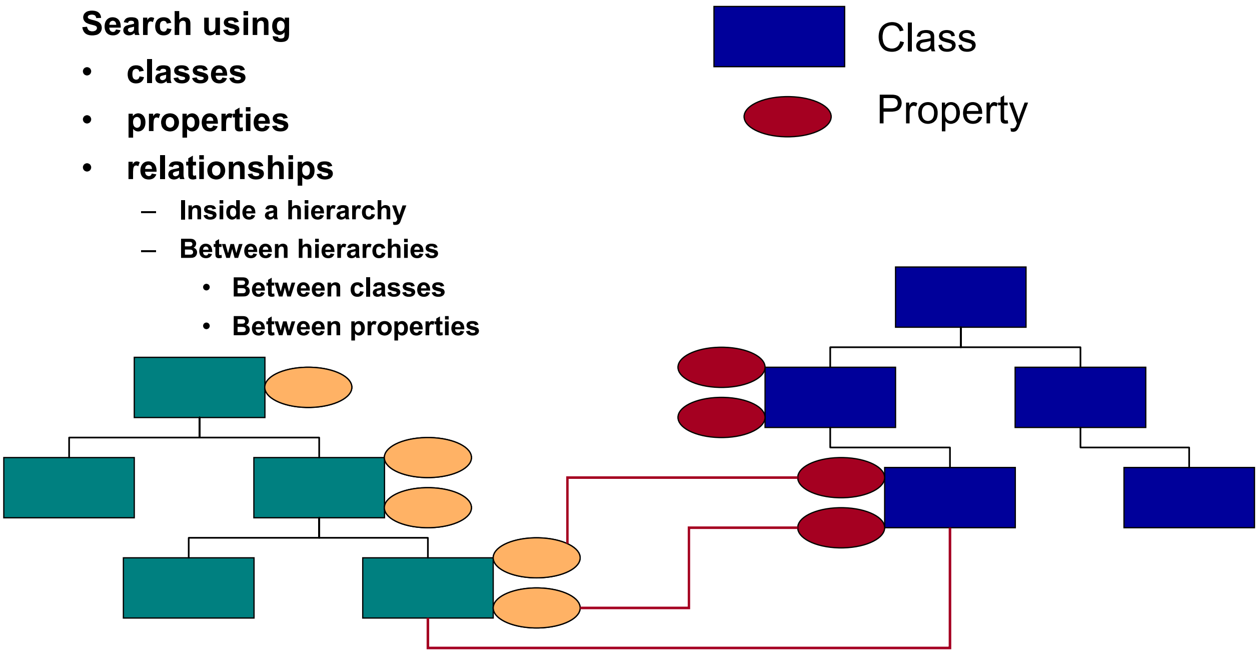
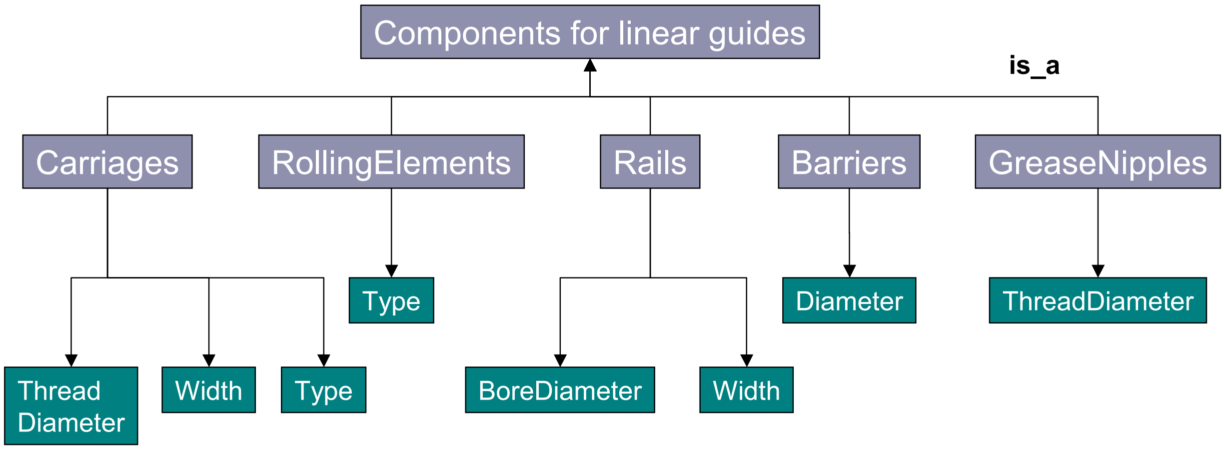


Figure 3: Search for replicate parts

A class system improves the search substantial. The search can be performed over the class structure using the properties and relations of the part classes. Property driven search delivers even if the part or the classification schema is not known. Results are all parts of possible different classes that possess the property. Knowledge of parts and classification schema is not anymore decisive in the search. The search can be performed just as good by less experienced designers.

2.3 Supporting automatic product configuration

The customization of products has growing impact on market success. Online catalogs offer potential customers the possibility to modify products configurations and check the outcome of their adjustments. Outcome is the need for representations of modification and combination possibilities in a modular design through rules (s. Fig. 4).



- **Carriage.ThreadDiameter = GreaseNipple.ThreadDiameter**
- **Carriage.Type = RollingElement.Type**
- **Carriage.Width = Rail.Width**
- **Rail.BoreDiameter = Barrier.Diameter**

Figure 4: Product configuration with class systems

This demands a broader description than i.e. numbering systems. The class system offers a ideal basis for modular design through the property based description and the multiple relations between classes and properties.

3. Prerequisites for introducing a class system

Main condition for the implementation of class systems is the support of object oriented class concept in PLM-Systems. Modern PLM-Systems facilitate the translation from numbering systems and master data to OO-Class oriented approaches. Beside the technical requirements concerning the IT a broad know how in the field of classification is needed. A extensive body of rules has to be considered while defining and structuring the classes. Following questions play an important role:

- Which objects will be managed by the class system?
- What is the maximal number of classes to be defined?
- How many hierarchic layers are appropriate?
- How will classes differentiate? What is the discriminator?
- Which properties are used for class definition?
- On which layer should the property be instantiated?
- Is new information generating a new class or a new property?
- What kind of relations between classes exist and are needed? (i.e. “is_case_of”)
- What kind of views on the data exist and are needed?

A thorough analyse of data and processes generates answers to most of the questions mentioned before. The class structure and definition of properties can be stipulated only working hand in hand with the specialist departments. The coordination process is usually accomplished in workshops under the moderation and guidance of classification experts.

Atos Origin is an international information technology services provider. Its business is turning client vision into results through the application of consulting, systems integration and managed operations, including outsourcing and on-line services. 30,000 employees world-wide deliver innovative solutions and strong sector experience to better manage the entire value network for companies and extended enterprises. The core areas of excellence comprise management consulting, enterprise and eBusiness solutions and outsourcing. Within the field of product lifecycle management Atos Origin offers a complete service portfolio from process analysis through system selection, customizing and system implementation up to user training. These services also cover the preparation of client-specific class systems and the implementation of these systems into the internal processes.

4. Conclusion

Product Lifecycle management (PLM) moves towards being the essential data management technology in business. PLM extends classical PDM functionality and enhances the pure technical order processing with all necessary information for support of the entire product life cycle. An example is the need for functions describing and managing information in the end of life phase of components and products like support of maintenance or recycling. Due to the extension of the function range, the integration of different software systems plays an important role in an efficient PLM solution. One of the basic conditions for PLM is, leaving out software and hardware technical criteria, the uniform or related data landscapes in cooperating enterprises.

The main aim of this contribution was to specify the considerably advantages that can be achieved by using class hierarchies for the description of product data. It went into detail in different lifecycle phases providing specific examples. In the early development stages function hierarchies for supporting PLM were considered ([6], [7]). Component hierarchies were treated pertaining general lifecycle phase overlapping matters: organization and non-ambiguous data, facilitate search operations, comparability of components, ease of attribute generation through inheritance, efficient search for similar or non-variable components [4] support for automatic product configuration.

5. References

[1] Altshuller, G.: And Suddenly the Inventor Appeared. Technical Innovation Center, Inc., Worcester, 1996.

[2] ISO (Hrsg.): ISO/CD 13584-10: Parts library: Conceptual Model of Parts Library. International Standardization Organization, Genf, 1995.

[3] ISO (Hrsg.): ISO/DIS 13584-1: Parts library: Overview and Fundamental Principles. International Standardization Organization, Genf, 1997.

[4] Meerkamm, H.; Heynen, C.; Sander, S.: Integriertes Komplexitätsmanagement auf Basis Intelligenter Lösungsbibliotheken. ZwF Zeitschrift für wirtschaftlichen Fabrikbetrieb 94 (1999) S. 498-502.

[5] O`Shea, M. A.: Produktkonzeption und -gestaltung - Ein Ansatz zur Überbrückung der methodischen Lücke. ZwF Zeitschrift für wirtschaftlichen Fabrikbetrieb 97 (2002).

[6] Otto, K.; Wood, K.: Product Design: Techniques in Reverse Engineering and New Produkt Development. Prentice Hall, Upper Saddle River, NJ, 2000.

[7] Pahl, G.; Beitz, W.: Konstruktionslehre: Methoden und Anwendung. Springer Verlag, Berlin, Heidelberg, New York, Barcelona, Budapest, Hong Kong, London, Mailand, Paris, Santa Clara, Singapur, Tokio, 1997.

[8] Roth, K.: Konstruieren mit Konstruktionskatalogen, Band 1 Konstruktionslehre. Springer Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, Hong Kong, Barcelona, Budapest, 1994.

[9] Sander, S.: Konzept einer digitalen Lösungsbibliothek für die integrierte Produktentwicklung. VDI Verlag, Düsseldorf, 2001.

[10] Stone, R. B.: Towards a Theory of Modular Design. Dissertation, The University Of Texas At Austin, Texas, 1997.

Authors:

Werner Puri — Chair of Engineering Design, University of Erlangen-Nuremberg, Germany
E-mail: puri@mfk.uni-erlangen.de

Stefan Sander — Atos Origin GmbH, Nuremberg, Germany
E-mail: stefan.sander@atosorigin.com

Harald Meerkamm — Chair of Engineering Design, University of Erlangen-Nuremberg, Germany
E-mail: meerkamm@mfk.uni-erlangen.de