INDUSTRIAL CAD/CAM APPLICATION AND SYSTEM ARCHITECTURE - A CLOSED LOOP

C. Werner Dankwort, Gerd Podehl
Institute for Computer Application in Engineering Design
University of Kaiserslautern
Kaiserslautern, Germany

ABSTRACT

In the past years, development and production processes in many companies have changed in a revolutionary way, leading to new demands in information and CAx technology. The R&D-departments of the German automotive industry installed a working group to develop a common long term CAD/CAM strategy[1]*. A preliminary result is the concept for an open CAx system architecture as a basis for realizing industrial requirements on CAD/CAM and for the cooperation with system vendors. The project ANICA was started in cooperation with five international CAD/CAM -suppliers in order to show the feasibility of this architecture. The access interfaces of different system kernels are analysed with the aim of developing a concept for a cooperating CAx system network. The concept will be put into practice with a software prototype basing on CORBA and OLE. The communication elements within such an architecture have to go far beyond conventional CAD data. This will lead to an extension of "feature" concepts including CAx functionality and dynamic information about the process chain of a product. The impact on modern concepts for user interfaces, on reverse engineering methods and on product data models will be discussed to finally close the loop to industrial CAx application.*

KEYWORDS

CAx Technology, Open System Architecture, CORBA, Process Chain, Features

1. INTRODUCTION

During the past years many significant changes have taken place in the industrial world. Today companies are decreasing not only their in-house production depth but also their in-house development depth. Car-manufacturers concentrate on their core processes: Styling, car body and total car. Cooperation between manufacturers and suppliers are getting more and more global crossing international borders.

The processes of developing and manufacturing a product must become faster and faster with the consequence of high parallelism and the replacement of conventional hierarchical organisation structures by project teams. The objectives are:

- Shorter time to market
- Higher product quality
- Reduction of costs

In the computer aided (CAx) area this situation leads to a paradox: In order to optimize the CAx communication within and between the companies the big manufacturers exercise an influence on the subcontractors to use exactly the CAD system the orderer uses. - On the other side, the big companies know quite well that there is an urgent need for working in an open and modular CAx system world.

In information technology the techniques have also changed drastically: Networks of workstations are now state-of-the-art for hardware while the software trends point to object-oriented systems.

Being aware of these problems and trends, the German automotive industry implemented a working group to develop a long term strategy for their future CAD/CAM applications.

2. CAD/CAM - STRATEGIC INITIATIVE OF THE GERMAN AUTOMOTIVE INDUSTRY

2.1. Objectives of a CAD/CAM - strategy

The objectives for a CAD/CAM-strategy can be divided into strategic ones concerning the processes within the development and manufacturing of a product, and into operational ones concerning the CAx field (Dankwort, 1995):

The strategic objectives are:

---

[1] AUDI, BMW, MB, PORSCHE, VW, University Kaiserslautern: Working Group "CAD/CAM Strategies of the German Automotive Industry",
G. Weißberger, H. Ederer, Dr. W. Renz, D. Leu, Dr. P. Kellner, Prof. C. W. Dankwort

- Reducing costs, shortening time to market, and increasing product quality
- Gaining control of the entire process (in-house and external)

The operational objectives are:

- Control of the data flow (CAx data and administrative data), meeting the demands of process communication (in-house processes and company overlapping processes)
- Flexible and economical CAx technology: Only possible through standards
- CAx-functionality: Principal possibility to create any required tool
- Open modular systems and a close cooperation between users and suppliers
- Independence from subcontractors (i.e. hardware and software suppliers)
- Decentralisation of resources and capacities
- Reduction of costs and expenses for technology (hardware, licences, etc.) and applications (user support, training, etc.)
- Solution for the actual problems of data exchange

The objectives are of quite different levels: The last two concern very pressing present problems and the normal user is only interested in solving these. But from the long term point of view, the other objectives are of higher importance, otherwise all companies will run into severe problems in the future.

2.2. Open system architecture

As a solution for these problems and for reaching the general objectives, the German automotive industry has developed an open CAx system architecture given in figure 1.

The goals which this architecture has to fulfil are:

- To avoid user dependence on a specific system (overcoming forced bondings - building partnerships)
- To support each process by specific optimized CAx tools
- To adjust CAx support by using software "building blocks"
- To assure that the user can freely act as the owner of his design results (without needing help or permission from the system supplier)
- To describe the complete product definition and its life cycle (product model)

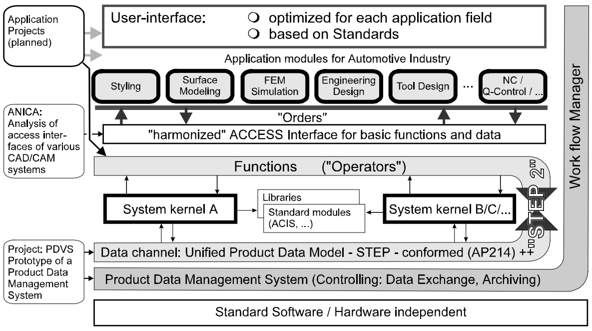

Figure 1: CAx Architecture of the German automotive industry

Technically speaking the architecture implies: From the users point of view, "the CAx system" does not exist any more. The system landscape consists of "building blocks" and "tools" depending on the individual requirements and need not necessarily come from the same vendor. The basic system kernels are - in theory - exchangeable. All levels in the architecture are clearly separated: The data base management, the system kernels, the user interfaces, the pure data and the CAD functionality (both being STEP compliant). The architecture is independent from underlying hardware and operating systems. As far as possible standards have to be used. The access interface, adjusted to data, functions and applications is the key point within the architecture.

2.3. Consequences for users and system suppliers

The requirements of companies using CAx techniques are fulfilled:
They are independent from the vendors. They have a high level of assurance of investments. The costs can be adjusted to the market situation and to the real needs of application. The entire process, in-house and between companies, can be supported by optimized CAx tools. The information flow is under control, because of using non proprietary data structures as close to STEP as possible. Even the problem of long term archiving design data may be solved by means of easily linking some data specific preprocessors.

On the other hand the CAx vendors will change the policy of their enterprises:
Their main business will no longer be the selling of software but more and more the offering of consulting to support the processes of their clients, which are the product manufacturers. To do this the new consulting enterprises will offer software from their own in-house development, from third parties and even from competitors. This means that they will

reduce their in-house development and production depth concerning software in a similar way as it is happening now in automotive industry.

3. PROJECTS FOR THE VERIFICATION OF THE CAD/CAM SYSTEM ARCHITECTURE

Projects can work on three levels within the architecture: Two projects have already been started by the working group "CAD/CAM strategies of the German automotive industry" in order to prove the feasibility of the concept in industrial practice: The project PDVS for the data specific level and ANICA for the more system technical level:

Product Data Management System (PDVS):
A software prototype is being developed on the basis of STEP AP214 (Application Protocol Automotive design) to show the possibility of integrating conventional CAD data and those data describing the structure of a part. The prototype will be integrated into one ore more commercial CAD systems and will be tested by pilot users in the automotive companies which are sponsoring this project. PDVS is under direction of the ProSTEP GmbH, Darmstadt, Germany.

Analysis of the Access Interfaces of various CAD/ CAM-Systems (ANICA):
The most important problem to solve for the feasibility of the architecture is the access interface, which must be harmonized to access the system kernels of various competing system vendors. The access to the CAD/CAM system kernels which will be available in the near future is being analysed in ANICA and a common access structure will be developed. Figure 2 shows the basic idea of this concept.

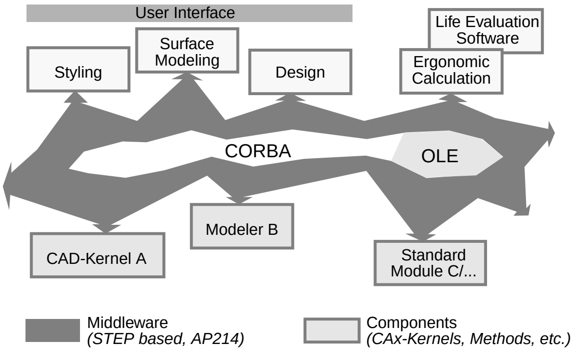

Figure 2: Concept of ANICA

A software prototype is also under development to prove the possibility of concurrent real time CAD design with several CAD systems on different hardware platforms. ANICA is a project of the German car manufacturers together with five system suppliers under the direction of the University of Kaiserslautern, with financial support from the Rhineland Palatinate Foundation for Innovations.

On the highest level, packages for special user applications can be developed. One project is under planning at the University of Kaiserslautern in cooperation with industry partners and software suppliers.

4. FORMALISATION OF THE CA - INFORMATION FLOW

4.1. Industrial workflow

Working on industrial problems, one always faces the enormous complexity of modern process chains and the huge amount of formal and informal information which must be exchanged to keep the process running. Therefore much effort is spent to build formalization schemes making it easier to gain control of the processes. One preliminary idea of how a formalization could look like is presented in this chapter.

To understand the industrial workflow (i.e., the processes of development and production) entirely, all information and process steps have to be represented in a clear, transparent way. A frequently used technique is the SADT scheme (structured analysis and design technique) or one of its modifications. As an example a basic process element is given in figure 3. Every box may contain additional processes which themselves can be detailed on deeper levels. One real process (sheet metal stamping in automotive industry) was analysed with SADT by the German Automotive Industry Association (VDA, 1991).

To entirely describe the flow of a process, the describing data can not only consist of conventional CAD data (i.e., geometry), but must be able to capture much additional information, e.g.:

- Structure of the part (geometric neighbourhood, bill of material)
- Design history, design methodology (including also the function of the parts)
- Relationships to model building and to manufacturing
- Administrative data (costs, capacities, time schedules, etc.)

Most of these data are beyond the possibilities of CAD/CAM-systems. This information is organized in conventional administrative data bases or in workflow manager systems respectively. It is stored in conventional formats, due to the possibilities of the used data base management system (DBMS).

4.2. Process elements

Following the ideas of the SADT process structure a basic process element is presented in figure 3.

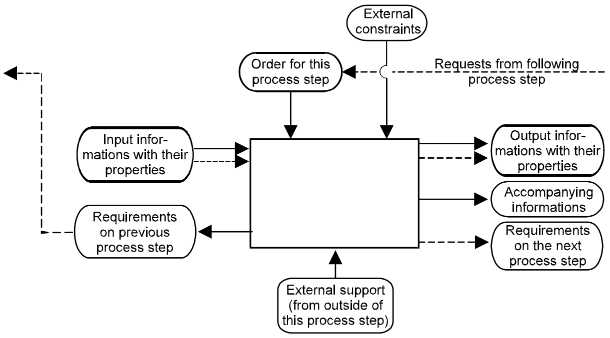

Figure 3: Basic process element

This pattern goes beyond the simple scheme of < input | process | output >. The element carries additional information and properties which can be described as follows:

- Every process step (represented by such an element) has requirements which the previous process steps must meet
- Every process step must meet the requirements, demanded by the subsequent steps
- Every process step requires specific properties for its input information
- The properties of the results of the actual step cannot be listed completely. They must be considered from the point of view of the actual step.

The key points within the process are the "arrows" between the process steps. Thus, the process should be written in a different way. In general, application processes are very complex with different steps running in parallel. For simplification only a short sequential process will be considered (the scheme introduced in this paper is capable of dealing with parallel processes as well, where one process gets input from and delivers output to several parallel steps):

----> step 1 ----> step 2 ---->

We use the following abbreviations:

ii.x = input information for step x
oi.x = output information of step x
ipr.x = properties of input information for step x
opr.x = properties of output information of step x
Px = processor of step x

Now two subsequent steps can be written as

< ii.1; ipr.1 | P1 | oi.1; opr.1 >

< ii.2; ipr.2 | P2 | oi.2; opr.2 >

Considering the properties of some information the complete set of properties is in general unknown. Of interest and only to be evaluated is the subset of properties which is important for the work in some special step of the process. Complex design steps, collecting several simpler ones can be described by building complex processors which include these simpler process elements.

For a well running process the "arrows" in figure 3 correspond to the condition

| oi.1; opr.1 > < ii.2; ipr.2 |

where the second part (input for step 2) must be a subset of the first part, which is the result of the previous step 1. In process step 1 there should not be produced more, than is required in one of the subsequent steps (and - of course - not less). In industrial life this idea is used to optimize the process chain by looking at the processes from the back end and then limiting the process outputs to the requests of subsequent steps (backward analysis).

In reality the formal link between two process steps often does not work, e.g. if "**| 1 >**" are a VDAFS data set and "**<2 |**" must be IGES. In such a case a "translator" is needed:

< oi.1; VDAFS | T(VDAFS/IGES) | ii.2; IGES >

The basic process element in figure 3 has a lot of "open ends". Considering our new scheme

| oi.1; opr.1 > < ii.2; ipr.2 | P2 | oi.2; opr.2 >

the properties of the input for step 2 must certainly meet the requirements of step 2. Moreover, the results of step 1 must meet the requirements of step 2. Following this logic, it might be useful to take the part

| oi.1; opr.1 > < ii.2; ipr.2 | P2 |

as a basic element. Technically it may be easier to integrate the requirements of the next steps within the processor P instead of deriving them from subsequent output data only. In such a case a basic process element shall be

< i | P | o >

as a building block of the process. In any case the processor P will capture the functionality from inside the process step and react on requests from outside (including external constraints, support, etc.):

< ii; ipr | P(functionality, ext. requests) | oi; opr>

If we assume that each step only produces results which are requested by subsequent steps, the formal treatment of the output could be omitted completely, because it simply is the union of all requests by subsequent steps (here: steps 2, 3, and 4):

< ii.2; ipr.2 | P2
P1 | > < ii.3; ipr.3 | P3
< ii.4; ipr.4 | P4

As real processes are in general not sequential, every process step should know, where to get its input information. The easiest way to achieve this is to store for every input information its source (i.e. the step where its comes from) which can be done simply by pointers. With **is** being the input source (in fact **is** is a vector of sources) and **o** just the union of subsequent external requirements, the basic process element could be written as

< is; ii; ipr | P(functionality, ext. requests) | o >

Figure 4 shows a picture of this proposed element and how the elements could be linked together in a process chain.

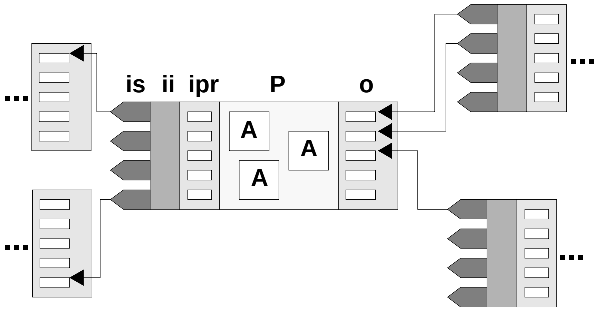

Figure 4: Process element with links to other elements. **A** stands for the algorithms which realize the functionality of **P.**

4.3. CA - design process

The design work of an engineer in front of the CAD screen may be also described using this pattern. In most cases some geometric entities have to be generated, modified or deleted, due to the intention of the designer. The process is running through a sequence of design steps. Within one CAD system the pattern reflects the structure of history based design:

<source; input geometry; required properties
| CAD function |
new geometry; requested properties >

As an example, the CAD function for calculating intersections will be examined: The input geometry can be surfaces, lines, or other entities. The sources will be defined by the users interactive mouse click on the screen or some batch processing. The required properties depend on the chosen algorithm in the CAD function to produce the results which subsequent steps have requested. If they requested a 3D intersection curve or just a number of equidistant points, different algorithms will do the computations, needing different kinds of input entities and parameters.

Again the link between two process elements is the critical point in the workflow: Does the "result" with its properties of one design step meet the requirements of the subsequent step and does it meet the designer’s requirements? In the usual way of working the designer is not oriented about the formal requirements of the following design steps, but he knows by experience what he has to do.

4.4. Communication within the ANICA concept

In the ANICA project a component based system is built, not comparable with conventional monolithic CAD systems (Dankwort et al., 1996). Software components are linked together by a software bus. For the implementation of this software bus there are several possibilities: pipes, remote procedure calls (RPCs), etc. Within the ANICA prototype CORBA is used (Common Object Request Broker Architecture) to be on a higher level standard.

A component based system can only be built, if the interfaces of the components are standardised, i.e. they have well defined signatures. The aim of the ANICA project is to identify some basic interfaces of CAD components and to describe them independently from the underlying system (Janocha et al., 1996).

From the designers point of view the work within the ANICA architecture allows to work with elements and CA-functionalities of different CA-systems at the same time.

The procedure of a very simple distributed CA-design with ANICA could run as follows:

- The user is working with entities of a local system kernel. (Local server "X")
- The local server "X" sends a request to remote server "A" to generate an element type "a"
- Server "A" sends in return to "X" the reference to element "a"
- Server "X" sends a request to "A": Send visualisation data of "a"
- ...
- Server "X" sends a request to "A": Do CAD function "F" with "a" (e.g. modification)

- ...

Some steps of this process could be written in a simplified way as follows:

< X; request; properties | A: creation | element "a"; reference >

< X; reference to "a"; properties | A: send | visualization data >

< A; visualization data; properties | X:visualize | picture of "a" >

and so on. Here the basic process elements are considered to work between different servers. In these distributed processes some problems are not solved up to now, so no attempt has been tried to apply the pattern sketched above.

5. USEFULNESS OF STRUCTURED PROCESS ELEMENTS FOR INDUSTRIAL CA-APPLICATIONS

Some possibilities to use the simple process description will be pointed out:

User interfaces:
Standard user interfaces of CAD/CAM-systems suffer from problems concerning the access logic (functionalities and/or element types) and from supporting too many functions with too many different options. As a result no user is able to make use of all the possibilities of a systems even if many hours of training were invested in him. The logic of the structured process element can be used to define a user interface with the possibility of equivalent access to all three aspects: Input, result and processor.

Simultaneous engineering:
A formal structuring of processes is necessary to decide on the degree of independence of different subprocesses and on the equivalence of process parts which could be used as exchangeable and repeatable building blocks.

Reverse engineering:
The fundamental idea of the process structuring lies in the importance of the requirements of subsequent process steps. This means that the desired results with their desired properties are the main aspect for the efforts to control the process steps.

Dynamic product information model:
Up to now the product data models only consist of static data (mainly geometry). These data models can describe a product only at a single moment in the product life cycle. From the authors' point of view there is no chance for *one* product data model describing a product during its complete life cycle. In the present data models no functionalities, no "purpose", no time dependence is captured. The process elements outlined here may capture all these aspects. The representation, communication and storage of design history and methodology may be possible.

Features:
In the FEMEX[1] group a new, general definition of features is being developed. These extended features also capture the specific views (of the user) onto the product description with respect to the classes of properties and to the phases in the product life cycle. The overlap between this definition and the basic process element is evident. Because of the industrial application aspects of the discussed process element it may be called *"feafi"* (feature for industry).

6. CONCLUSIONS

With the proposed CAD/CAM system architecture it will be possible to overcome the philosophy of using one big CAD/CAM system for one company. The impact on the strategies of the system suppliers was pointed out, the main business will be the support of product development and manufacturing processes within and between companies. The proposed formal description of processes may be a helpful tool. These ideas are of course in the beginning. Projects like ANICA have just started or are in the planing phase to modify the CAD/CAM system landscape with respect to the changes in the industry.

7. REFERENCES

Dankwort, C. W., (1995), "CAx-Systemarchitektur der Zukunft"; *VDI Berichte,* Vol.1216, pp. 103-116.

Dankwort, C. W., Janocha, A., (1996), "Von Monolithen zu Komponenten: CAx-Architekturen im Wandel"; *Proceedings of CAD '96,* Vol. 1, Kaiserslautern, pp. 358-368.

Janocha, A., Arnold, F. (1996), "Architekturkonzepte integrierter CAx-Systeme"; *Proceedings of the International Conference on Computer Integrated Manufacturing, Zakopane,* (to be published).

---

[1] FEMEX (Feature Modelling EXperts): A group of several international research institutes, system suppliers and application companies with the aim of harmonizing the international feature activities.