Integrated Data Management in CIM Systems

J. Göers, U. Osnabrück[1]
K. Ecker, TU Clausthal[2]

Abstract . The data management in the various components of CIM systems is still realized by individual databases where communication between the components is established via interfaces such as STEP. More efficiency, however, is achieved by an integrated approach which is based on an object-oriented database system. This would allow an integrated data management that avoids data redundancy and inconsistencies, and supports all kind of complex structured data. In this talk the requirements for an integrated and distributed data management are discussed.

1 Introduction

The situation in companies, especially in those with computer aided manufacturing (CIM), becomes more and more difficult with respect to data management. The difficulty increases with the amount of the data arizing there and with their diversity. These problems are mainly a consequence of incompatible data management systems used in different divisions of the company. Obviously, this incompatibility complicates communication between CIM components because isolated solutions do not provide direct information transfer. The consequence is that information exchanges between these areas are not only time and cost consuming, but has also serious drawbacks such as information losses. The following criteria characterize the present situation:

– redundancy of data because of multiple storage,
– inconsistencies between data of different CIM areas: this happens if e.g. updates of data are not distributed fast enough among the CIM components,
– huge amount of data,
– no integrated view of data and missing support of data dependencies in different CIM areas.

One way to surmount these problems would be an approach where all data are modeled in an object-oriented data model (OODM) and stored in a distributed object-oriented database system (OODBS). In Section 2 we first describe the data flow in CIM systems and analyze resulting requirements for global CIM integration. Section 3 contains an overview on already existing integration concepts. In Section 4 we propose a new integration concept based on an object-oriented database system that has been developed in our Institute, and discuss its advantages upon existing concepts.

2 CIM-Demands

The requirement of a consistent CIM integration concept arizes from the flow of data between different areas of a company. To analyze this flow we start from the following general definition of a CIM system:

CIM is an integration of PPS (production planning system), CAE (computer aided engineering), and CAM (computer aided manufacturing), where CAE comprizes the

[1] Jutta Göers, Institut f. Angewandte Informatik, University of Osnabrück, Germany.
E-mail: jutta@informatik.uni-osnabrueck.de

[2] Klaus H. Ecker, Institut f. Informatik, TU Clausthal, Germany.
E-mail: ecker@informatik.tu-clausthal.de

areas CAD (computer aided design), CAP (computer aided planning), and CAQ (computer aided quality). Between these components (i.e. between the corresponding departments of a company) there is a steady flow of information, as shown in Figure 1.

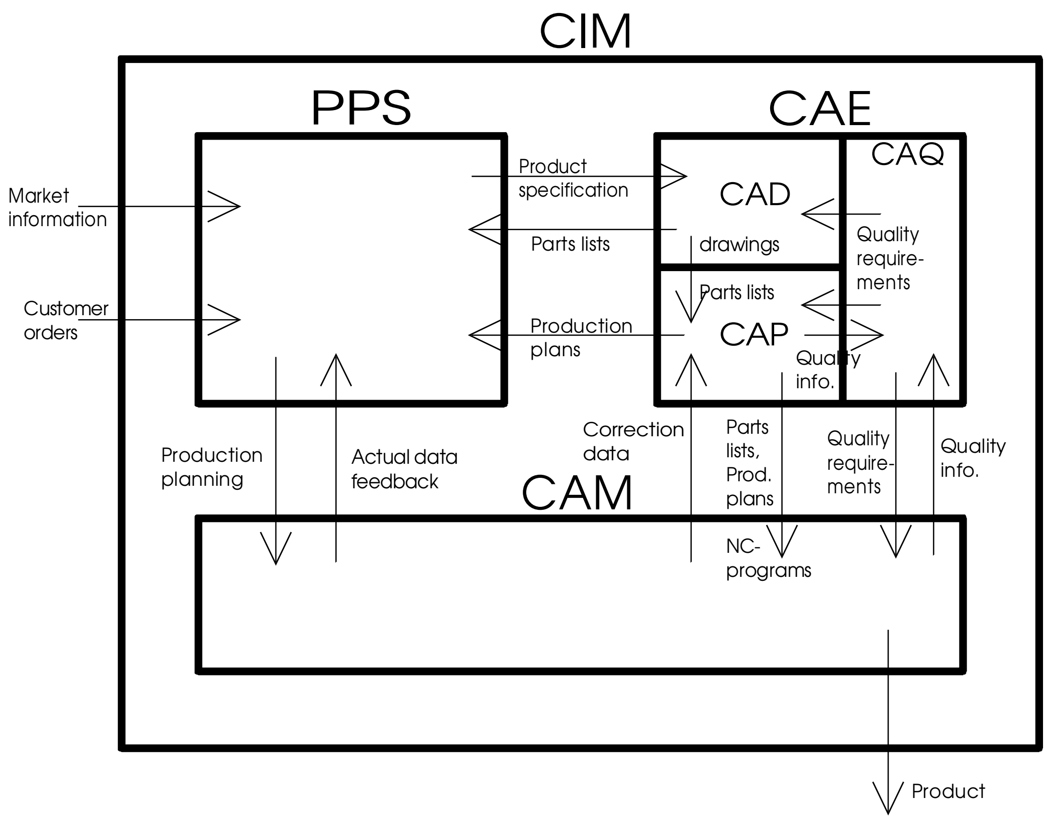

Figure 1 *Data flow in CIM systems*

Sketching a typical development of the data flow, we start with information entering the system: market trends and customer orders. This information is analyzed in the production planning and -control department. Product specifications of customer orders are forwarded to the design (CAD) division, and general data for production planning are transmitted to the manufacturing division.

In CAD product designs, drawings and parts lists are developed under consideration of quality requirements from CAQ. These are transmitted to the planning division (CAP) for further processing and to PPS for long term storage. In CAP blue prints and NC-programs are developed on the basis of the CAD data and then transmitted to the manufacturing division. If production turns out to be not possible due to insufficient or erroneous data, corresponding messages will be returned to CAP. But if production is successfully completed, the PPS division is informed appropriately.

Finally the product is delivered to the customer. This short description of the data flow shows already that almost all divisions need to communicate with each other, and almost everywhere data are generated that are of relevance in other areas of CIM. Due to the diversity of these areas the data can be of different structure. Hence, the data must be appropriately adjusted before their transmission to another division, or the receiving division must extract the relevant information from the transmitted data.

The different nature of data can for example be seen from constructional/design data, information about materials, machines, employees, customers, etc. Examples of extracted data are lists of parts, tolerances, or metric information. It follows from the above considerations that there are a lot of requirements to be supported by an integrated CIM concept. These are summarized as follows:

1. There must be a communication interface between the CIM areas.
2. The integration concept should offer a highly flexible and complex modeling of data types.
3. The communication interface should allow restructuring of transmitted data and extraction of partial information.
4. Data should be queried easily and stored without redundancy.

3 Overview on Existing Integration Concepts

The development of integration concepts for CIM systems was influenced by the current state of the information technique. The 80s, e.g. were characterized by the development of isolated solutions for the various areas of CIM. In particular, commercial systems and technical solutions were strictly separated from each other. This separation was even promoted by many software developers.

Later, however, it was realized that essential improvements of the trade competition of an enterprise are only possible by company-wide solutions where information from all company divisions are taken into account. This was the reason that first integration steps were taken in computer aided production.

Following [Sch90], five integration steps in the development of CIM systems can be distinguished:

Integration step 1: Coordination of unconnected systems (see Figure 2). There is no support for data exchange between components. Data transmission is performed by hand which, besides the unjustified expense of entering data again and the danger of typing errors, results in high redundancy and has the danger of inconsistencies.

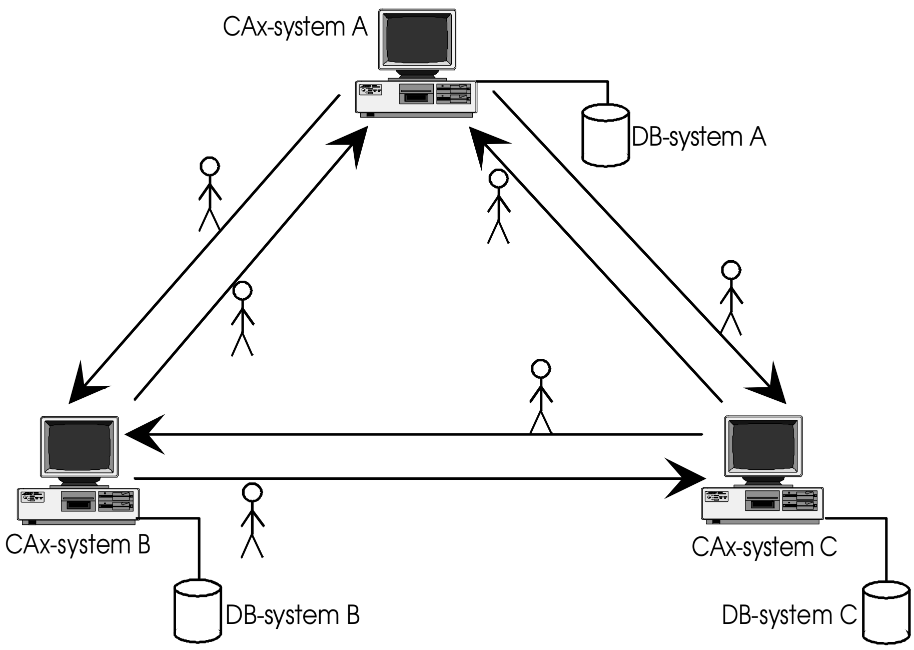

Figure 2 *Integration step 1: Organization between separated system components* [RG91]

Integration step 2: Integration of unconnected systems by tools (Figure 3). Systems are connected by a network which is able to transmit data between components. Due to different data formats transformations between various representation models and internal representations are required. In each direction of a link there is a processor which takes care of the data adaptation. This type of integration does not ensure data integrity, and it still suffers from data inconsistencies. If there are n connected systems, this kind of solution requires altogether $n(n-1)$ processors to handle the communication.

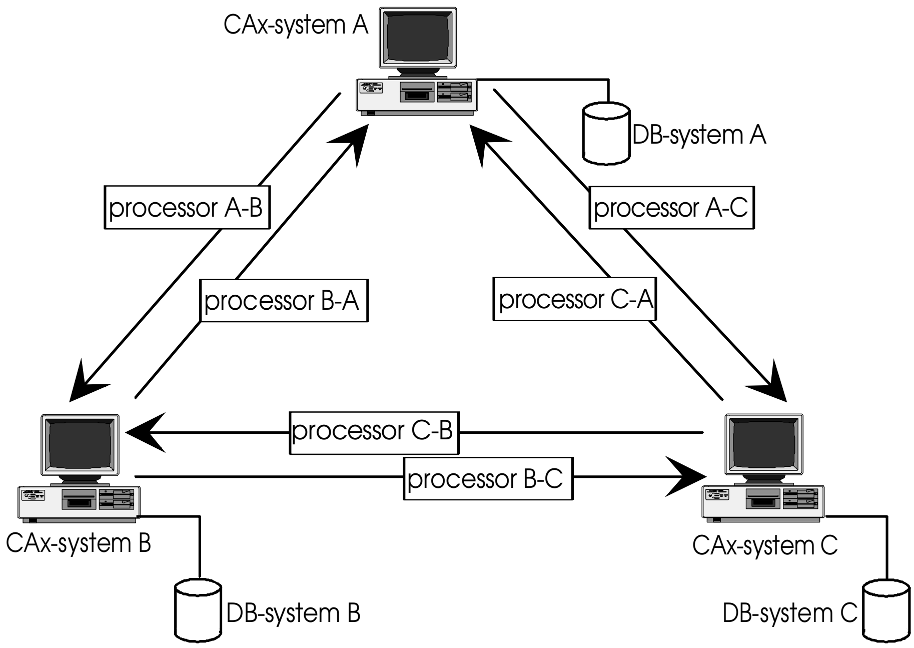

Figure 3 *Integration step 2: Integration of separated system components by tools*

Integration step 3: Data transfer by means of mailbox systems. Data are transmitted between the system components by means of an interface file which both parties are able to interpret. Data exchange is being performed via special mailbox systems. This solution requires special processors for re-formatting the files. Again $n(n-1)$ processors are required.

Integration step 4: Common reference model (see Figure 4). In contrast to the previous steps, there is no direct data transmission. Transformation is realized by a global reference model. Before transmission, data are transformed into the reference model, and after transmission the target system performs transformation into its own data format. Hence the are only two processors in each component system, one for transforming data into the reference model, and another processor for the backward transformation. Hence this solution requires altogether only $2n$ processors.

Integration step 5: Common database (see Figure 5). In this integration step there exists a common database for all system components which stores all the various data of the company only once. Each component is allowed to access the data according some defined view. This solution results in high data consistency and protects data from unauthorized access. On the other hand, due to the data centralization the central database is a bottleneck which causes delays in case of heavy traffic. Another drawback arizes from the fact that each system component will have data that are only of local interest.

A variant of this integration step allows to store these "local" data in addition in a local database with the consequence of data redundancy and the danger of inconsistencies.

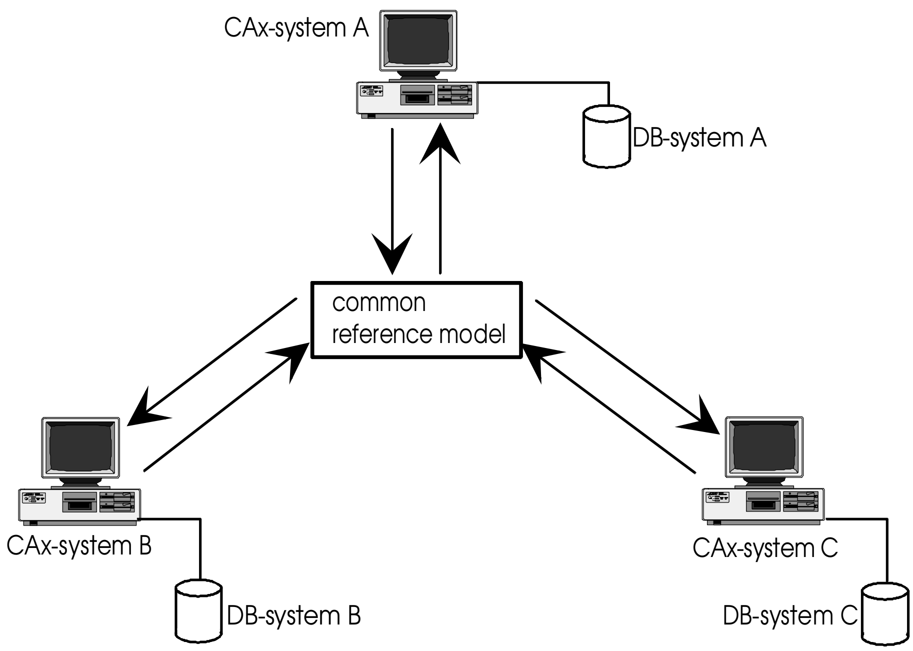

Figure 4 *Integration step 4: Common reference model for all system components*

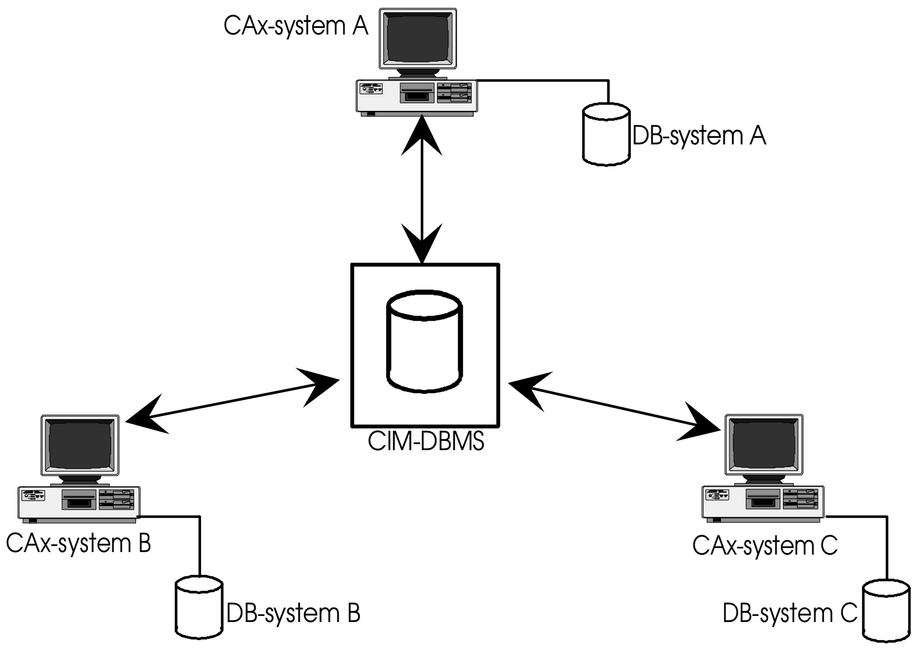

Figure 5 *Integration step 5: Common data base for all system components*

4 A New Integration Concept

Recalling the requirements on integrated CIM systems elaborated in Section 2, we see that two of them suggest the usage of an OODBS (object-oriented database system): a flexible modeling of complex data types, non-redundant storage of data as far as possible, and simple but powerful query methods. The requirements of non-redundancy of storage and powerful query methods are generally obeyed in modern databases. The realization of arbitrary data types, however, requires an object-oriented system. Hence we have to find out HOW far system integration can be supported by an OODBS, and how far the other/remaining requirements can be realized.

The concept proposed here starts, in accordance with the 5th integration step outlined in Section 3, from a common database system. Our approach is based on an *object-oriented* and *distributed* database system. The reason for a distributed system is that in each CAx component there are data which are either of only local relevance, or they need to be protected against accesses from other areas of the company.

For maintaining global data, i.e. those data that need to be accessed by all divisions, there are several possibilities. One solution would be a global server keeping all global data: each division sends all those data to the server for which access is required from other areas. This approach can be realized by a client-server architecture. In addition to these global data the server keeps also all centrally stored system data (see Figure 6, dashed lines).

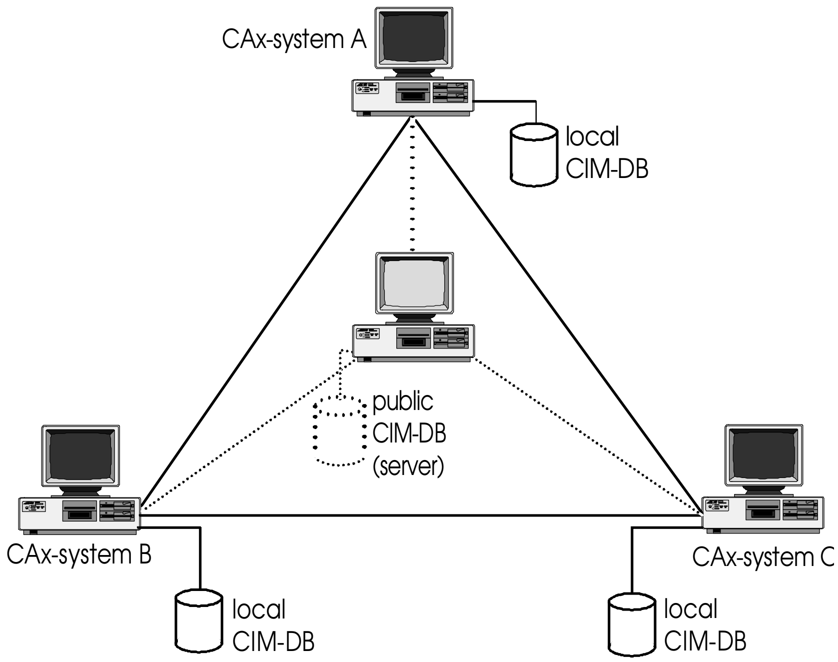

Figure 6 *A distributed database system as the kernel of a CIM system*

Another possibility is to store all kinds of information locally, no matter if the data are of local or of global interest. Each database component needs a - generally accessible - ”table of contents” which shows the globally accessible data and where they are stored (see Figure 6, solid lines). Both solutions have their advantages:

– Since there is only one data model throughout the whole system (i.e. a distributed database system is applied) there are no transformations needed between different data formats or data models.

– All data are stored only once in the system, i.e. there are no redundancies, and inconsistencies are thus avoided.

– Data protection is easily realized: In case of the client-server architecture only the data of the server can be accessed from all parties. If there is no server, data protection mechanisms like views provided by the database system allow only restricted view to the data of a division by showing only a part of the table of contents of their data.

– If a client breaks down the other divisions may continue, as long as they do not need data from the failed division.

– In both architectures the system performance is reasonable. Though in client-server architectures the server is the bottleneck of the system it turns out in practice that the majority of data are kept locally, which are in addition the most frequently accessed data.

From the above conditions we see how an OODBS can be used as communication interface between the divisions of a company. Hence from our initial list of requirements there is one condition left: the communication interface should allow restructuring of transmitted data and extraction of partial information.

To realize the latter condition the object algebra associated with the data model has to provide special operations for restructuring data. Examples are operators for nesting or unnesting attributes, or for the insertion of set or tuple constructors. These operators must be information preserving in the sense that though data are transmitted in certain views, the original data are not changed. Thus there are two ways of data selection: If the user is allowed to access only part of the information, data protection mechanisms can restrict the possible views. On the other hand, the receiver can restrict and restructure the allowed data intentionally and as he need them for his application; this is provided by the operators from the object algebra.

In complex applications not only the pure data are of interest but also information about these data, such as their structure, type, etc. Hence it should be possible for the user to access the so-called meta-information. If the same data model is applied for the meta-information the operations of the object algebra and hence the same query languages based on them can be applied to get this type of information. There are only few systems where the meta-information is organized in this way. Details can for example be found in [Goe93] or in [GH93].

We thus showed that the last requirement of Section 2 can easily be realized in an appropriate OODBS.

5 Realization of an Integrated Database System

An integration concept as described above has been developed at the Technical University Clausthal. The object-oriented database system OSCAR [HFW90] is based on the data model EXTREM [HH91] together with an object algebra that includes restructuring mechanisms as mentioned at the end of Section 4. EXTREM was also used to model the meta-scheme of the OODBS [GH93, Goe93]. The meta scheme allows the modeling of a different applications in one single database, in particular the data of the different CIM areas such as PPS, CAD, CAP, etc.

The integration concept was verified for the area of CAD. The CAD system (product model) CHARM [EGH92] developed at the TU Clausthal was used to design objects which were then managed in OSCAR. A general test with realistic CIM data from all divisions is due to further investigations. Clearly, the efficiency of our approach can only be evaluated on the basis of data from real systems.

In this paper we have demonstrated that an object-oriented database system can be advantageously used to manage information in CIM in a consistent and integrated way, and how the various requirements for CIM are being handled.

References

EGH92 K. H. Ecker, J. Göers, R. Hirschberg, CHARM II - An alternative approach for geometric modeling. In F.-L. Krause, D. Ruland, H. Jansen (eds.): *CAD '92*, Springer-Verlag, 1992.

GH93 J. Goeers, A. Heuer, Definition and Application of metaclasses in an object-oriented database model. In *Proceedings of the 9th International Conference on Data Engineering*, Vienna, April 1993.

Goe93 J. Göers, *Ein neues Konzept zur integrierten CIM-Informationsverwaltung und seine Realisierung am Beispiel eines CAD- und Objektbanksystems*. Doct. Thesis, TU Clausthal, 1993.

HH91 C. Hörner, A. Heuer, EXTREM - the structural part of an object-oriented database model, Technical Report 91/5, Institut fuer Informatik, TU Clausthal, 1991.

HFW90 A. Heuer, J. Fuchs, U. Wiebking, OSCAR: The design of an object-oriented database system with a nested relational kernel, In *Proceeding of the 9th International Conference on Entity-Relationship Approach*, Lausanne, 1990.

Sch90 A.-W. Scheer, Der computergesteuerte Industriebetrieb, Springer-Verlag, Berlin Heidelberg, 1990.

RG91 D. Ruland, H. Gotthard, Entwicklung von CIM-Systemen mit Datenbankeinsatz. Hanser-Verlag, München, 1991.