

Maßgeschneiderte CAx-Systeme auf Basis von Komponenten

Dipl.-Ing. B. Swieneczek, Dipl.-Inform. F. Arnold, Dipl.-Ing. Th. Kilb, Dipl.-Ing. A. Janocha
anica@mv.uni-kl.de

Lehrstuhl für Rechneranwendung in der Konstruktion,
Fachbereich Maschinenbau und Verfahrenstechnik, Universität Kaiserslautern

1 Stand der Technik

Der Trend der letzten Jahre im CAx-Bereich geht eindeutig in Richtung 3D-Modellierung. Der Einsatz dieser Technologie ist jedoch erst dann wirtschaftlich sinnvoll, wenn die generierten Daten nicht ausschließlich als Ersatz für 2D-Zeichnungen dienen, sondern während des gesamten Produktentstehungsprozesses eingesetzt werden und auf diese Weise Datendurchgängigkeit gewährleistet wird. Mittlerweile wird ein umfangreiches Spektrum von Anwendungen eingesetzt. Beispielfhaft seien hier Berechnungs- und Simulationsprogramme oder die 3D-Produktvisualisierung in nicht-technischen Bereichen (z. B. Marketing, Vertrieb) genannt.

Viele CA-Systeme bieten zwar eine große Auswahl an Modulen für nahezu alle Bereiche der Produktentwicklung, allerdings ist kein System, unabhängig von seiner Komplexität, in der Lage, alle Anforderungen seiner Anwender zu erfüllen. Deshalb kommen in immer größerem Umfang spezielle Programme für individuelle Probleme zum Einsatz.

Der Anwender sieht sich jedoch mit Schwierigkeiten konfrontiert, wenn er versucht, für spezielle Probleme spezielle Anwendungen unterschiedlicher Systemhersteller einzusetzen. Um die Integration der verschiedenen Programme zu ermöglichen, muß er sich auf neutrale Standardschnittstellen für den Produktdatenaustausch (IGES, VDAFS, STEP) verlassen, wobei hier mit Informationsverlusten zu rechnen ist. Außerdem muß er sich mit differierenden Benutzerführungen vertraut machen.

Im Bewußtsein dieser Probleme entwickelte die Arbeitsgruppe "CAD/CAM-Strategien der deutschen Automobilindustrie" einen Vorschlag für eine offene CAx-Systemarchitektur /1/, /2/, /3/. Diese sollte in der Lage sein, alle CAx-Komponenten, die im Laufe des Produktentstehungsprozesses verwendet werden, zu integrieren. Es sollte unter anderem die folgenden Anforderungen erfüllen:

- Offenheit
- Interoperabilität
- Investitionssicherheit
- Aufhebung der Zwangsbindung des Anwenders an einen Systemhersteller
- Vermeidung redundanter Systeme

Die Berücksichtigung der internationalen Standards STEP für den Bereich der Produktdatenmodellierung und CORBA für den Bereich der verteilten objektorientierten Systeme, die in den folgenden Abschnitten kurz dargestellt sind, war für die Erfüllung dieser Anforderungen eine wichtige Voraussetzung /5/, /6/.

2 Grundlagen

2.1 STEP

1984 begann die International Organization for Standardization (ISO) die Entwicklung von STEP mit folgendem Ziel:

Eine eindeutige Repräsentation von computer-interpretierbaren Produktinformationen für den gesamten Lebenszyklus eines Produktes [ISO-10303-1-93]

STEP ist primär die Beschreibung eines Produktdatenmodells. Der STEP-Standard enthält eine Anzahl von Regeln nicht ausschließlich für die Modellierung von Produktdaten sondern auch Beschreibungsmethoden, Implementierungsmethoden sowie Methoden zur Prüfung der Konformität von Implementierungen.

Die Beschreibungssprache für STEP Informationsmodelle ist EXPRESS (ISO 10303-11). EXPRESS erlaubt die Beschreibung des Datenmodells mit Entitäten einschließlich Attributen und Beziehungen zwischen diesen.

Die Anforderungen an Produktdatenmodelle hängen stark vom industriellen Einsatz in speziellen Applikationen ab. Aus diesem Grund werden Produktdatenmodelle für verschiedene Anwendungsgebiete, die sogenannten Applikationsprotokolle (AP), definiert.

Um die STEP-Philosophie der Automobilindustrie im Bereich Mechanik zu konkretisieren, wurde das Applikationsprotokoll AP 214 "Core Data for Automotive Mechanical Design Processes" definiert. AP 214 beinhaltet geometrische und nicht-geometrische Daten von Bauteilen wie auch Produktdaten mechanischer Bauteile und Baugruppen in der Automobilentwicklung.

2.2 Common Object Request Broker Architecture (CORBA)

Ein weiterer für die Realisierung der CAX-Architektur verwendeter internationaler Standard ist CORBA. Es ist ein von der Object Management Group (OMG) definierter Standard für interoperable Softwaresysteme unter Verwendung von objektorientierter Technologie.

Die zentrale Komponente der Spezifikation ist der Object Request Broker (ORB). Der ORB ist die Software (eine sog. Middleware), die zwischen den Applikationen im Netzwerk vermittelt. Er verwaltet die Interaktion zwischen verschiedenen heterogenen Applikationen und ermöglicht die Kommunikation zwischen Objekten über Plattformgrenzen hinweg. Durch die Verwendung eines ORB kann ein Client auf transparente Weise Methoden von Serverobjekten aufrufen. Diese Methoden können beispielsweise Objekte erzeugen, manipulieren oder löschen. Hierbei muß der Client keine Kenntnis darüber haben, wo das aktuelle Objekt sich befindet. Außerdem bleiben dem Client Details wie Programmiersprache, Betriebssystem oder andere Aspekte der Serverimplementierung verborgen. Dadurch ermöglicht der ORB die Interoperabilität zwischen Applikation in heterogenen verteilten Umgebungen und sorgt für eine nahtlose Verbindung von mehreren Objektsystemen.

Für die Definition der Schnittstellen der Objekte innerhalb einer CORBA-Anwendung wurde mit der *Interface Definition Language* (IDL) eine spezielle Notation als Teil des CORBA-Standards

entwickelt. Aus einer mit Hilfe von IDL beschriebenen Schnittstelle lassen sich Implementierungsvorlagen für verschiedene Programmiersprachen ableiten, z. B. C, C++ oder Java.

ORBs erlauben auch die Integration bestehender Systeme. In einer ORB-basierten Applikation wird das bestehende System quasi "eingepackt", d. h. eine die bestehende Anwendung umschließende Software übersetzt zwischen der standardisierten Schnittstelle und der Schnittstelle der Anwendung. Diese Vorgehensweise wird im Softwareengineering als "wrapping" bezeichnet, die dabei entstehende Software als "wrapper" oder Adapter wie in Bild 1 zu sehen ist /9/.

CORBA hat mittlerweile eine hohe Akzeptanz erreicht und wird von zahlreichen kommerziellen Produkten unterstützt. Es existieren unterschiedliche Implementierungen des abstrakten CORBA-Standards verschiedener Hersteller, die ab CORBA 2.0 auch untereinander kommunizieren können.

3 Der CAX-Objektbus

Der CAX-Objektbus /4/ wurde als Basis für eine offene CAX-Systemarchitektur entwickelt. Als Kommunikationsschnittstelle für verteilte, objekt-orientierte CAX-Systeme ist er in der Lage, den Zugriff auf CAX-Objekte, also Produktdaten und die zugehörigen Funktionalitäten, über Plattform- und Herstellergrenzen hinweg zu realisieren (siehe Bild 1).

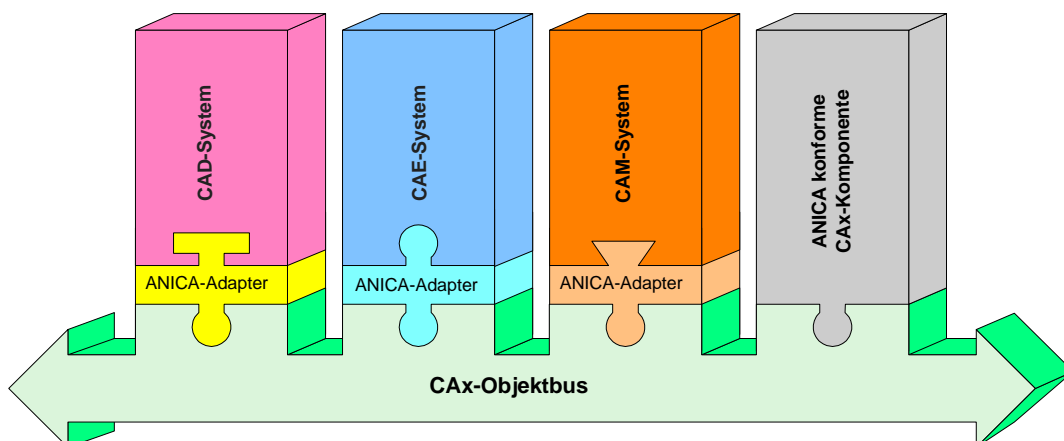


Bild 1: ANICA als Vorstufe zu CAX-Komponentensystemen

Der Grund für die Verwendung der internationalen Standards STEP und CORBA für die Realisierung des CAX-Objektbusses liegt darin, daß nahezu alle Anforderungen in den Bereichen Produktdatenmodellierung und Systemkommunikation erfüllt werden sowie in der Steigerung der Akzeptanz dieser Kommunikationsplattform bei Anwendern und Systemherstellern durch die weite Verbreitung dieser Standards. Eine weitere wichtige Voraussetzung für die Definition des CAX-Objektbusses war ein im Projekt ANICA (ANalysis of access Interfaces of various CAX systems, vgl. /9/) durchgeführter Vergleich verschiedener Programmierschnittstellen von CAD-Systemen und CAD-Kernen (u. a. CATIA, Pro/Engineer, ACIS, CAS.CADE). Hieraus wurde eine gemeinsame Menge von Funktionen abgeleitet, die auch durch den CAX-Objektbus unterstützt werden soll.

Die Schnittstelle des CAX-Objektbusses erlaubt den Zugriff auf CAX-Klassen, die ihrerseits Daten und die dazugehörigen Methoden umfassen. Während die Daten eine Teilmenge der Entitäten

von STEP AP 214 darstellen werden die Methoden aus der vergleichenden Analyse der Systeme verwendet. Diese CAX-Klassen werden in IDL beschrieben und stehen dann den Programmierern von CAX- Komponenten zur Verfügung.

Damit bestehende CAX-Systeme den CAX-Objektbus verwenden können, müssen Adapter (wrapper) implementiert werden, die die Umsetzung von Funktionsaufrufen und Daten der Programmierschnittstellen (API: Application Programming Interface) zu den CAX-Objekten übernehmen. Diese werden in Bild 1 als ANICA-Adapter bezeichnet. Diese Adapter werden für ANICA-konforme CAX- Komponenten nicht mehr benötigt.

Von entscheidender Bedeutung an diesem neuen Kommunikationsansatz ist der Verzicht auf jegliche Form des dateibasierten Datenaustausches. Die Kooperation verschiedener Komponenten wird ausschließlich online und ohne das mehrfache Speichern von Daten realisiert.

4 Der CAX-Objektbus in der Praxis

Grundsätzlich lassen sich drei verschiedene Strategien beim Einsatz des CAX-Busses unterscheiden. Diese sind von der Verfügbarkeit spezialisierter CAX-Komponenten abhängig (siehe Bild 2).

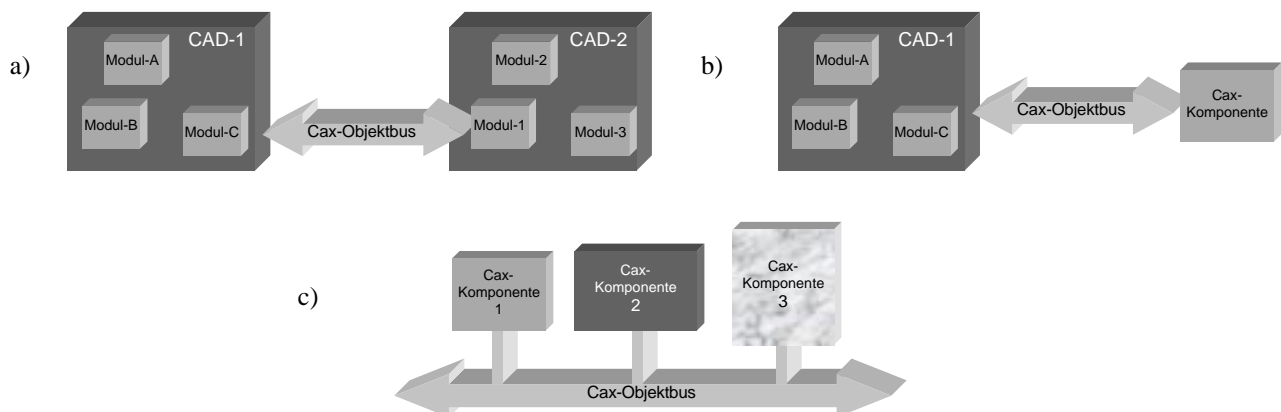


Bild 2 : Verschiedene Anwendungen des CAX-Objektbusses

In Variante (a) werden noch keine CAX-Komponenten eingesetzt, sondern es werden bestehende CAX-Systeme miteinander verbunden. Dadurch können Daten und Funktionen eines Systems dem Anwender des anderen Systems zugänglich gemacht werden. Erkennbar ist der modulare Aufbau der beiden CAD-Systeme, wobei in diesem speziellen Fall nur ein einziges Modul von CAD-2 an das System CAD-1 angebunden wird. Bei dieser Lösung sind lediglich Investitionen für die zusätzlichen ANICA-Adapter erforderlich. Für den Anwender ergibt sich der Vorteil, daß er zur Nutzung von "fremden" Funktionen nicht die Bedienung eines weiteren CAX-Systems erlernen muß, sondern mit seiner gewohnten Benutzerumgebung zusätzliche Funktionen aufrufen kann.

Lösung (b) stellt die Anbindung von spezialisierten CAX-Komponenten, die von Anfang an mit der CAX-Bus-Schnittstelle ausgestattet sind, an bestehende CAX-Systeme dar. Durch den CAX-Objektbus ist gewährleistet, daß diese Komponenten plattform- und herstellerübergreifend

verwendet werden können. Diese Lösung eignet sich beispielsweise zur Einbindung von Simulationslösungen (z. B. Kinematik, Ergonomiesimulation mit Menschmodellen) oder Berechnungsverfahren (z. B. FEM, CFD).

Der Anwendungsfall (c) stellt die konsequenteste Umsetzung der ANICA-Architektur dar (vgl. /7/ /8/). Hier werden ausschließlich CAx-Komponenten miteinander verbunden. Der Rahmen um die einzelnen Komponenten stellt die temporäre Verknüpfung der Komponenten für eine spezifische Anwendung dar. Durch die hohe Flexibilität und die Auswahl der am besten geeigneten Komponenten kann mit dieser Lösung das Rationalisierungspotential des neuen Ansatzes am besten ausgeschöpft werden.

5 Ausblick

Die Summe der Vorteile von komponentenbasierten CAx-Systemen wird für den Anwender erst in vollem Umfang realisierbar, wenn marktgängige Systeme diesen Ansatz übernehmen. Bild 1 zeigt jedoch, daß auch eine Kombination der im letzten Abschnitt dargestellten Anwendungsmöglichkeiten der Systemarchitektur möglich sind. So könnten zunächst vorhandene Komplettsysteme über den CAx-Objektbus miteinander verknüpft werden und, sobald diese verfügbar sind, ANICA-konforme CAx-Komponenten hinzugefügt werden. Anschließend können dann Komplettsysteme durch mehrere CAx-Komponenten substituiert werden.

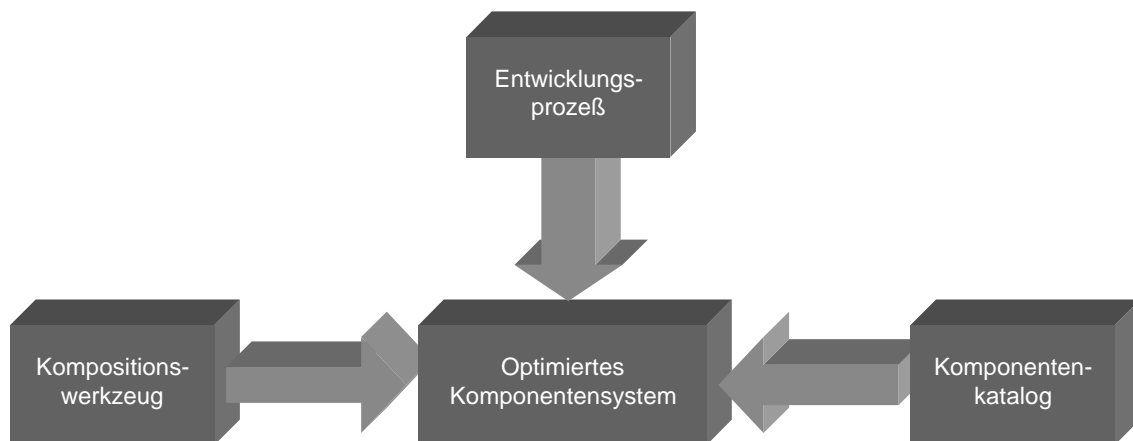


Bild 2: Generierung von maßgeschneiderten CAx-Komponentensystemen

Insgesamt kann diese Entwicklung dazu führen, daß Anwender spezialisierte Komponenten verschiedener Hersteller zu einem individuell angepaßten System zusammenfügen. Dies vermindert die Abhängigkeit von einzelnen Systemherstellern. Wie in Bild 3 dargestellt, bestimmen Parameter des Entwicklungsprozesses die Konfiguration des Komponentensystems. Mit Hilfe eines Kompositionswerkzeuges werden verschiedene Komponenten, die zuvor aus dem Komponentenkatalog ausgewählt wurden, miteinander verknüpft. Falls durch variierende Prozeßparameter Änderungen erforderlich werden, ist der Austausch einzelner Komponenten mit sehr geringem Aufwand möglich.

Auf diese Weise entstehen schlanke Systeme, die in jedem funktionalen Teilbereich mit "best of class"-Anwendungen verschiedener Hersteller arbeiten. Eine einheitliche Benutzerführung bzw. Be-

nutzeroberfläche, die um die jeweiligen Bedienungselemente für Spezialfunktionen ergänzt wird, vermindert den Einarbeitungsaufwand gegenüber mehreren Komplettsystemen.

6 Literatur

- /1/ Dankwort C. W.: CAx Systems Architecture of the Future. In: Roller D.; Brunet P. (Eds.): CAD Systems Development. Tools and Methods, Springer-Verlag, 1997.
- /2/ Dankwort C. W.; Janocha A. T.: Von Monolithen zu Komponenten: CAx-Architekturen im Wandel. In: D. Ruland (Hrsg.): Tagungsband CAD 96. Verteilte und intelligente CAD-Systeme, Kaiserslautern, März 1996.
- /3/ Dankwort C. W.; Kellner P.; Leu D.; Müller G.; Renz W.; Weißberger G.: CAD in der Automobilindustrie - Aktuelle Bedürfnisse und mittelfristige Ziele. CAD-CAM REPORT Nr. 1, 1997.
- /4/ Iselborn B.: Entwurf eines CAx-Objektbusses. CAD '98: Tele-CAD. Produktentwicklung in Netzwerken. Darmstadt, 5.-6. März 1998.
- /5/ Janocha A. T.: CAx-Systemintegration auf Basis von CORBA und STEP. Produktdaten Journal, Nr.1, Juni 1996, S. 45-48.
- /6/ Janocha A. T., Arnold F.: "Architekturkonzepte integrierter CAx-Systeme", in: Proceedings of the International Conference on Computer Integrated Manufacturing (CIM '96), Zakopane, Poland, May 14 - 17, 1996, Vol III, pp. 183-190.
- /7/ Janocha A. T., Dankwort C.W.: Moderne CAx-Systemarchitekturen: Von proprietären Systemen zu Komponenteware. CAD-CAM REPORT Nr. 9, September 1997.
- /8/ Janocha A. T., Gandyra M.: Richtungsweisender Einsatz der Komponententechnologie für CAx- Systeme. VDI-Berichte 1357: Neue Generation von CAD/CAM-Systemen - Erfüllte und enttäuschte Erwartungen, S. 87-104.
- /9/ Arnold F., Janocha A. T., Swienczek B., Kilb T.: " Die CAx-Integrationsarchitektur ANICA und ihre erste Umsetzung in die Praxis", erscheint in: Tagungsband zum Workshop "Integration heterogener Softwaresysteme" im Rahmen der 28. GI-Jahrestagung Informatik '98 - Informatik zwischen Bild und Sprache, Magdeburg, 21. September 1998