Fraunhofer-Institut
für Arbeitswirtschaft und Organisation
Institutsleiter:
Univ. Prof. Dr.-Ing. habil. Prof. e. h. Dr. h. c. Hans-Jörg Bullinger

Postfach 80 04 69, D-70504 Stuttgart
Nobelstraße 12, D-70569 Stuttgart
Telefon (07 11) 9 70 01
Telefax (07 11) 9 70 - 22 99

WDK-workshop on

Product structuring

June 22 - 23, 1995

Delft University of Technology

Delft, The Netherlands

Fraunhofer-Institut für Arbeitswirtschaft und Organisation

Abstract

For handling the documentation of highly complex technical products like trucks, a data base system is crucial. The data must be in a consistent state and there should be no unnecessary redundancies. All departments which require product data should have efficient access to (only) the data they need, and this data must be represented according to the respective departmental views (like bills of material, assembly plans, etc.).

The Fraunhofer-Institut für Arbeitswissenschaft und Organisation (IAO), Stuttgart, has developed a prototype information system based on an object-oriented database system which enables users to issue information and update requests during the whole product life cycle. The underlying object-oriented data model conforms perfectly to the users' perception of the real-world products. It enables

- redundancy free data storage,
- fast response times, and
- reusability of object class definitions and application code,

and thus simplifies maintenance of the system.

This prototype manages the product data in an integrated manner throughout all departments of a truck manufacturing company.

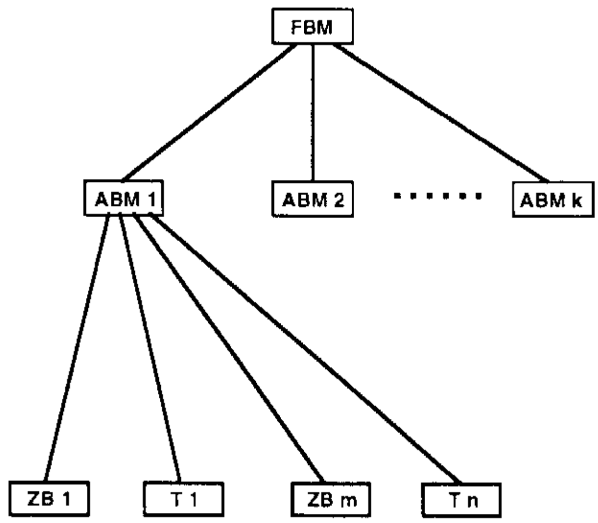
Reducing Complexity of Products by Rapid Product Development

International competition is characterized by an increasing dynamics of innovation. The gradually decreasing time of product marketing - in single cases, it is even shorter than the time of product development - requires enterprises to establish a growing number of new product developments and faster prototype development cycles in order to compete on the international market successfully. Thus, customer-oriented products which are tailored to the needs of particular target groups, as well as the early advertising of products gain more importance. The strategy of taylorization of work processes and the resulting, highly-specialized work distirbution lead to deeply structured, hierarchical forms of organisation, inhibiting not only fast reactions to changing customer requirements and short interaction cycles in product development, but also disabling innovation in enterprises. Consequently such enterprises are not flexible enough to adapt to today's market's dynamics.

Well known approaches such as tayloristic development or concurrent simultaneous engineering are completed by Rapid Product Development dependent on the degree of innovation. This contribution of the workshop will discuss the differences of these approaches, by a description of properties and methods, under consideration of developing innovative products.
The advantages of Rapid Product Development will be shown in a case study, which will complete the reduction of complexity of the product structure.

Product structuring in the automotive industry means a classification by

- Type (FBM)
- Aggregate (ABM)
- Assembly (ZB), Parts (T)



Step	Structure of Products (Trucks		
1	FBM	Fahrzeugbaumuster	Type
2	ABM	Aggregate baumuster	Aggregates
	M	Dieselmotor	Diesel Engine
	G	Getriebe (G)	Gear Box
	V	Vorderachse (A)	Front Axle
	H	Hinterachse (A)	Back Axle
	L	Servolenkung (C	Servo Steering
	F	Fahrgestell	Chassis
	A	Fahrerhaus (F)	Driver's Cab
3	ZB	Zusammenbau	Assembly
	T	Teile	Parts

Illustration 1: Product structure

The introduction of codes gives the designer the oppurtunity to offer the customer a large variety of products by using various different features for the configuration of products, e.g., trucks.
Standardized configurations do not require codes.

With regard to utilization there are three variants of codes:

1. Codes exist in the structure FBM to ABM, only.
   This means that for any code influencing the ABM, as well as for each code combination, a new ABM has to be generated (2^{n-1} problematic). Due to the high number of ABMs, this variant was not taken into consideration.

2. Codes exist in the structure ABM to ZB/T, only.
   This means that the utilization of new aggregates (e.g., automatic gear box instead of a six-speed gear box) would always require the creation of a new FBM, in this context each new combination of aggregates would require a new FBM. In the event that a new aggregate for a different utilization has to be built, it is possible to integrate the part into the base aggregate by using the respective code.

3. Codes exist in the structures FBM to ABM and ABM to ZB/F.
This kind of representation allows an optimized allocation of aggregates to the respective FBM and its contents.

Example: The customer desires an automatic gear box instead of a four-speed gear box.

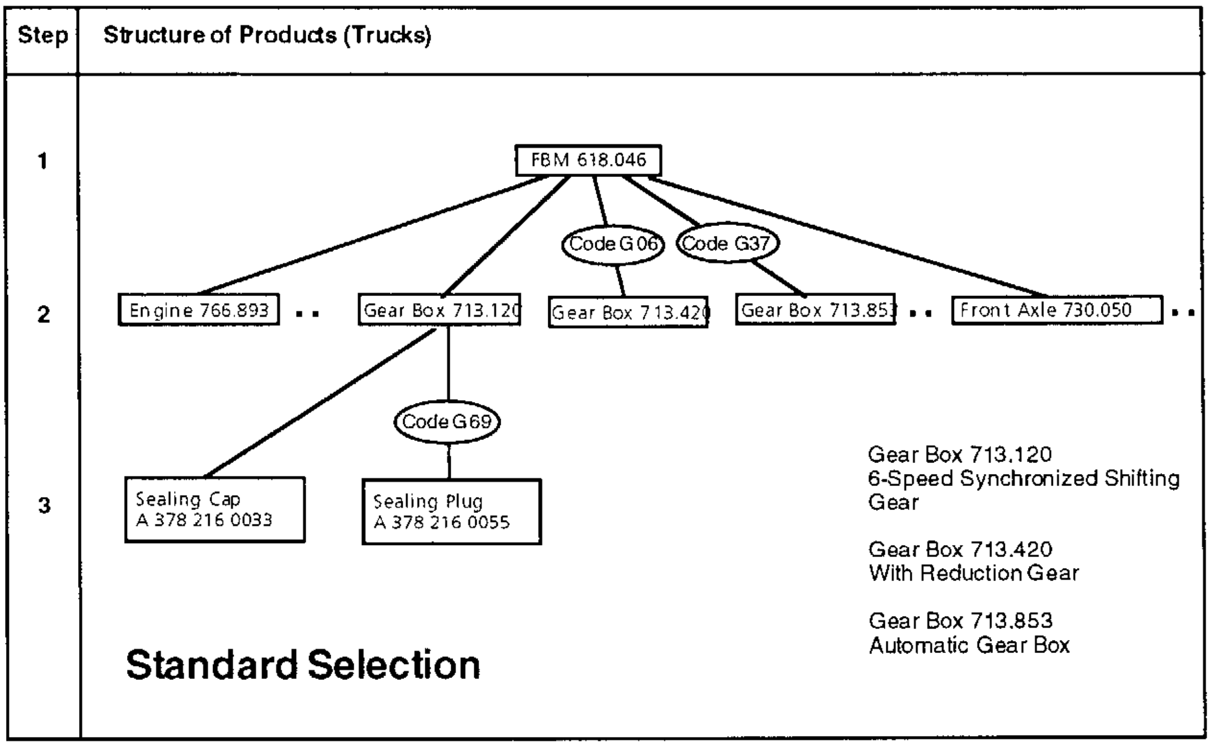


Illustration 2: Control of Aggregates and Parts via Codes.

In variant 1, a new FBM has to be generated.
In variant 2, the ZB/T of both aggregates, at the worst, have to be represented in a parts list.
In variant 3, the required aggregate is addressed by the FBM via the respective code.

Should there be several codes within one aggregate which apply to a single ZB/T with regard to its various utilizations in the FBM (e.g., utilization in FBM 1 as series, in FBM 2 via code), the approach is to integrate the type of utilization into the structure.

Development of a System for the Truck Configuration

In our institute a prototype information system has been developed based on an object-oriented database system which manages the product data in an integrated manner throughout the whole product development and production process. Applications with view to the entire process chain have been modelled within the system - including transactions as customer orders, management of bills of material, design changes, assembly plans, planning and calculations.

Requirements of different departments have been analyzed. Based on this data stating the respective views of the various users an overall model of the motor vehicle was compiled.

The individual departments were on the one hand interested in overlapping features and on the other in data only useful to them. Before, the overlapping data had usually been redundantly stored which caused long delay time and sometimes inconsistencies in the case of data changes.

As a result, the database had to ensure that the data remained in a consistent state and that there are no unnecessary redundancies. All departments which require product data need to have efficient access to (only) the data they need, and this data must be represented according to the respective departmental views (like bills of material, assembly plans, etc.).

In the development process of the object model of the vehicle data, objects like trucks, subassemblies of different complexity and single parts turned out to be the most important objects. Most of the remaining data (like prices, graphics, descriptions of parts, etc.) was attached as features to the main object. The complex nesting of the subassembly structure was modelled as objects nested in other objects exactly as the object-oriented paradigm suggests. This nesting involved many levels.

The object-oriented data model perfectly conforms to the users' perception of the real world products, and thus simplifies interaction by the user as, e.g., refining and also changing the data model in the product development, according to new customer requirements.

The objects are also grouped into classes and subclasses. The class hierarchy consists of only a few levels compared to the levels of nesting objects. This is due to the fact that in this application the focus of interest is more on the complex structure of the products than on the classification of the products. For the development system only one series of trucks was modelled. If their had been more series the hierarchies would have grown accordingly. The methods to be performed on the instances of this class were stored together with the class definitions. The required behaviour of these methods was, of course, determined by the business transactions that had been defined in the beginning.

The information of the different departments is generated from the object model of the motor vehicles, e.g., different kinds of parts lists, cost information, etc. This generation is performed by means of a query to the object-oriented database which were coded as methods (procedures) that were stored in the database along with the respective object classes.

The resulting system with the business transactions on top of the database system enabled the concurrent execution.

The resulting system with the business transactions on top of the database system enabled concurrent execution of different processes working with up-to-date, non redundant, shared information.

For the purpose of comparison the same application was implemented usling a relational database system.

The complex nesting of objects was more natural to be modelled by the object-oriented system than by the relational one. The availability of more data types (like for example graphical data) simplified the implementation of several business transactions.

For some queries, like compiling a list of the usages of a given part, the object-oriented system had faster response times. In the relational system acceptable performance could only be achieved by denormalization and thus introduction of data redundacies. This trade-off between non-redundant data and performance does not exist in the object-oriented system.

Conclusion

In the described project, we have developed a prototype of an integrated information system based on an object-oriented database that realizes the integration of electronical information processing for different phases along the process chain of an automotive company. As a result, the information flow between the departments is accelerated, inconsistencies arising from redundant data management are avoided, and up-to-date information can be shared in concurrent processes.

The developed system takes advantage of object-oriented system features for handling the challenges of complex technical data.

Naturally manufacturing companies nowadays use already a variety of databases and other information processing systems (CAD, CAP, etc.) and cannot start form scratch replacing all of their existing systems at once with new, object-oriented systems.

A feasible strategy, however, is to use this newer technology first for few restricted applicatons and step by step include more data and applications. This means that at the beginning interfaces do existing datab ases and other systems have to be used.