

# Modellierung von Entwurfsumgebungen und deren Abbildung in ein Datenbank-integriertes Ingenieursystem

Bernd Sutter<sup>1</sup>

R+V Allgemeine Versicherung

6200 Wiesbaden

**Korrekturadresse: Bernd Sutter, Steingasse 23, 6200 Wiesbaden**

## Zusammenfassung

Rechnergestützte Integration in technischen Entwurfsanwendungen bedeutet eine für den Entwerfer durchgängige Unterstützung während des gesamten Entwurfsablaufs. In diesem Beitrag wird der Aufbau eines rechnergestützten **Entwurfsumgebungsmodells** vorgestellt, in dem die relevanten Elemente der Entwurfsumgebung berücksichtigt und deren wechselseitigen Beziehungen reflektiert werden, so daß damit die Grundlage für eine durchgängige Unterstützung bereitgestellt wird. Das Entwurfsumgebungsmodell wird in mehrere Teilmodelle untergliedert, in denen die Entwurfsobjekte, die Entwurfsmethodiken, die Entwurfswerkzeuge und die Entwerfer ebenso dargestellt werden wie der dynamische Entwurfsprozeß bzw. die dynamischen Wechselwirkungen zwischen den Entwerfern bei einer kooperativen Entwurfsvorgehensweise. Zur systemseitigen Abbildung wird ein **DB-integriertes Ingenieursystem** vorgeschlagen und eine mögliche Systemarchitektur vorgestellt.

## Schlüsselwörter:

Entwurfsumgebungsmodell, Entwurfsobjekte, Aufgabenspezifikation, Versionen, Entwurfsprozeß, Entwurfskooperation, Workstation/Server-Datenbanksystem, integriertes Ingenieursystem

## Abstract

Computer-oriented integration in technical design applications means a continuous support of the designer during the whole design process. In this paper, we present the structure of a computer-oriented model of the design-environment which considers all significant elements of a design environment as well as the relationships among them. The design-environment model is divided into several sub models representing the design objects, the design methodologies, the design tools, or the designers themselves as well as the dynamic design process and the cooperative interactions among the designers. A database integrated engineering system realizing the design-environment model is shown and a feasible system architecture is presented.

## Key words:

design-environment model, design objects, task specification, versions, design process, design cooperation, workstation/server database system, integrated engineering system

---

1. Die Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter in der Arbeitsgruppe Datenverwaltungssysteme von Prof. Th. Härder an der Universität Kaiserslautern.

# 1 Einleitung

Eine durchgängige, rechnerseitige Unterstützung der Arbeit von Ingenieuren in den technischen Anwendungsbereichen, wie etwa dem Maschinenbau, der Architektur, dem Bauwesen, der Anlagenplanung, der Elektronik oder dem noch wesentlich jüngeren Bereich des VLSI-Chip-Entwurfs, ist eine der großen Herausforderungen an die Informatik der Gegenwart.

Der Prozeß des Entwurfs und der Fertigung eines Produktes in den genannten Anwendungsbereichen wurde in den letzten Jahrzehnten zunehmend durch den Einsatz von Rechnern beeinflusst und verändert. Zunächst wurden in administrativen und betriebswirtschaftlichen Bereichen Rechner zur Unterstützung bei der Bearbeitung der anfallenden Aufgaben eingesetzt, was einen signifikanten Rationalisierungseffekt in den verschiedenen dispositiven Aufgabenbereichen (Materialwirtschaft, Produktionsplanung, Produktionssteuerung usw.) bedeutete. Eine wesentliche Anforderung an die Rechnerunterstützung bestand in der effizienten Verwaltung der in diesem Bereich anfallenden Informationsmengen. Die Entwicklung von Datenbankverwaltungssystemen (DBS), die sich zur gleichen Zeit vollzog, wurde dabei in erheblichem Maße von den Anforderungen aus diesen Planungsbereichen mitbestimmt, so daß die in der Folge entwickelten bekannten Datenmodelle (hierarchisches, netzwerkartiges und relationales Modell) und die kommerziell verfügbaren DBS eine geeignete Grundlage für diese Planungsbereiche bildeten.

Die Rechnerunterstützung in den übrigen Arbeitsbereichen der Konstruktion und der Fertigung war zunächst weit weniger ausgeprägt. Durch den Einsatz von CAD-Systemen (Computer Aided Design, CAD) wird ein **Produktmodell** für ein Entwurfsobjekt aufgebaut, das die Grundlage für die weiteren Fertigungsschritte bilden soll. Die gegenwärtige Situation ist jedoch weitgehend durch isolierte Systemlösungen bestimmt, die jeweils ihr eigenes Produktmodell verwalten. Eine datenseitige Verknüpfung zwischen den CAD-Systemkomponenten ist meist nur über standardisierte Austauschformate möglich (z.B. [18, 30, 4]), was i. allg. mit einem z. T. erheblichen Informationsverlust verbunden ist.

Eine logische Konsequenz des weiterführenden Rechnereinsatzes sind die Bemühungen um eine Integration der rechnerunterstützten Teilfunktionen zu einem rechnerintegrierten Gesamtsystem (vgl. z.B. [7, 8, 9, 23, 27]). Um dieses Ziel zu erreichen, ist neben der Erstellung eines geeigneten Produktmodells eine rechnerseitige Verwaltung von weiteren im Entwurfsvorgang relevanten Entwurfs-elementen notwendig. Vor allem im VLSI-Entwurfsbereich sind unter dem Begriff **CAD-Frameworks** entsprechende Ansätze zu finden (z.B. das NELSI-CAD-Framework [32], das NMP-CADLAB-Framework [15] oder die CAD-Framework-Initiative [5, 6]). Die Aufgabe eines CAD-Frameworks ist das Bereitstellen eines Software-Rahmens, der allgemeingültige Funktionen anbietet (z.B. Datenmanagement, Benutzer-Schnittstelle), so daß die CAD-Werkzeuge, mit denen die eigentliche Entwurfsaufgabe durchgeführt wird, in ein CAD-Framework eingepaßt werden und auf die angebotene Funktionalität aufsetzen können. In [13] wird ein CAD-Framework als die Zusammenfassung aller Funktionen betrachtet, die zur Un-

terstützung des CAD-Tool-Entwicklers, des CAD-Systemintegrierers und des Entwerfers (Endbenutzers) notwendig sind.

In den meisten Framework-Ansätzen wird die Notwendigkeit, ein umfassendes Modell zur Beschreibung einer Entwurfsumgebung aufzubauen, vernachlässigt. Um das Ziel einer durchgängigen rechnerseitigen Unterstützung von ingenieurwissenschaftlichen Entwurfsumgebungen zu erreichen, ist jedoch die Entwicklung eines **rechnerinternen Entwurfsumgebungsmodells (RIEM)** notwendig, in dem alle relevanten Informationsstrukturen einer Entwurfsumgebung erfaßt und ihre Wechselwirkungen beschrieben werden (vgl. auch [2]). Das Produktmodell zur Repräsentation der Entwurfsobjekte ist hierbei ein wesentlicher Bestandteil des Entwurfsumgebungsmodells.

Zur systemseitigen Abbildung eines Entwurfsumgebungsmodells wird in diesem Beitrag die Konzeption und Entwicklung eines **DB-integrierten Ingenieursystems** vorgeschlagen, das an der Benutzerschnittstelle die für den Entwurfsablauf notwendige Funktionalität zur Verfügung zu stellen hat. Die nachfolgenden Ausführungen beziehen in das Entwurfsumgebungsmodell zunächst nur die Entwurfsphase ein; sie sind aber prinzipiell auf die nachfolgenden Entwicklungsphasen erweiterbar.

Um in integrierten Ingenieursystemen eine möglichst ganzheitliche Modellierung der Daten der verschiedenen Entwurfsaspekte (also sowohl der Entwurfsdaten als auch der übrigen Entwurfsumgebungsinformationen) und ein hohes Maß an Datenintegration zu erreichen, ist der Einsatz von Datenbanktechnologien unumgänglich. Eine zentrale Herausforderung der gegenwärtigen Datenbankforschung besteht daher in der Erarbeitung und der Nutzbarmachung von Datenbankkonzepten für technische Anwendungen.

In diesem Beitrag werden zunächst die verschiedenen Elemente, die in einer Entwurfsumgebung von Bedeutung sind, herausgearbeitet und deren Zusammenwirken verdeutlicht. Es werden verschiedene **Teilmodelle** zur abstrakten, anwendungsbereich-unabhängigen Repräsentation einer Entwurfsumgebung unterschieden und der prinzipielle Aufbau eines rechnergestützten Entwurfsumgebungsmodells beschrieben. Hauptsächlich wird auf Modellierungsaspekte zur Darstellung der Entwurfsobjekte und des gesamten, dynamisch ablaufenden Entwurfsprozesses. Dabei ist zu klären, wie das koordinierte und zusammenwirkende Arbeiten mehrerer Entwerfer innerhalb eines Entwurfsprozesses geeignet repräsentiert werden kann. Der letzte Teil beschäftigt sich mit allgemeinen Abbildungsfragen eines DB-integrierten Ingenieursystems. Hierbei wird auch auf die Konzeption und Architektur eines strukturell objektorientierten Workstation/Server-Datenbanksystems in einer arbeitsplatzorientierten Ablaufumgebung eingegangen, das eine geeignete Grundlage für die Handhabung, die Organisation und die Modellierung der im Verlauf des gesamten Entwurfsvorgangs anfallenden Informationsstrukturen bietet.

## 2 Ein Modell zur anwendungsunabhängigen Beschreibung von Entwurfsumgebungen

Zum besseren Verständnis der im folgenden eingeführten Modellelemente einer Entwurfsumgebung wird an dieser Stelle ein Anwendungsbeispiel aus dem mechanischen Konstruktionsbereich eingeführt, auf das in den nachfolgenden Kapiteln zurückgegriffen wird. Ausgangspunkt des Szenarios ist der Auftrag an den Chefkonstrukteur der Planungsabteilung, ein Getriebe zu entwerfen. Dazu wird ihm eine Aufgabenstellung für das Getriebe vorgegeben, in der die Antriebs- und die Abtriebsdrehzahl sowie die Abmessungen des Getriebes spezifiziert sind (vgl. Bild 1). Zusätzlich wird gefordert, daß am Ende des Entwurfsauftrags ein Simulationsprogramm auf dem Getriebeentwurf angewendet wird und der Chefkonstrukteur den Getriebeentwurf freigibt.

Der Chefkonstrukteur erstellt zunächst einen Getriebeplan für das Entwurfsobjekt Getriebe  $G_{xy}$ . Dabei legt er fest, daß das Getriebe aus zwei Getriebestufen aufzubauen ist, einer Stirnradgetriebestufe (GS1) und einer Planetenradgetriebestufe (GS2), zwischen denen eine Verbindungswelle (VW1) eingesetzt wird. Er verfeinert mit Unterstützung eines CAD-Systems entsprechend die Baugruppenstruktur des Getriebes und legt die Aufgabenstellung der Entwurfsobjekte fest, was u.a. die Vorgabe von Anschlußpunkten zwischen den Getriebestufen und der Verbindungswelle zur Folge hat. Schließlich sollen die beiden Getriebestufen sowie die Verbindungswelle durch je einen Konstrukteur seiner Abteilung weiter ausgearbeitet werden. Dazu definiert er drei Aufträge, denen er je einen Konstrukteur zuordnet. Die Auftragspezifikation ist jeweils mit der Spezifikation der Entwurfsobjekte identisch.

Der hier vorgestellte Modellansatz unterscheidet zwischen einem **entwurfsbereich-unabhängigen Entwurfsumgebungsmodell**, das eine allgemein (also für verschiedene Entwurfsbereiche) verwendbare Basis zur Beschreibung von Entwurfsumgebungen darstellt, und den **anwendungsspezifischen Modellen**, die zur Darstellung einer konkreten Entwurfsumgebung (in einem Entwurfsbereich) durch eine Konkretisierung bzw. "Instanziierung" aus dem entwurfsbereich-unabhängigen Entwurfsumgebungsmodell abgeleitet werden.

Eine nähere Betrachtung der verschiedenen ingenieurwissenschaftlichen Entwurfsbereiche zeigt, daß zur Repräsentation einer Entwurfsumgebung in dem entwurfsbereich-unabhängigen Entwurfsumgebungsmodell **sechs wichtige Teilmodelle** zu unterscheiden sind [29], die jeweils für die Darstellung von dedizierten Entwurfsaspekten verantwortlich sind:

- **Entwurfsobjekt-Modell**

Im Entwurfsobjekt-Modell werden die Entwurfsobjekte repräsentiert, d.h., es entspricht dem Produktmodell eines Entwurfsobjektes. In dem hier vorgestellten Modellansatz wird jedoch von einer detaillierten Repräsentation der primären Entwurfsdaten abstrahiert. Es stehen die strukturellen Beziehungen zwischen den Entwurfsdaten im Blickfeld (z. B. deren hierarchische Strukturierung).

- **Entwurfsphasen-Modell**

Das Entwurfsphasen-Modell übernimmt die Aufgabe, Entwurfsphasen und Entwurfsabläufe bzw. Entwurfsmethodiken zu beschreiben. Zunächst sind die Entwurfsphasen, in die der Gesamtentwurf unterteilt ist, zu identifizieren und deren Abfolge zu spezifizieren (vgl. z.B. [31]). Innerhalb einer Entwurfsphase können ebenfalls feste Ablaufstrukturen identifiziert werden, die abhängig von der konkreten Entwurfsaufgabe (also dem zu erstellenden Entwurfsobjekt) die durchzuführenden Teilschritte und deren Abfolge definieren.

- **Entwurfswerkzeug-Modell**

In dem Entwurfswerkzeug-Modell sind sämtliche rechnergestützten Werkzeuge der Entwurfsumgebung zu spezifizieren. Für jedes Werkzeug sind u.a. die Art der Werkzeug-Integration, die konkrete Ablaufumgebung (Hardware- und Software-Umgebung, Aufrufaktivierung usw.) und die Datenversorgung (Ein- und Ausgabedaten sowie deren Datenrepräsentation) anzugeben.

- **Entwurfssubjekt-Modell**

Dieses Modell umfaßt die Personen (Ingenieure, Konstrukteure, Arbeitsplaner etc.), die für die Entwurfsumgebung von Interesse sind (vgl. [26]).

Mit den bisher eingeführten vier Teilmodellen werden die verschiedenen Elemente modelliert, die eine umfassende Beschreibung einer Entwurfsumgebung ermöglichen. Mit diesen Teilmodellen ist es jedoch bislang nicht möglich, den konkreten Prozeß eines Entwurfsvorgangs systemseitig abzubilden<sup>2</sup>. Zur Darstellung des dynamisch ablaufenden Entwurfsprozesses werden im folgenden zwei weitere Teilmodelle eingeführt, das Entwurfsprozeß- und das Entwurfskooperations-Modell.

- **Entwurfsprozeß-Modell**

In einem rechnergestützten Modell einer Entwurfsumgebung ist eine möglichst vollständige Erfassung des Entwurfsprozesses, also des dynamischen Entwurfsvorgangs, zu berücksichtigen [3, 24]. Das Entwurfsprozeß-Modell stellt dazu Ablaufeinheiten bereit, die eine systemseitige Abbildung des konkreten dynamischen Entwurfsprozesses sicherstellen. Damit lassen sich die in betrieblichen Abläufen vorzufindenden Auftragsstrukturen und die Bearbeitung von Aufträgen durch die beteiligten Ingenieure modellieren. Das Entwurfsprozeß-Modell steht mit allen fünf weiteren genannten Teilmodellen in Beziehung (vgl. Bild 6).

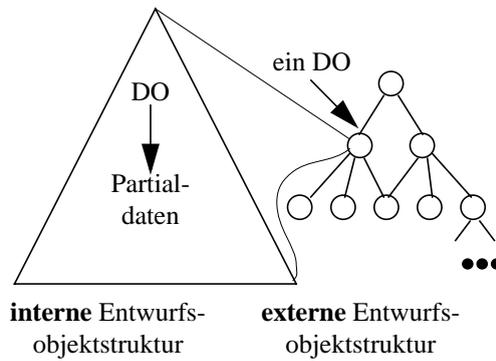
- **Entwurfskooperations-Modell**

Die Erledigung einer komplexen Entwurfsaufgabe wird i. allg. von mehreren Entwerfern (in einer oder verschiedenen betrieblichen Organisationseinheiten) durchgeführt. Jeder Entwerfer hat eine oder mehrere Teilaufgaben auszuführen, die alle durch ein übergeordnetes gemeinsames Ziel verbunden sind, nämlich das Erfüllen der gesamten komplexen Entwurfs-

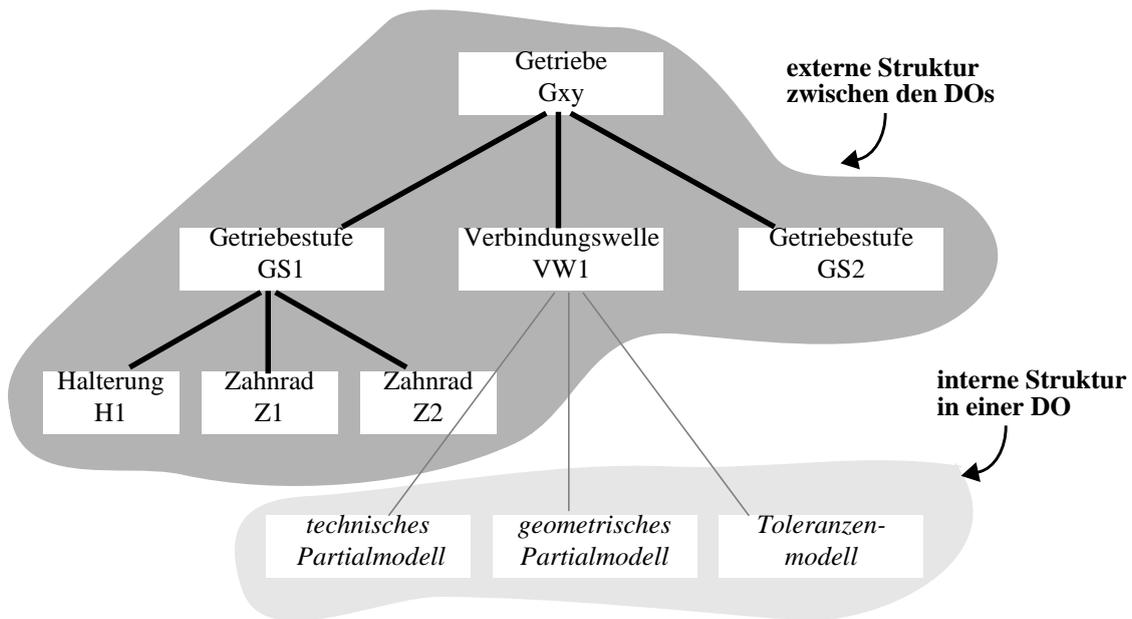
---

2. Das Entwurfsphasen-Modell beinhaltet Entwurfsmethodiken, die lediglich eine Vorgabe für den Ablauf des Entwurfsprozesses darstellen!





a) externe und interne Entwurfsobjektstruktur



b) Beispiel für eine konkrete Entwurfsobjektstruktur in einem mechanischen Entwurfsbereich

Bild 7: Das Entwurfsobjektmodell und ein Beispiel aus dem mechanischen Entwurfsbereich bezeichnet. Die **externe Struktur** ist durch die hierarchische Zerlegung komplexer Entwurfsobjekte innerhalb des Entwurfsvorgangs in Sub-Entwurfsobjekte (**Sub-EOs**) und deren Beziehungen untereinander festgelegt. Dies wird in Bild 7 verdeutlicht, in dem die Struktur für das Entwurfsobjekt Getriebe Gxy verfeinert dargestellt ist. Die externe EO-Struktur wird durch die Baugruppenstruktur zwischen den Entwurfsobjekten spezifiziert. Die interne Entwurfsobjektstruktur für das Getriebe Gxy ist durch das Toleranzmodell sowie das technische und geometrische Partialmodell festgelegt.

### 3.2 Beschreibung der Korrektheit eines Entwurfsobjektes über Features

Für ein Entwurfsobjekt sind nicht allein die primären Entwurfsdaten im Entwurfsprozeß von Interesse. Ein ebenso hoher Stellenwert ist der Konsistenz und der Korrektheit der primären Entwurfsdaten während und am Ende des Entwurfsprozesses beizumessen. Hier lassen sich drei

Anforderungsgruppen festlegen, die Einfluß auf eine Bewertung der Konsistenz und Korrektheit eines Entwurfsobjektes nehmen:

- **Spezifikation** einer Aufgabenstellung

Die Zerlegung von Entwurfsobjekten in Sub-Entwurfsobjekte bei der Top-Down-Entwurfsvorgehensweise ist immer mit der Spezifikation einer Aufgabenstellung für die Sub-Entwurfsobjekte verbunden. (z.B. Abmessung oder Anschlußpunkte, vgl. Bild .....)

- **Allgemeine Konsistenzbedingungen** auf einem Entwurfsobjekt

Für ein Entwurfsobjekt sind eine Reihe von allgemeinen Konsistenzbedingungen zu beachten, die dieses unabhängig von einer konkreten Aufgabenstellung zu erfüllen hat.

- **Zustandsinformationen** über den **Stand und Ablauf des Entwurfsprozesses** auf einem Entwurfsobjekt

Hier werden alle Kriterien zusammengefaßt, die den Zustand eines Entwurfsobjektes bzw. den zugehörigen Entwurfsprozeß charakterisieren. Beispielsweise kann die Bedingung spezifiziert sein, daß für ein Entwurfsobjekt ein vorgegebener Validierungstest erfolgreich bzw. in definierten Toleranzgrenzen durchgeführt werden muß. Des weiteren werden im Entwurfsprozeß die Entwurfsobjekte i. allg. mit expliziten Freigabemitteilungen versehen.

Die hier aufgezählten Anforderungen werden im folgenden als **Features** bezeichnet. Features sind also allgemeine Anforderungsbeschreibungen an ein Entwurfsobjekt. Die Menge sämtlicher Features, die von einem Entwurfsobjekt zu erfüllen sind, wird als **Soll-Feature-Menge** bezeichnet. Diese Menge umfaßt sowohl die Features zur Beschreibung einer konkreten Aufgabenstellung als auch die allgemeinen Konsistenzbedingungen, die erfüllt sein müssen, sowie die Features, die sich auf den Entwurfsprozeß beziehen. Die Menge der Features, die von einem bislang erstellten Entwurfsobjekt bereits erfüllt werden, wird im folgenden als **Ist-Feature-Menge** bezeichnet.

Die durch ein Feature spezifizierte Anforderung an ein Entwurfsobjekt wird in dem hier beschriebenen Modell durch ein **Prädikat** definiert. Wird das Prädikat zu wahr ausgewertet, so ist das Feature erfüllt und das Feature wird in die Ist-Feature-Menge des Entwurfsobjektes aufgenommen. Das Prädikat ist durch einen **Prädikatausdruck** definiert, in den zum einen Eigenschaften des Entwurfsobjektes in Form von **Entwurfsdaten** und zum anderen eine **Wertespezifikation** (also eine Art Gültigkeitsbereich) auf den ausgewählten Entwurfsdaten eingehen. Ein Feature muß des weiteren eine **Berechnungsfunktion** bereitstellen, die die Bestimmung der Entwurfsdaten aus dem Entwurfsobjekt durchführt. Die Berechnungsfunktion kann eine sehr einfache Selektionsfunktion sein, die den Wert einer bestimmten Objekteigenschaft (also einen Attributwert) des Entwurfsobjektes selektiert. Sie kann aber auch ein sehr komplexes Verfahren beinhalten. Die Auswertung eines Features erfolgt also in zwei Schritten: Zunächst ist die **Berechnungsfunktion** auszuführen und anschließend ist das **Prädikat** auszuwerten.

### 3.3 Versionierung von Entwurfsobjekten

Innerhalb des Entwurfsprozesses wird für ein Entwurfsobjekt  $i$ . allg. nicht sofort eine fertige Lösung bereitgestellt, vielmehr sind während des Entwurfsprozesses eine Reihe von **Versionen für ein Entwurfsobjekt** zu erstellen (Entwurfsobjekt-Version, **EOV**). Gleiches gilt für die internen Entwurfsobjektdaten. Während des Entwurfsvorgangs können beliebig viele Versionen dieser internen Entwurfsobjektdaten erstellt werden.

Die Versionierung von ganzen Entwurfsobjekten und von den internen Entwurfsobjektdaten haben unterschiedliche Bedeutung. Das Anlegen einer neuen Version auf den internen Entwurfsobjektdaten bedeutet, daß eine neue Version erzeugt und in den zugehörigen Versionsgraphen eingehängt wird. Ein detailliertes Versionsmodell für komplex strukturierte Datenobjekte ist in [19] beschrieben.

Die Versionierung eines ganzen Entwurfsobjektes ist stets mit einer **Konfiguration** verbunden. Eine Konfiguration eines Entwurfsobjekts auf der untersten Entwurfsobjektstufe (d.h., es gibt keine Sub-Entwurfsobjekte) bedeutet, daß *aus den verschiedenen Entwurfsobjektdaten "passende" Versionen zusammengebaut* werden und so eine Version (Konfiguration) des Entwurfsobjektes generiert wird. Auf einer höheren Stufe der Entwurfsobjektstruktur ist *neben der Konfiguration der internen Struktur auch die externe Entwurfsobjektstruktur zu berücksichtigen*, d.h., in die Konfiguration gehen auch Versionen der Sub-Entwurfsobjekte ein, die selbst das Ergebnis einer Konfiguration sind. In Bild 8 wird der Konfigurationsvorgang an einem Beispiel erläutert. Für das Entwurfsobjekt Getriebe werden zwei Konfigurationen ( $V_1$  und  $V_2$ ) erzeugt. Die Konfigurationen setzen Entwurfsobjektdaten (Versionen der Entwurfsobjektdaten Partial-Entwurfsobjekten) und Versionen der Sub-Entwurfsobjekte zusammen. Die Version  $V_1$  des Entwurfsobjekts oder Gxy ist aus der Version  $V_7$  der internen Entwurfsobjektdaten sowie jeweils den Versionen  $V_1$  der Sub-Entwurfsobjekte GS1, VW1 und GS2 konfiguriert. Die Version  $V_2$  des Entwurfsobjekts Gxy ergibt sich Version  $V_8$  der internen Entwurfsobjektdaten.

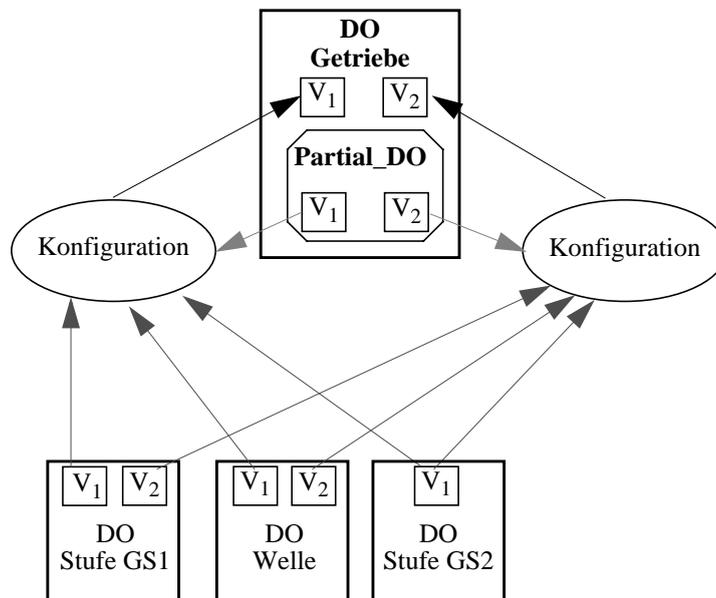


Bild 8: Beispiel einer Entwurfsobjekt-Konfiguration

Im Rahmen des gesamten Entwurfsprozesses müssen Aussagen über die **Qualität einer DOV** eines Entwurfsobjekts möglich sein. Die Qualität einer DOV läßt sich bzgl. der Aufgabenstellung oder allgemeiner bzgl. der für die DO spezifizierten Soll-Features bewerten. Das bedeutet, daß die Qualität einer DOV durch die **Ist-Feature-Menge**, die für diese DOV bestimmt wird, **festgelegt** ist. Demnach lassen sich drei ausgezeichnete Versionsarten unterscheiden:

- **nicht qualifizierte** Versionen

Für nicht qualifizierte Versionen ist die Ist-Feature-Menge noch nicht ausgewertet worden, es sind daher keine Aussagen über die Qualität der DOV möglich.

- **vorläufige** Versionen

Für vorläufige Versionen wurden die auf dem Entwurfsobjekt spezifizierten Features ausgewertet. Die Ist-Feature-Menge der DOV ist eine echte Teilmenge der Soll-Feature-Menge. Das bedeutet, daß im weiteren Entwurfsprozeß noch mindestens eine weitere, verbesserte DOV zu erstellen ist.

- **vollständige** Versionen

Eine vollständige Version erfüllt die gesamte Feature-Menge, d.h., die Ist-Feature-Menge enthält alle für das Entwurfsobjekt spezifizierten Features.

Zwei Versionen eines Entwurfsobjektes DO werden als **Alternativen** bezeichnet, wenn beiden die gleiche Ist-Feature-Menge zugeordnet ist, durch den Konfigurationsvorgang aber eine unterschiedliche Zusammensetzung festgelegt wurde.

## 4 Das Entwurfsprozeß-Modell

Zur Erfassung des Entwurfsprozesses ist das Entwurfsprozeß-Modell eingeführt worden, das geeignete Konzepte zur Repräsentation des dynamischen Entwurfsprozesses zur Verfügung zu stellen hat. Die Beschreibung eines Ablaufs geschieht i. allg. über die Einführung geeigneter Ablaufeinheiten, den **Aktivitätsträgern**, die mit bestimmten Eigenschaften versehen sind und systemseitig verwaltet werden.

### 4.1 Eigenschaften zur Charakterisierung von Aktivitätsträgern

Aktivitätsträger sind jeweils durch eine Reihe von **Merkmalen und Eigenschaften** charakterisiert sind, die sich aus dem konkreten Umfeld, in dem die Aktivitätsträger wirken, ergeben. So ist hier zunächst die Frage zu klären, welche Eigenschaften für Aktivitätsträger in Entwurfsumgebungen erforderlich sind.

- Unterstützung eines zielgerichteten Vorgehens während des Entwurfsvorgangs

In den hier betrachteten Entwurfsumgebungen ist der Entwurfsprozeß durch die hierarchische Entwurfsobjektstruktur und die Entwurfsphasen bestimmt. Die Aktivitätsträger müssen ein **zielgerichtetes Vorgehen des Entwurfsprozesses** unterstützen.

- Langlebigkeit

Da einzelne Entwurfsaktionen zum Teil sehr lange andauern können (Tage, Wochen oder Monate), müssen auch die **Aktivitätsträger langlebig** sein. Das bedeutet, daß sie stabil (d.h., ein Aktivitätsträger bleibt nach einer Systemunterbrechung erhalten) und ihre Wirkungen dauerhaft (d.h., nach einer Systemunterbrechung muß auf “Wiederaufsetzpunkte” zurückgegriffen werden, die während des Entwurfsprozesses anzulegen sind) sein müssen.

- Wechselwirkungen zwischen Aktivitätsträgern

In Entwurfsumgebungen sind unterschiedliche **Wechselwirkungen zwischen Aktivitätsträgern** gefordert. In gewissen Situationen sind **kontrollierte Wechselwirkungen** mit anderen Aktivitätsträgern gewünscht bzw. gefordert, um so ein gemeinsames, zielgerichtetes Vorgehen zu ermöglichen. In anderen Situationen ist eine **strikte Isolation** zwischen (aktiven) Aktivitätsträgern gefordert. Dies ist beispielsweise erforderlich, wenn durch einen Aktivitätsträger Datenzugriffe erfolgen, die isoliert ablaufen sollen.

- Interne Strukturierung der Aktivitätsträger

Die **interne Strukturierung** der Aktivitätsträger kann durch einen **vordefinierten Ablaufplan** fest vorgegeben sein<sup>3</sup> oder aber erst während des Ablaufs **dynamisch aufgebaut** werden.

- Konsistenzsicherungen

Die **Konsistenzsicherungen**, die während des Ablaufs bzw. bei der Beendigung eines Aktivitätsträgers zu berücksichtigen sind, beziehen sich zum einen auf die **innerhalb des Aktivitätsträgers bearbeiteten Entwurfsobjekte** und zum anderen auf den **Entwurfsablauf**.

- Geeignete Reaktionen im Fehler- und Konfliktfall

Aktivitätsträger müssen Mechanismen bereitstellen, die im **Fehler- bzw. Konfliktfall eine geeignete anwendungsorientierte Reaktion** erlauben. Hier ist das von den klassischen Transaktionskonzepten bekannte komplette Zurücksetzen nur in Ausnahmefällen eine sinnvolle Systemreaktion.

Die Frage, welche Typen von Aktivitätsträger in dem Entwurfsprozeß-Modell einzuführen sind, wird wesentlich davon beeinflusst, welche unterschiedlichen Anforderungen an die Aktivitätsträger gestellt werden und wie stark diese sich hinsichtlich ihre Eigenschaften unterscheiden.<sup>4</sup> In dem hier dargestellten Modell werden zwei verschiedene Aktivitätsträger eingeführt: **DESIGN ACTIVITY-** und **DESIGN OPERATION-**Aktivitätsträger.

---

3. Der Ablaufplan ist in dem Entwurfsphasen-Modell abgelegt.

4. Dies ist sowohl eine Modellierungsfrage als auch eine Frage der Anforderungen in einer konkreten Entwurfsumgebung, so daß die Validierung der hier vorgeschlagenen Konzepte einer Realisierung der Aktivitätsträger in einer konkreten Entwurfsumgebung vorbehalten sein wird.

## 4.2 Einführung der DESIGN ACTIVITY(DA)

Zur Bearbeitung einer (Teil-)Aufgabe in einem komplexen Entwurfsvorgang werden in innerbetrieblichen Vorgängen meist sog. **Aufträge mit einer vorgegebenen Zielstellung** erteilt, die dann von einem der beteiligten Ingenieure zu bearbeiten sind. Dies wird im Entwurfsprozeß-Modell über die Einführung eines speziellen Aktivitätsträgers, der **Design Activity (DA)**, reflektiert. Die erteilten Aufträge können meist nicht isoliert abgearbeitet werden (gleiches gilt für die zu deren Bearbeitung aktivierten DAs, daher wird im folgenden nur noch von den DAs geredet), vielmehr müssen sie zum Erreichen eines gemeinsamen Ziels zusammenwirken.

Für jede DA sind eine Reihe von Beziehungen zu anderen Elementen des Entwurfsumgebungsmodells festzulegen:

- **Zwischen den DAs** selbst existieren eine Reihe von **Wechselwirkungen** mit z.T. unterschiedlicher Bedeutung. Wie oben beschrieben kann die Abarbeitung eines Auftrages durch eine DA nicht isoliert betrachtet werden, vielmehr werden in einem Auftrag häufig Subaufträge erteilt (deren Ergebnisse zur Erledigung des eigentlichen Auftrags verwendet werden) bzw. Folgeaufträge gestartet. Dies führt zu dem Aufbau einer **Auftragshierarchie** zwischen den DAs. Das bedeutet, daß jede DA (mit Ausnahme einer ausgezeichneten Top-DA, die für den kompletten Entwurf zuständig ist) innerhalb einer anderen DA erzeugt wird und als deren Sub-DA abläuft.
- Für jede DA ist ein Entwurfsziel in Form einer **Auftragsspezifikation** festzulegen, die bei Beendigung der DA erfüllt sein muß. Da dieses Ziel eng mit der Aufgabenspezifikation des oder der Entwurfsobjekte verknüpft ist, die innerhalb der DA zu bearbeiten sind, wird in dem hier beschriebenen Modell die Auftragsspezifikation einer DA über die Soll-Features des oder der zugeordneten DOs festgelegt.

Eine DA hat ihren Auftrag erfüllt, wenn sie mindestens eine Version des DO (also eine DOV) erzeugt hat, die die in der Auftragsspezifikation angegebenen Features als Ist-Features beinhaltet.

- Jede DA hat Beziehungen zu den DOs, für deren Bearbeitung sie zuständig ist. Während der Ausführung der DA werden (i. allg. mehrere) DOVs für ein DO konfiguriert, deren Qualität durch eine Auswertung der Features bestimmt wird. Aufgrund der angesprochenen **kooperativen Wechselwirkungen** kann eine DA auf vorläufige DOVs anderer DAs zugreifen und diese in den weiteren Entwurfsprozeß einbeziehen. Dies wird im nächsten Kapitel noch verdeutlicht.
- Zur **Bearbeitung** eines Auftrages wird der DA ein **Ingenieur** zugeordnet (dieser ist in dem Entwurfssubjekt-Modell erfaßt), der für die Durchführung des Auftrags verantwortlich ist.

## 4.3 Einführung der DESIGN OPERATION (DOP)

Zur weiteren internen Strukturierung einer DA wird ein zweiter Aktivitätsträger eingeführt, die **Design Operation (DOP)**. Im folgenden werden die wichtigsten Eigenschaften einer DOP aufgezählt:

- Eine **DOP liest und schreibt Versionen** von komplex-strukturierten Daten (den in Bild 2 eingeführten Partialdaten).
- Die in einer DOP verfügbaren Entwurfsdaten werden zur Durchführung der Entwurfsaufgabe den Entwurfswerkzeugen zur Verfügung gestellt. Die Werkzeuge selbst und deren Anbindung sind in dem Entwurfswerkzeug-Modell festgelegt.
- Eine **DOP kann intern strukturiert sein**, wodurch u.a. eine Abfolge der Datenbereitstellung bzw. der Werkzeugaufrufe innerhalb der DOP festgelegt ist.
- Mit der Beendigung einer DOP wird die **Schemakonsistenz der geschriebenen Versionen** der Partialdaten (und natürlich deren Persistenz) zugesichert.

#### 4.4 Interne Strukturierung der Aktivitätsträger

Bild 9 gibt einen Überblick über die mögliche Unterteilung des gesamten Entwurfsprozesses durch die eingeführten Aktivitätsträger. Der gesamte Entwurfsprozeß (verantwortlich hierfür ist die Top-DA) wird zunächst durch die DA-Aktivitätsträger hierarchisch strukturiert, für die jeweils eine eigene Auftragspezifikation festgelegt ist. Die **Wechselwirkungen zwischen DAs** sind durch das im nächsten Kapitel beschriebene Kooperationsmodell festgelegt.

Die **interne Struktur einer DA** wird zum einen durch Operationen zur Verwaltung der hierarchischen DA-Struktur, zur Handhabung der Wechselwirkungen zwischen den DAs und zur Konfiguration bzw. Handhabung von versionierten Entwurfsdaten bestimmt. Zum anderen ist

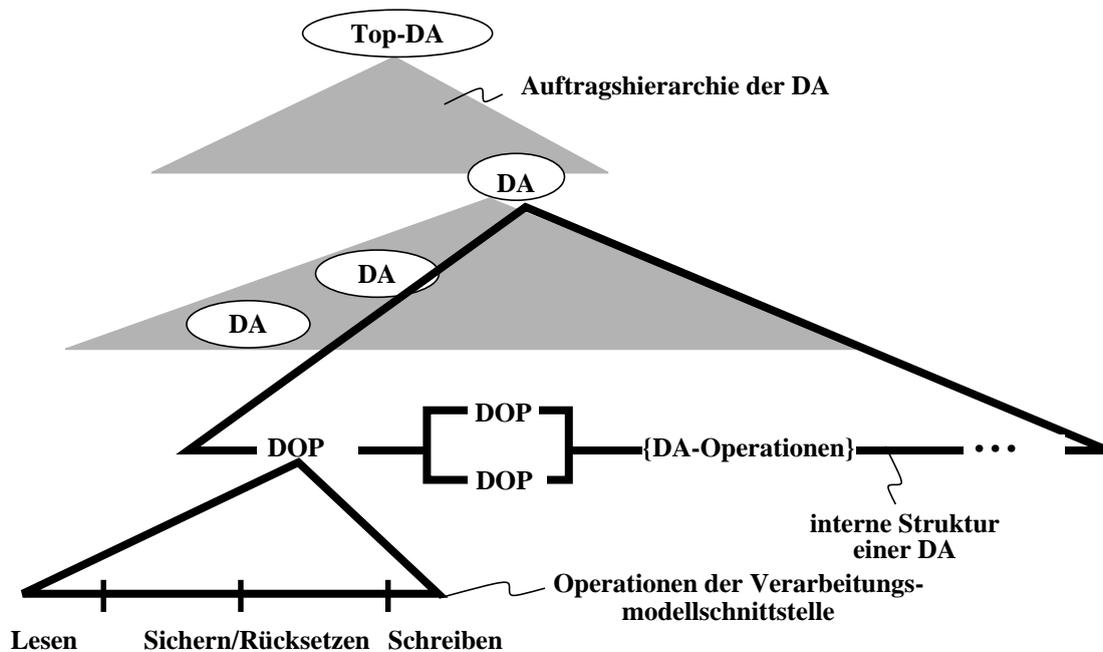


Bild 9: Unterteilung des gesamten Entwurfsprozesses durch die Aktivitätsträger

die interne Struktur durch die Einführung des DOP-Aktivitätsträgers geprägt. Das Ergebnis einer DOP sind immer eine oder mehrere (schemakonsistente) Versionen der Partial\_DO-Objekte, d.h., außerhalb der DOPs sind nur diese schemakonsistenten Partial\_DO-Versionen sichtbar.

Schließlich ist **in einer DOP** der Zugriff auf die Entwurfsdaten, die im Datenverwaltungssystem abgelegt sind, über die Operationen der Verarbeitungsmodell-Schnittstelle realisiert. Jede DOP kann selbst wieder intern strukturiert sein. In Kapitel 6 wird kurz auf die Verarbeitungsmodell-Schnittstelle eines strukturell objektorientierten DBS eingegangen, deren Funktionalität hier eine geeignete Basis bildet.

Bislang wurden noch keine Aussagen darüber gemacht, **wie die verschiedenen Aktivitätsträger intern aufgebaut sind**. Hier sind zunächst zwei Formen zu unterscheiden:

- Die interne Struktur wird **dynamisch während des Entwurfsprozesses** aufgebaut, d.h., der Entwerfer bestimmt interaktiv die nächste auszuführende Aktion in einem Aktivitätsträger (z.B. Erzeugen einer Sub\_DA, Starten einer DOP, Starten eines Werkzeuges usw.).
- Der Ablauf (bzw. die Auftragserteilung) ist durch **Ablaufpläne a priori festgelegt**. Der Ablaufplan muß auch weiterhin die Möglichkeit bieten, aufgrund von systemseitig oder vom Entwerfer getroffenen Entscheidungen bestimmte (alternative) Entwurfswege weiter zu verfolgen. Zur Beschreibung der Ablaufpläne sind in der Informatik verschiedene Konzepte bekannt, z.B. eine Beschreibung über Petrinetze [25] oder mittels eines weiteren Konzeptes, den ConTracts [33]. Die Modellierung der Ablaufpläne erfolgt im Entwurfsphasen-Modell.

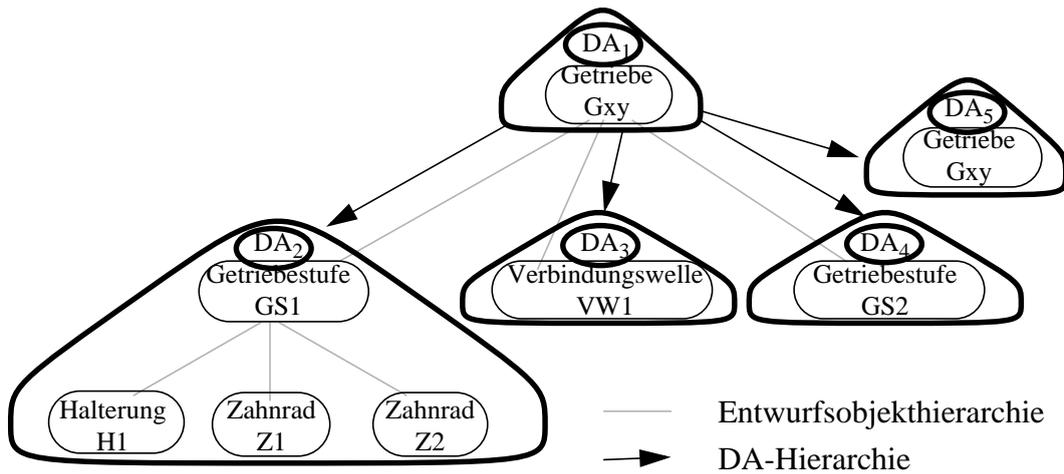
Ablaufpläne können auf jeder Ebene der Aktivitätsträger angesiedelt sein (also in einer DA bzw. in einer DOP). Dies bedeutet, daß Ablaufpläne ein zur Aktivitätsträger-Hierarchie orthogonales Konzept bilden, durch das der Freiheitsgrad hinsichtlich der internen Struktur eines Aktivitätsträgers eingeschränkt wird.

#### 4.5 Ein Anwendungsbeispiel aus dem mechanischen Konstruktionsbereich

Zum besseren Verständnis der bisher eingeführten Modellelemente einer Entwurfsumgebung wird an dieser Stelle ein Anwendungsbeispiel aus dem mechanischen Konstruktionsbereich eingeführt.

Das Anwendungsbeispiel behandelt den Entwurf eines Getriebes. Ausgangspunkt des Szenarios ist der Entwurf des Getriebes  $G_{xy}$  (vgl. Bild 7), für das eine Aufgabenstellung vorgegeben ist.<sup>5</sup> Des weiteren ist eine DA definiert, in der der Chefkonstrukteur das Getriebe zu entwerfen hat. Das bedeutet, daß am Ende der DA ein durch den Chefkonstrukteur freigegebener Getriebeentwurf aufgestellt sein muß. Der Chefkonstrukteur erstellt zunächst einen Getriebeplan. Dabei legt er fest, daß das Getriebe aus zwei Getriebestufen aufzubauen ist, einer Stirnradgetriebebestufe (GS1) und einer Planetenradgetriebebestufe (GS2), zwischen denen eine Verbindungswelle eingesetzt wird. Er verfeinert entsprechend die Baugruppenstruktur des Getriebes (damit

5. Im Rahmen dieses Beispiels werden jeweils nur einige Features der Entwurfsobjekte eingeführt, was jedoch zur Demonstration der Vorgehensweise ausreicht.



a) Aufbau der Auftragshierarchie und Zuordnung von Entwurfsobjekten DO zu den DAs

Entwurfsobjekt	Soll-Features	Auftragsspezifikation für DA
Getriebe Gxy	Antriebs-, Abtriebsdrehzahl 5000,60 Abmessungen L,B,H 90,45,45 "Simulation durchführen" "Freigabe"	DA <sub>1</sub> DA <sub>1</sub> DA <sub>5</sub> DA <sub>1</sub>
Getriebestufe GS1	Antriebs-, Abtriebsdrehzahl 5000,2500 Abmessungen L,B,H 35,40,40 Anschlußpunkt PGS1 (35,20,20)	DA <sub>2</sub> DA <sub>2</sub> DA <sub>2</sub>
Zahnräder Z <sub>1</sub> , Z <sub>2</sub>	"Auswahl aus Normteilkatalog N4711"	DA <sub>2</sub>
Getriebestufe GS2	Antriebs-, Abtriebsdrehzahl 2500,60 Abmessungen L,B,H 40,40,40 Anschlußpunkt PGS2 (50,20,20)*	DA <sub>4</sub> DA <sub>4</sub> DA <sub>4</sub>
Verbindungswelle VW <sub>1</sub>	Krafteinteilungsstellen 20,30 Anschlußpunkte P <sub>1</sub> VW <sub>1</sub> (35,20,20) P <sub>2</sub> VW <sub>1</sub> (50,20,20)*	DA <sub>3</sub> DA <sub>3</sub> DA <sub>3</sub>

\* Diese Features werden erst in nachträglichen Verhandlungen festgelegt, vgl. Kapitel 5

b) Soll-Feature-Spezifikation der Entwurfsobjekte (vereinfacht)

Bild 10: Aufbau einer Auftragshierarchie und Soll-Feature-Spezifikation

wird die Entwurfsobjekt-Hierarchie aufgebaut) und legt die Aufgabenstellung der Entwurfsobjekte fest (vgl. Bild 10b), was u.a. die Vorgabe von Anschlußpunkten zwischen den Getriebestufen und der Verbindungswelle zur Folge hat. Schließlich sollen die beiden Getriebestufen sowie die Verbindungswelle durch je einen Konstrukteur seiner Abteilung weiter ausgearbeitet

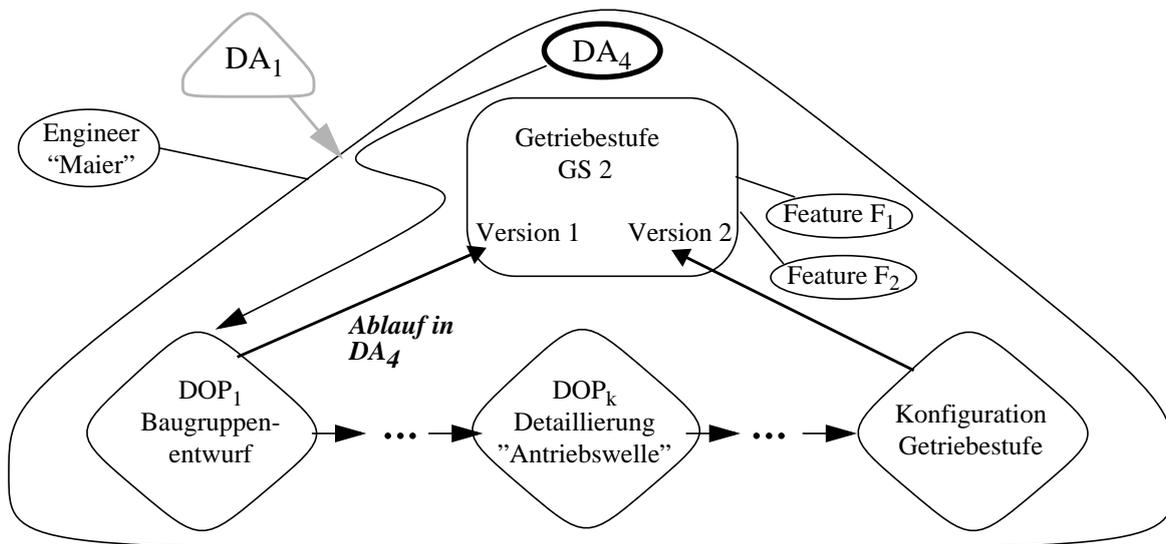


Bild 11: Detaillierung der DESIGN ACTIVITY DA4

werden. Dazu generiert er drei DAs, denen er je einen Konstrukteur zuordnet. Die Auftragspezifikation ist jeweils mit der Spezifikation der Entwurfsobjekte identisch.

Im folgenden soll der Ablauf von DA<sub>4</sub>, die für den Entwurf von Getriebestufe GS2 verantwortlich ist, näher betrachtet werden (vgl. Bild 6). DA<sub>4</sub> wird vom Ingenieur "Maier" bearbeitet. Die Auftragspezifikation ist durch zwei Soll-Features vorgegeben, die das zu entwerfende Objekt GS2 zu erfüllen hat. Es wird hier unterstellt, daß der Anwender (Ingenieur) an einem Workstation-Arbeitsplatz sitzt. An der Systemoberfläche wird jedem aktiven Aktivitätsträger ein eigenes Fenster zugeordnet, über das der Anwender mit dem Aktivitätsträger kommunizieren kann. In dem hier vorgestellten Szenario ist zunächst ein Fenster für DA<sub>4</sub> vorgesehen. Zur Durchführung der Entwurfsaufgabe muß der Anwender zunächst eine Verfeinerung von GS2 vornehmen, d.h. deren Baugruppenstruktur erstellen (dies schließt die Spezifikation einer Aufgabenstellung für jedes untergeordnete Bauteil ein). Hierfür will er ein entsprechendes Werkzeug benutzen. Dazu wird in DA<sub>4</sub> eine DOP gestartet. Auch für diese DOP wird ein eigenes Fenster erzeugt, in dem dann das Werkzeug abläuft. Bei einem (erfolgreichen) Ende dieser DOP ist die Entwurfsobjektstruktur von GS2 verfeinert worden. Die definierte Baugruppenstruktur für GS2 kann als eine erste, nicht qualifizierte Version des Entwurfsobjektes GS2 betrachtet werden. In den weiteren Arbeitsschritten in DA<sub>4</sub> sind die Entwurfsobjekte der eben aufgebauten Baugruppenstruktur zu konkretisieren. So wird beispielsweise zur Detaillierung des Einzelteils "Antriebswelle" eine weitere DOP gestartet, in der der Einzelteilentwurf durchgeführt wird. In dieser DOP können zur Durchführung der Aufgabe mehrere Werkzeuge eingesetzt werden. So kann beispielsweise zunächst ein CAD-Modellierungssystem aktiviert werden, bevor ein Finite-Elemente-Programm die Belastbarkeit der entworfenen Welle überprüft.

Nachdem die verschiedenen Entwurfsobjekte (möglicherweise in mehreren Versionen) konstruiert worden sind, wird das Entwurfsobjekt GS2 zusammengesetzt, d.h. aus den Sub-DOs konfiguriert. Damit ist eine Version (Konfiguration) von GS2 erstellt, für die die Ist-Feature-Menge zu bestimmen ist. Sind sämtliche Features erfüllt (die Behandlung vorläufiger Versio-

nen wird im nächsten Kapitel behandelt), so erhält die übergeordnete Design Activity  $DA_1$  die erstellte Version von  $GS_2$  und  $DA_4$  wartet darauf, daß sie nach einer expliziten Bestätigung von  $DA_1$  beendet werden kann. Nachdem  $DA_1$  die Ergebnisse von  $DA_2$ ,  $DA_3$  und  $DA_4$  zur Verfügung gestellt wurden, wird in  $DA_1$  das gesamte Getriebe zusammengesetzt, d.h., in  $DA_1$  wird für das Getriebe  $G_{xy}$  eine Konfiguration durchgeführt. Zur Überprüfung der Funktionsfähigkeit soll ein komplexes Simulationsprogramm eingesetzt werden, das nur von einem Experten bedient werden kann. Dazu wird eine weitere DA generiert ( $DA_5$ ), deren Auftragspezifikation aus der Durchführung des Simulationsprogrammes besteht. Werden von diesem Programm keine Fehler aufgedeckt, so sendet  $DA_1$  den Sub-DAs die Bestätigung zu, daß ihr Auftrag erfüllt ist und sie beendet werden können. Der Chefkonstrukteur führt ein Freigabeverfahren für das Getriebe  $G_{xy}$  durch und kann schließlich das Ergebnis an eine übergeordnete DA weiterreichen.

## 5 Das Entwurfskooperations-Modell

Ein **Kooperationsmodell** hat Mechanismen bereitzustellen, die eine kooperative Entwurfsvorgehensweise geeignet unterstützen (vgl. auch [11, 17, 19]), d.h. das Zusammenwirken zwischen den DAs<sup>6</sup> zum Erreichen des gemeinsamen Ziels handhaben. Eine Untersuchung von verschiedenen Entwurfsbereichen hat in dem aufgebauten Entwurfsumgebungsmodell zu einer Unterscheidung von drei unterschiedlichen Kooperationsbeziehungen geführt [29]:

- **Auftragsbeziehung**

Mit **Auftragsbeziehung** wird die bereits eingeführte Beziehung zwischen einer DA und der von ihr erzeugten Sub-DAs bezeichnet. Gegenstand der Kooperation ist hier die Auftragspezifikation der Sub-DAs und deren Realisierbarkeit.

- **Verhandlungsbeziehung**

Die **Verhandlungsbeziehung** ist zwischen zwei DAs geknüpft, die über eine Verfeinerung bzw. Ergänzung ihrer Auftragspezifikation verhandeln. Gegenstand der Kooperation ist auch hier die Auftragspezifikation einer DA.

- **Austauschbeziehung**

Über die **Austauschbeziehung** wird die Weitergabe von vorläufigen Versionen eines Entwurfsobjektes einer DA an andere DAs repräsentiert. Eine DA, die vorläufige Versionen anderer Entwurfsobjekte nutzt, kann damit die eigene Entwurfsaufgabe fortführen, wohlwissend, daß die genutzten vorläufigen Versionen noch Änderungen unterworfen sein können. Gegenstand der Kooperation sind hierbei also vorläufige Entwurfsobjektversionen, deren Austausch über die Qualität (Ist-Features) der Versionen koordiniert wird.

---

6. Wird im folgenden vom Zusammenwirken bzw. Kooperieren zwischen DAs geredet, so sind damit meist die Ingenieure gemeint, die den DAs zugeordnet sind.

In den nachfolgenden Ausführungen werden die drei Kooperationsbeziehungen genauer vorgestellt. Dazu wird deren Semantik verfeinert und die Kooperationsoperationen (sowie deren Zusammenwirken) beschrieben, über die die kooperativen Wechselwirkungen zwischen den DAs ablaufen. Schließlich werden die Kooperationsbeziehungen anhand eines Beispiels verdeutlicht, das auf dem im letzten Kapitel vorgestellten Anwendungsszenario aufbaut.

### **Die Auftragsbeziehung**

Über diese Beziehung wird eine DA mit den von ihr erzeugten Sub-DAs in Beziehung gesetzt. Eine DA kann beliebig viele Sub-DAs erzeugen<sup>7</sup>. In einer Sub-DA wird ein Teil der Entwurfsaufgabe einer DA bearbeitet, d.h., die Sub-DA trägt zum Erreichen der Aufgabe (also des Entwurfsziels) der DA bei. Dazu wird für jede Sub-DA eine eigene Auftragspezifikation festgelegt. Im weiteren Entwurfsprozeß wird dann in der DA auf die Ergebnisse der Sub-DAs Bezug genommen und diese in den Entwurfsvorgang einbezogen.

Durch die Auftragsbeziehung wird eine **wechselseitige Abhängigkeit zwischen der DA und der Sub-DA** festgelegt. Im allgemeinen kann nicht zugesichert werden, daß die beim Erzeugen einer Sub-DA mitgegebene Auftragspezifikation keinerlei Änderungen unterworfen ist. Es sind Entwurfssituationen möglich, in denen der Ingenieur einer auftraggebenden DA die Auftragspezifikation einer Sub-DA abändern muß. Gründe dafür sind etwa eine Änderung seiner eigenen Auftragspezifikation oder die Feststellung, daß die zuvor erstellte Spezifikation einer Sub-DA nicht zu dem erwarteten Ergebnis führt. Über die Operation `MODIFY_SPECIFICATION` kann er die Auftragspezifikation einer Sub-DA abändern, indem er Soll-Features verändert oder hinzufügt. Stellt umgekehrt ein Ingenieur, der einer Sub-DA zugeordnet ist, im Verlauf der Entwurfsarbeit fest, daß die ihm übertragene Auftragspezifikation so nicht erfüllbar ist, so muß er dies unter Benennung der entsprechenden Features über die Operation `IMPOSSIBLE_SPECIFICATION` der auftraggebenden DA mitteilen. Dabei sollte er Vorschläge für eine angepaßte Soll-Feature-Spezifikation mitgeben. In der Folge kann der Ingenieur der auftraggebenden DA die Sub-DA beenden (`DELETE`-Operation) oder ihr eine modifizierte Spezifikation (`MODIFY_SPECIFICATION`-Operation) übergeben.

### **Die Verhandlungsbeziehung**

Ein Kooperationsmodell muß die Möglichkeit bieten, die zu Beginn einer DA festgelegte Auftragspezifikation während des weiteren Entwurfsprozesses aufgrund von Anforderungen anderer DAs zu verfeinern oder zu ergänzen, ohne daß damit einen Widerspruch zur anfänglichen Auftragspezifikation entsteht. Zur Beschreibung dieser Kooperationsform wird eine zweite Kooperationsbeziehung, die Verhandlungsbeziehung, eingeführt. Zunächst ist zu klären, welche DAs der Auftragshierarchie sinnvollerweise miteinander in Verhandlung treten können:

---

7. DA und Sub-DA bezeichnen beide eine Design Activity mit identischer Funktionalität; die unterschiedlichen Namen sollen lediglich einer Unterscheidung der DAs ermöglichen.

- Werden in einer DA zur Bearbeitung eines Auftrages mehrere Sub-DAs erzeugt, so müssen deren Ergebnisse (d.h. die entwickelten Entwurfsobjekte) i. allg. aufeinander abgestimmt sein (vgl. Beispiel in Bild 10). Da nicht alle Abhängigkeiten zwischen den Entwurfsobjekten zu Beginn feststehen und daher nicht vollständig in der Auftragspezifikation enthalten sind, muß während des Entwurfsprozesses zwischen den Sub-DAs einer DA über die Verfeinerung der Auftragspezifikation verhandelt werden.
- Auch zwischen einer DA und den von ihr erzeugten Sub-DAs sind Verhandlungsbeziehungen sinnvoll. Über die Auftragsbeziehung können bislang nur imperative Modifikationen der Auftragspezifikation vorgenommen werden.
- Das in einer DA entwickelte Entwurfsobjekt kann im Entwurf verschiedener weiterer DOs eingesetzt werden (ein Einzelteil kann beispielsweise in verschiedenen Baugruppen einbezogen werden), für deren Entwurfsprozeß ebenfalls DAs zuständig sind. Daher ist es sinnvoll, zwischen zwei beliebigen DAs eine Verhandlungsbeziehung aufbauen zu können, um so den Entwurf eines DOs möglichst entsprechend den Forderungen aller Benutzer gestalten zu können, die das Entwurfsobjekt nutzen wollen.

Soll zwischen zwei DAs eine Verhandlung über die Auftragspezifikation der einen DA (oder von beiden DAs) durchgeführt werden, so kann von der einen DA über die PROPOSE-Operation der zweiten DA ein Vorschlag über eine Verfeinerung bzw. Ergänzung der Soll-Feature-Menge mitgeteilt werden. Hiermit wird ein Verhandlungsprotokoll gestartet. In der zweiten DA kann der Spezifikationsvorschlag angenommen (AGREE-Operation) oder verworfen (DIS-AGREE-Operation) werden. Das Verhandlungsprotokoll erlaubt eine iterative Vorgehensweise, d.h., es können mehrere bzw. alternative Vorschläge über die PROPOSE-/AGREE-/DIS-AGREE-Operationen verhandelt werden.

### **Die Austauschbeziehung**

Während der Ausführung einer DA braucht der zugeordnete Ingenieur Informationen über den Entwurfsvorgang kooperierender Ingenieure, d.h., er möchte auf die von ihnen erzeugten **vorläufigen Versionen der Entwurfsobjekte zugreifen** können. Dies ist z.B. dann notwendig, wenn in einer DA eine Konfiguration für ein DO durchzuführen ist und die assoziierten Sub-DOs in anderen DAs entwickelt werden. Über die REQUIRE-Operation kann eine DA eine DOV einer anderen DA anfordern. Dazu muß die **geforderte Qualität der DOV durch die Angabe einer Feature-Menge**, die in der DOV erfüllt sein müssen, bei der REQUIRE-Operation spezifiziert werden. Existiert eine DOV mit den geforderten Features, so kann die Anforderung erfüllt werden und die DOV wird der anfordernden DA bereitgestellt. Ansonsten kann zu diesem Zeitpunkt keine DOV bereitgestellt werden. Jedoch wird die Austauschbeziehung aufgebaut und wenn zu einem späteren Zeitpunkt von der DA eine DOV mit den geforderten Features bereitgestellt wird (PROPAGATE-Operation, siehe unten), so wird dies der anfordernden DA über einen Notifikationsmechanismus mitgeteilt. In der anfordernden DA ist sehr wohl bekannt, daß die bereitgestellte DOV noch weiteren Änderungen unterworfen sein kann.

Umgekehrt muß eine DA vorläufige oder vollständige Versionen, die von anderen DAs genutzt werden können, explizit durch eine PROPAGATE-Operation “nach außen” verfügbar machen. In einer DA können mehrere PROPAGATE-Operationen durchgeführt werden, d.h., es werden mehrere Versionen des Entwurfsobjektes für andere DAs verfügbar gemacht. Durch eine nachfolgende PROPAGATE-Operation können jedoch nur Versionen bereitgestellt werden, deren Ist-Feature-Menge keine Teilmenge einer bereits propagierten Version ist.<sup>8</sup>

Wird eine neue Version über die PROPAGATE-Operation verfügbar gemacht, so ist dafür zu sorgen, daß alle DAs, die eine ältere Version nutzen, deren Ist-Features in der neuen Version ebenfalls erfüllt werden, nach einer gewissen Übergangszeit die neue Version in ihren Entwurf einbeziehen. Dies ist notwendig, weil am Ende der DA nur noch eine vollständige Version verfügbar ist. Dies ist auch von Seiten der nutzenden DAs eine sinnvolle Vorgehensweise, da diese sehr wohl die gesamte Auftragspezifikation des einbezogenen Entwurfsobjektes kennen und somit von Beginn an die Einbeziehung der vollständigen Version geplant ist.

Es ist Aufgabe des Kooperationsmodells, eine Unterstützung bei der Aktualisierung benutzter Versionen anzubieten. Dazu werden **Aktionen** eingeführt. Aktionen bilden eine Art Trigger-Konzept [21], die beim Propagieren einer neuen DOV aktiviert werden. Zentrale Idee des Aktion-Konzeptes ist es, in Folge einer propagierten neuen DOV in den benutzenden DAs Folgeaktionen anzustoßen, die abhängig von den geänderten Features der neuen DOV unterschiedlich sein können.

Die in einer Aktion definierten Folgeaktionen können einfache Mitteilungen an die benutzende DA sein, in der die Bereitstellung der neuen DOV mitgeteilt wird.<sup>9</sup> Die benutzende DA kann im weiteren Entwurfsablauf auf die neue DOV Bezug nehmen. Es sind aber auch komplexere Aktionen denkbar, die in der benutzenden DA den Teil des gesamten bisherigen Entwurfsvorgangs, der von der benutzten DO abhängig ist, unter Einbeziehung der neuen DOV automatisch wiederholen.

Zur Verdeutlichung des Ablaufs einer DA sowie der Einbindung der Operationen des Kooperationsmodells ist in Bild 12 ein Zustandsdiagramm für eine DA abgebildet. Zustandsübergänge werden durch Operationen in einer DA bzw. durch Operationen von anderen DAs ausgelöst. Die GENERATE-Operation bewirkt den Zustandswechsel von UNKNOWN nach INITIATED. Mit dieser Operation werden eine Auftragspezifikation und ein verantwortlicher Ingenieur festgelegt. Eine START-Operation bewirkt den Übergang in den Zustand ACTIVE, in dem der eigentliche Entwurfsprozeß abläuft. Eine DA wechselt in den Zustand NEGOTIATE, wenn entsprechend dem Kooperationsmodell Wechselwirkungen mit anderen DAs zu behandeln sind. Ist mindestens eine DOV erstellt worden, die der Auftragspezifikation genügt, so kann die DA in den Zustand READY\_FOR\_TERMINATION wechseln. Erst durch eine expli-

---

8. Natürlich können intern in einer DA DOVs mit einer geringeren Qualität erstellt werden, auf diese kann aber keine PROPAGATE-Operation ausgeführt werden.

9. Über eine solche Mitteilung kann eine DA in Kenntnis gesetzt werden, daß eine mit einer bestimmten Qualität angeforderte Version, die bislang nicht zur Verfügung gestellt werden konnte, nun bereitsteht.

zite Bestätigung des Auftraggebers, also der diesen Auftrag generierenden DA, geht die DA in den Zustand FINISHED über. Eine DELETE-Operation bewirkt einen Zustandswechsel nach UNKNOWN.

### Anwendungsbeispiele zur Benutzung der Kooperationsbeziehungen

Zum Schluß dieses Abschnitts wird die Nutzung der Kooperationsbeziehungen an dem in Bild 10 aufgebauten Anwendungsszenario demonstriert.

Während des Entwurfsprozesses in DA<sub>4</sub> stellt sich heraus, daß die vorgegebenen Abmessungen zu klein sind, daß also die Getriebestufe in diesen Abmaßen nicht realisiert werden kann. Über die IMPOSSIBLE\_SPECIFICATION-Operation wird DA<sub>1</sub> benachrichtigt und gleichzeitig ein Vorschlag unterbreitet, nämlich eine Zugabe in der Länge L um 10 Einheiten. Nun müssen in DA<sub>1</sub> geeignete Maßnahmen getroffen werden. Es wird entschieden, DA<sub>4</sub> den zusätzlichen Platz zur Verfügung zu stellen (mittels der MODIFY\_SPECIFICATION-Operation). Gleichzeitig müssen aber in der Getriebestufe GS1 5 Einheiten eingespart werden und der Anschlußpunkt PGS1 auf (35, 20, 20) gesetzt werden (MODIFY\_SPECIFICATION-Operation an DA<sub>2</sub>). Auch in DA<sub>3</sub> wird die Spezifikation geändert und der Anschlußpunkt P1VW1 auf (35, 20, 20) gesetzt.

Während des weiteren Entwurfsprozesses wird in DA<sub>4</sub> festgestellt, daß bislang noch kein Anschlußpunkt zur Verbindungswelle VW1 festgelegt wurde. Daraufhin wird zwischen DA<sub>4</sub> und DA<sub>3</sub> eine Verhandlungsbeziehung aufgebaut. Mittels der PROPAGATE-Operation macht DA<sub>4</sub> den Vorschlag, den Anschlußpunkt (damit ist hier die Krafteinleitungsstelle gemeint) in dem Punkt (45, 25, 25) festzulegen. Da in DA<sub>3</sub> auch der Anschlußpunkt zu GS1 zu berücksichtigen ist, kann DA<sub>3</sub> dem Vorschlag nicht zustimmen und gibt mit der DISAGREE-Operation einen

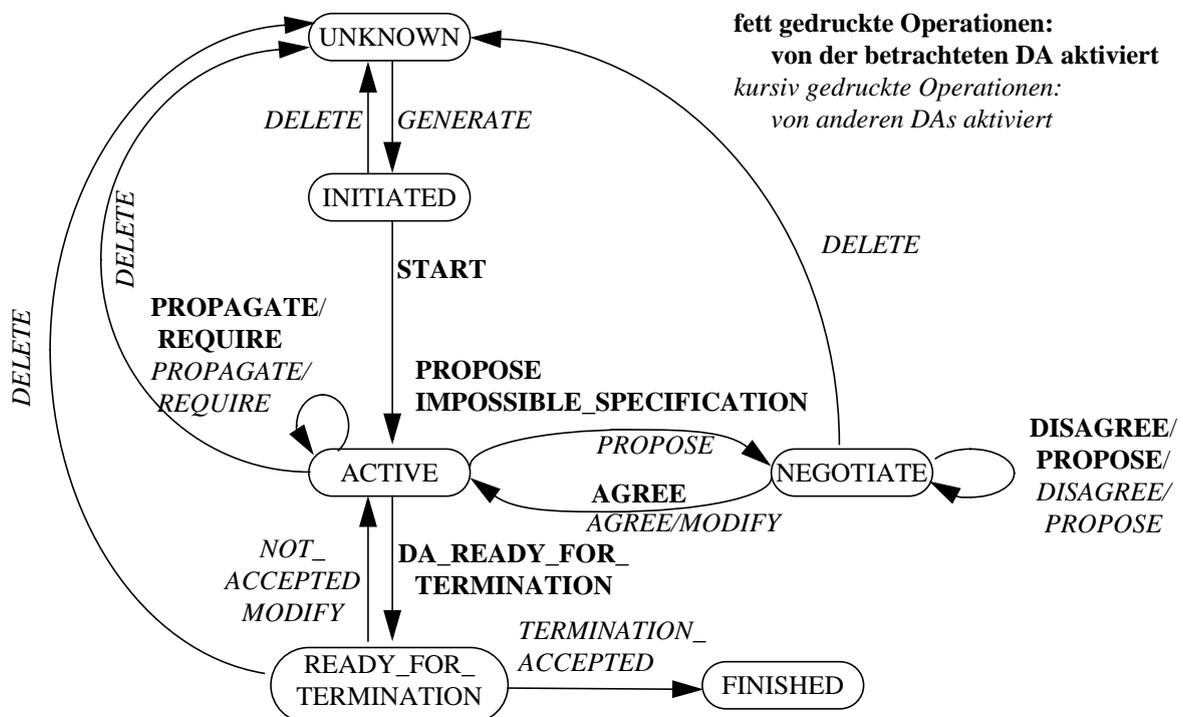


Bild 12: Zustände und Zustandsübergänge einer Design Activity

Gegenvorschlag mit (z.B. die Koordinaten (45, 20, 20)), worauf DA<sub>4</sub> wieder mit einer erneuten PROPAGATE-Operation antwortet (z.B. die Koordinaten (50, 20, 20)). Da DA<sub>3</sub> mit diesem Vorschlag einverstanden ist, wird dieser Punkt mit der AGREE-Operation bestätigt und die Verhandlung zwischen den beiden Aktivitätsträgern beendet.

In DA<sub>1</sub> ist das Getriebe zusammenzusetzen. Für eine erste Version (Konfiguration) reicht es dem Chefkonstrukteur, wenn er über die Austauschbeziehung eine vorläufige Version der Entwurfsobjekte GS1, GS2 und VW1 erhält, in denen die Features erfüllt sind, die die Abmessungen und die Anschlußpunkte betreffen. Er definiert entsprechende REQUIRE-Operationen an DA<sub>2</sub>, DA<sub>3</sub> und DA<sub>4</sub>. DA<sub>2</sub> und DA<sub>3</sub> haben bereits eine Version propagiert, die die geforderten Features erfüllt, während DA<sub>4</sub> noch keine entsprechende Version bereitgestellt hat. Erst zu einem späteren Zeitpunkt propagiert auch DA<sub>4</sub> eine vorläufige Version, die den Anforderungen, die über die REQUIRE-Operation gefordert wurden, genügt. Über eine definierte Aktion wird DA<sub>1</sub> von der neuen DOV des Entwurfsobjektes GS2 informiert. Daraufhin könne in DA<sub>4</sub> mit dem Zusammenbau begonnen und als Ergebnis eine erste Version von G<sub>xy</sub> konfiguriert werden. Danach wird in DA<sub>1</sub> die Ist-Feature-Menge bestimmt. Hat die Version bereits eine Qualitätsstufe erreicht, die von anderen DAs sinnvoll genutzt werden kann, so wird die Version über die PROPAGATE-Version den anderen DAs verfügbar gemacht.

## 6 Ein Architekturvorschlag für ein DB-integriertes Ingenieursystem

Nachdem nun die wesentlichen Konzepte des Entwurfsumgebungsmodells vorgestellt wurden, wird in diesem Abschnitt ein Architekturvorschlag für ein DB- integriertes Ingenieursystem zur systemseitigen Abbildung des Modells vorgestellt. Zunächst werden einige allgemeine Abbildungskriterien aufgezählt, die den Aufbau eines DB-integrierten Ingenieursystems bestimmen. Im folgenden wird dann auf das zugrundegelegte strukturell objektorientierte **Workstation/Server-Datenbanksystem (WSDBS)** sowie auf eine mögliche Systemarchitektur eingegangen.

### 6.1 Abbildungskriterien für ein DB-integriertes Ingenieursystem

Aus den bisherigen Modellüberlegungen wird deutlich, daß eine integrierte Verwaltung der Produktdaten eines Entwurfsobjektes (der primären Entwurfsdaten sowie der strukturellen Metadaten auf den Entwurfsdaten) und der weiteren Entwurfsumgebungsinformationen die Grundlage für eine durchgängige Entwurfsunterstützung bildet. Dieser Gedanke bestimmt wesentlich die hier vorgestellte Systemarchitektur eines integrierten Ingenieursystems. Insgesamt ist der im folgenden aufgezeigte Architekturvorschlag hauptsächlich durch nachfolgende Abbildungskriterien bestimmt:

- Der Arbeitsplatz eines Ingenieurs ist durch eine leistungsfähige, weitgehend **autonome Workstation** geprägt, die über ein lokales Netzwerk mit anderen Workstations und mit einem oder mehreren **Server-Rechnern** (die zentrale Dienstleistungen bereitstellen) verbunden ist.

- **Zentrale Systemkomponente** in einem integrierten Ingenieursystem ist die **Datenverwaltungskomponente**. Sie hat die Aufgabe, die primären Entwurfsdaten der Entwurfsobjekte, die Metainformation über die Entwurfsobjekte und darüber hinaus die Informationsstrukturen, die sich aus einer Abbildung des Entwurfsumgebungsmodells ergeben, zu verwalten.
- Zur systemseitigen Abbildung der aufgeführten **Teilmodelle** zur Beschreibung einer Entwurfsumgebung werden in dem hier unterbreiteten Systemvorschlag eine Reihe von **Basisdiensten** bereitgestellt, die an ihrer Schnittstelle die geforderte Funktionalität anbieten. In ihrer Gesamtheit bilden die Basisdienste eine Zusatzebene aufbauend auf der Datenverwaltungskomponente. Sie bieten eine anwendungsorientierte Systemschnittstelle an, die an den Anforderungen von technischen Entwurfsanwendungen ausgerichtet ist.
- Komplexe Entwurfsaufgaben werden von mehreren Entwerfern (kooperativ) durchgeführt. Daraus folgt, daß eine Anwendung in einem integrierten Ingenieursystem nicht isoliert abläuft, sondern vielmehr ein **Mehrbenutzerbetrieb** in einer verteilten Workstation/Server-Systemumgebung ermöglicht werden muß. Die Organisation und Verwaltung eines Mehrbenutzerbetriebes hat natürlich Konsequenzen auf die Art und Weise der **Verteilung der angesprochenen Basisdienste**, insbesondere was deren Aufteilung zwischen Server- und Workstation-Seite anbelangt.

## 6.2 Ein strukturell objektorientiertes Workstation/Server-Datenbanksystem

Die hier vorgeschlagene Systemarchitektur (vgl. Bild 8) ist durch die oben aufgeführten Abbildungskriterien bestimmt. Zur Datenverwaltung wird ein strukturell objektorientiertes Workstation/Server-Datenbanksystem eingesetzt [Hü92]. Die generellen Vorteile eines DBS zur Produktdatenverwaltung sind allgemein bekannt und sollen daher hier nicht weiter diskutiert werden.

Das in [16] vorgeschlagene WSDBS ist ein spezielles verteiltes DBS, das aus zwei sich gegenseitig ergänzenden Komponententypen zusammengesetzt ist, die entsprechend ihrer Funktionalität als Server-DBS bzw. Client-DBS bezeichnet werden. Das **Server-DBS** bietet die allgemein üblichen Dienste eines Datenbanksystems an, die dann von dem **Client-DBS** genutzt werden. Dieser Verteilung liegt die Vorstellung zugrunde, daß das Server-DBS auf einem leistungsfähigen Mainframe-Rechner angeordnet ist und die Client-DBS jeweils auf einer Workstation ablaufen, der die technischen Anwendungsprogramme zugeordnet sind, und die dann den Arbeitsplatz des Benutzers bilden (vgl. Bild 8). Durch diese WSDBS-Architektur wird eine flexible und dynamische Datenverteilung unterstützt. Die von einer Anwendung auf der Workstation benötigten Daten werden über das Client-DBS vom Server-DBS angefordert, womit ein dynamischer Verteilungsvorgang auf die Workstation-Seite verbunden ist. Die Daten werden auf der Workstation für die Dauer der Verarbeitung ausschließlich durch das Client-DBS verwaltet.

Die Realisierung eines WSDBS wird mit dem Nonstandard-Datenbanksystem PRIMA erreicht [12]. Dabei übernimmt der PRIMA-Kern die Aufgaben eines Server-DBS, während das Client-

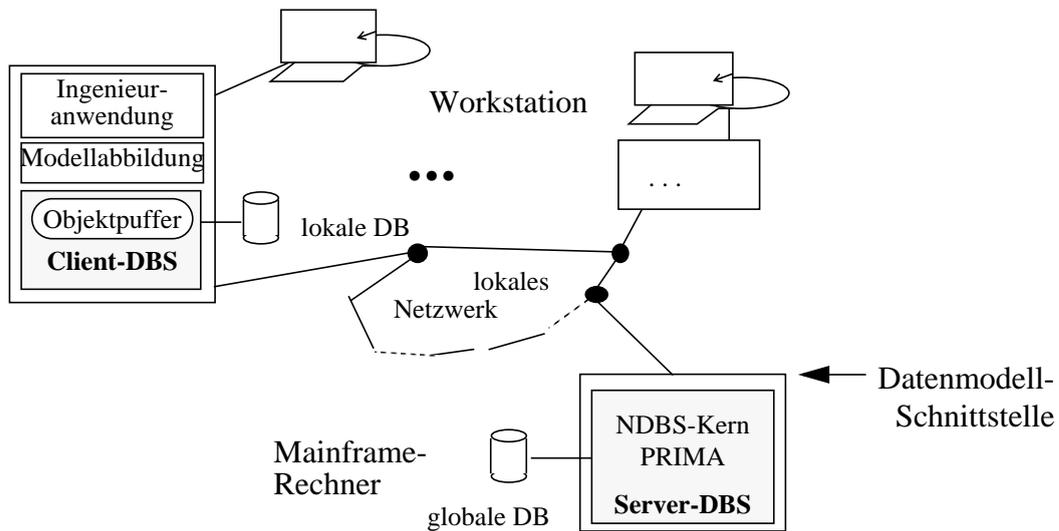


Bild 8: Struktur eines Workstation/Server-DBS (WSBDBS)

DBS auf Workstation-Seite hauptsächlich die Verwaltung eines integrierten Objektpuffers zur Aufgabe hat (vgl. unten).

Das durch PRIMA realisierte strukturell objektorientierte MAD-Modell [22] erlaubt die deskriptive Spezifikation und Handhabung komplexer, heterogen zusammengesetzter Objekte, die im MAD-Modell u.a. eine *netzwerkartige* bzw. *rekursive Struktur* annehmen können.

In der Umgebung des PRIMA-Systems ist eine Verarbeitung auf der Seite des Server-DBS und anwendungslokal auf Seite des Client-DBS möglich. Während für die Verarbeitung auf Server-DBS-Seite die Algorithmen (in Form von DB-Anweisungen) zu den Daten übertragen werden, werden bei der lokalen Verarbeitung die Daten zu den Verarbeitungsalgorithmen gebracht. Im folgenden werden die wesentlichen Eigenschaften des Verarbeitungsmodells zusammengefaßt. Eine umfassende Vorstellung ist in [14, 16] zu finden.

Die lokale Verarbeitung auf Client-DBS-Seite erfordert ein **zweistufiges Verarbeitungskonzept**. Die Idee hinter diesem Vorgehen ist einfach: können im Vorfeld der eigentlichen Verarbeitung ein bzw. mehrere "Verarbeitungsgegenstände" bereits ermittelt und zur lokalen Datenhaltung auf die Workstation-Seite transferiert werden (CHECKOUT), so sind die folgenden Verarbeitungszugriffe um so direkter und damit kostengünstiger zu realisieren. Die Ergebnismenge wird in den dort integrierten **Objektpuffer** abgelegt. In der zweiten Stufe erfolgt dann die eigentliche Verarbeitung. Hier ist eine mehr prozedurale oder navigierende Vorgehensweise gefordert, da so eine einfache, sehr direkte und effiziente Datenversorgung der Anwendungsalgorithmen erreicht werden kann [10]. Nach erfolgter Verarbeitung müssen dann die veränderten Daten zurück zur globalen Datenhaltung übermittelt werden (verzögertes Einbringen der Änderung, CHECKIN).

### 6.3 Systemarchitektur eines WSDBS-basierten integrierten Ingenieursystems

Nachdem nun die wesentlichen Abbildungskriterien aufgezählt und ein Workstation/Server-DBS als Basis zur Verwaltung der anfallenden Informationsstrukturen vorgestellt wurde, wird

jetzt eine Systemarchitektur für ein darauf aufbauendes integriertes Ingenieursystem eingeführt, mit dem die systemseitige Abbildung des Entwurfsumgebungsmodells erfolgt.

Neben einer Verwaltung der primären Entwurfsdaten erfordert eine Abbildung des oben beschriebenen Entwurfsumgebungsmodells Konzepte zur Versionierung komplexer Entwurfsdaten. Aufbauend auf dem NDBS-Kern wird hierfür eine eigene Komponente bereitgestellt, die **Objektversions-Verwaltungskomponente (OVV)**, die die Definition, Verwaltung und Handhabung versionierter Entwurfsdaten erlaubt.

Die Abbildung der übrigen Teilmodelle erfordert die Spezifikation weiterer **Basiskomponenten** und deren Einordnung in den aufgespannten Architekturrahmen. In den bisherigen Ausführungen wurde insbesondere auf das Entwurfsprozess- und das Entwurfskooperations-Modell eingegangen, so daß hier deren Abbildung näher betrachtet werden soll. Diese Entwurfsumgebungsdaten werden ebenfalls mit dem MAD-Datenmodell beschrieben und in der globalen Datenbank abgelegt. Die beiden **Komponenten Design-Manager und Kooperationsmanager** übernehmen die Aufgabe einer Abbildung der beiden genannten Teilmodelle. Die Komponente des Design-Managers stellt Operationen zur Verwaltung der DA- und DOP-Aktivitätsträger sowie zur Handhabung der Features (Erzeugen, Löschen, Modifizieren, Auswerten, Beziehungen zu DOs, DOVs bzw. DAs knüpfen oder lösen usw.) bereit. Die Kooperationsmanager-Komponente bietet die Operationen zur Handhabung der drei Kooperationsbeziehungen an. Darüber hinaus muß die Komponente die Wahrung der Kooperationsprotokolle, die auf den Kooperationsbeziehungen definiert sind und Einfluß auf die Abfolge der Kooperationsoperationen haben, gewährleisten. Um eine atomare Abarbeitung der Operationen der beiden Komponenten zuzusichern, wird jede Operation dieser Komponente als eine eigene Datenbank-Transaktion realisiert. Dies hat zur Folge, daß der DB-Zustand der Entwurfsumgebungsdaten immer den aktuellen Entwurfzustand widerspiegelt.

Zur Abbildung des gesamten Entwurfsumgebungsmodells sind eine Reihe weiterer Basiskomponenten notwendig, auf die hier nicht näher eingegangen werden soll.

Für jede Basiskomponente ist zu klären, ob sie auf der Server- oder der Workstation-Seite der Systemarchitektur anzuordnen ist. Für das zugrundeliegende NDBS wurde bereits an früherer Stelle eine Aufteilung in **Server-DBS** und **Client-DBS** und deren Zuordnung zur Server- bzw. Workstation-Seite vorgestellt.

Für die Design-Manager- und die Kooperationsmanager-Komponente ist zu beachten, daß die angebotenen Operationen nicht isoliert für eine Entwurfsanwendung abgearbeitet werden können. Vielmehr benötigen die meisten Operationen der beiden Komponenten Informationen über den aktuellen Entwurfzustand der übrigen beteiligten Entwurfsanwendungen. Werden die Komponenten jeweils auf Workstation-Seite angeordnet, so ist eine ständige Abstimmung zwischen den verteilten Systemkomponenten notwendig. Die Systemarchitektur in Bild 9 sieht vor, die Kooperationsmanager-Komponente ganz und die Design-Manager-Komponente teilweise auf der Server-Seite anzuordnen. Die Design-Manager-Komponente auf Workstation-Seite hat die Aufgabe, eine Ablaufumgebung für die vom Anwender gestarteten Aktivitätsträger bereitzustellen.

## 7 Zusammenfassung

In diesem Beitrag wurden Überlegungen zur Entwicklung eines DB-integrierten Ingenieursystems vorgestellt. Dabei stand der strukturelle Aufbau eines formalen Modells zur allgemeinen Beschreibung von Entwurfsumgebungen im Vordergrund. Natürlich können in einem formalen Entwurfsumgebungsmodell nur ein Teil der Aspekte und Ereignisse, die den Entwurfsvorgang beeinflussen, direkt unterstützt und abgebildet werden. Zusätzlich ist der tatsächliche Entwurfsvorgang häufig durch informelle Methoden bestimmt<sup>10</sup>, die sich nicht rechnerseitig erfassen lassen, so daß deren Resultat nachträglich in geeigneter Weise im rechnerinternen Modell reflektiert werden muß.

Das vorgestellte anwendungsneutrale **Entwurfsumgebungsmodell** ist in eine Reihe von Teilmodellen mit einer dedizierten Funktionalität untergliedert. Es wurden die Informationsstruk-

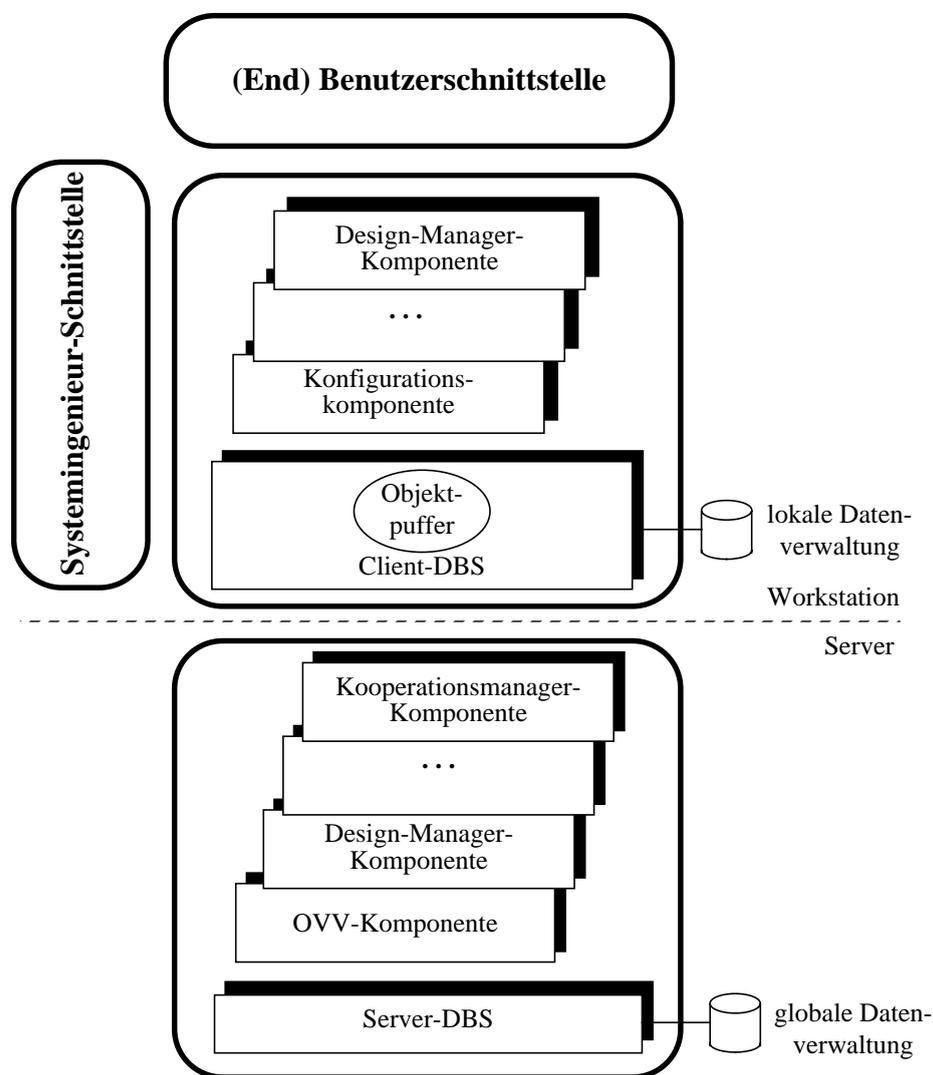


Bild 9: Verfeinerte Systemarchitektur eines integrierten Ingenieursystems

10. Ein Beispiel hierfür sind Absprachen zwischen Entwerfern, die in einem zwanglosen Gespräch in einer Arbeitspause getroffen werden.

turen zur Darstellung der Entwurfsumgebung sukzessive herausgearbeitet. Dabei wurde insbesondere auf die Repräsentation der Entwurfsobjekte (und deren strukturelle Information wie Versionen, Konfigurationen, Entwurfsobjekthierarchien usw.) sowie die Darstellung des dynamischen Entwurfsprozesses eingegangen, der über die eingeführten Aktivitätsträger modelliert wird. Aus Komplexitätsgründen abstrahiert das vorgestellte Entwurfsumgebungsmodell häufig von Detailinformationen.

Im Anschluß wurde ein Modell zur Unterstützung einer kooperativen Entwurfsvorgehensweise, ein Kooperationsmodell, vorgestellt. Ziel des Kooperationsmodells ist es, das Zusammenwirken zwischen den Aktivitätsträgern (bzw. den Ingenieuren, die diesen Aktivitätsträgern zugeordnet sind) geeignet zu unterstützen. Das Kooperationsmodell bietet dazu Mechanismen an, die zum einen ein wechselseitiges Verhandeln über die Auftragspezifikation der Aktivitätsträger dynamisch während des Entwurfsprozesses ermöglichen und zum anderen einen qualifizierten Austausch von vorläufigen Versionen der erstellten Entwurfsobjekte. Gegenüber der in den konventionellen DB-Anwendungen durch das Transaktionskonzept sehr restriktiv gehandhabten Isolation und Synchronisation der Anwendungen wird durch das Kooperationsmodell ein an den Anforderungen in technischen Anwendungen orientierte aufgeweichte Isolation und Synchronisation ermöglicht, die sich an den explizit über Features spezifizierten Eigenschaftsanforderungen der Entwurfsdaten sowie der Versionen der Entwurfsdaten orientiert.

Das Entwurfsobjekt-, das Entwurfsprozeß- und das Entwurfskooperations-Modell sind im wesentlichen für eine durchgängige Entwurfsunterstützung verantwortlich, so daß in diesem ersten Modellansatz auf eine Verfeinerung der übrigen Teilmodelle verzichtet wurde, die nun in weiterführenden Arbeiten zu erfolgen hat.

Im letzten Teil wurde auf die Systemarchitektur von DB-integrierten Ingenieursystemen zur Abbildung des Entwurfsumgebungsmodells näher eingegangen. Es wird ein strukturell objektorientiertes Datenbanksystem zur integrierten Verwaltung der Produktdaten und der Entwurfsumgebungsdaten eingebunden. Die unterlegte strukturell objektorientierte Datenmodell-Schnittstelle und die eher navigierende Verarbeitungsmodell-Schnittstelle sind nach dem bisherigen Kenntnisstand gut geeignet, den Anforderungen von technischen Anwendungen gerecht zu werden.

Es ist die Aufgabe weiterführender Arbeiten, das hier eingeführte Entwurfskooperations- und Entwurfsprozeß-Modell weiter zu verfeinern und auf eine formale Grundlage zu stellen. Des Weiteren wurde bislang noch kein konkretes Modell zur Behandlung von Fehler- und Konfliktfällen zwischen bzw. in den Aktivitätsträgern erarbeitet. Ein solches Modell ist jedoch für den erfolgreichen praktischen Einsatz von integrierten Ingenieursystemen eine notwendige Voraussetzung.

## **Danksagung**

Ich bedanke mich bei meinem akademischen Lehrer Prof. Dr. Th. Härder für die zahlreichen Anmerkungen beim Erstellen des Manuskriptes sowie bei meinen ehemaligen Arbeitskollegen,

die in zahlreichen, fruchtbaren Diskussionen wesentlich zur Klärung des vorgestellten Modells beigetragen haben.

## 8 Literatur

1. Anderl, R.: Integriertes Produktmodell. In: Zeitschrift für wirtschaftliche Fertigung und Automatisierung, Carl Hanser Verlag No. 12, 1989.
2. Berkel, T., Hübel, Ch., Jansen, H., Ruland, D., Siepmann, E., Wilkes, W.: Technische Informationssysteme zur Unterstützung von Entwurfsprozessen. In: Proceedings der GI-Fachtagung Datenbanksysteme in Büro, Technik und Wissenschaft, IFB, Springer-Verlag, Kaiserslautern, 1991.
3. Rehm, S., Raupp, T., Ranft, M., Längle, R., Härtig, M., Gotthard, W., Dittrich, K. R., Abramowicz, K.: Support for Design Processes in a Structurally Object-Oriented Database System; in Proc. 2nd int. Workshop on Object-Oriented Database Systems, 1988.
4. N. N.: EDIF Electronic Design Interchange Format Version 200, Electronics Industries Association, Washington, 1987.
5. N.N.: Framework Architecture Reference, Draft Proposal, Architecture Task Group of the CAD Framework Initiative, Version 0.4, Boulder, Oct. 1990.
6. Kenneth, W.F., Kleinfeldt, S., Kosarchyn, M., Perez, E.B.: Design Methodology Management - A CAD Framework Initiative Perspective. In: 27th Proceedings of the ACM/IEEE Design Automation Conference, Orlando, 1990.
7. Harrington, J.: Computer Integrated Manufacturing, Krieger Publishing Company, Malabar, Florida, 1973
8. Helberg, P.: PPS als CIM-Baustein, Berlin, 1987
9. Hellwig, H.-E., Goslar, Paulus, M.: Informationsverteilung in integrierten Produktionssystemen. In: VDI-Z Zeitschrift für integrierte Produktionstechnik, Bd.128, Nr.1/2, S.1-12, 1986
10. Härder, T., Hübel, Ch., Meyer-Wegener, K., Mitschang, B.: Processing and Transaction Concepts for Cooperation of Engineering Workstations and a Database Server. In: Data and Knowledge Engineering, 3, 87-107, 1988
11. Hübel, C., Käfer, W., Sutter, B.: Controlling Cooperation Through Design-Object Specification - a Database-oriented Approach. In: Proc. of the European Design Automation Conference, Brüssel, Belgien, März 1992
12. Härder, T., Meyer-Wegener, K., Mitschang, B., Sikeler, A.: PRIMA - A DBMS Prototype Supporting Engineering Applications. In: Proceedings of the VLDB-Conference, Brighton, 1987
13. Harrison, D.S., Newton, R., Spickelmier, R.L., Barnes, J.T.: Electronic CAD Frameworks. In: Proceedings of the IEEE, Vol. 78, No. 2, February 1990
14. Hübel, Ch., Sutter, B.: Aspekte der Datenbankanbindung in workstation-orientierten Ingenieur Anwendungen. In: Proc. der 19. Jahrestagung der Gesellschaft für Informatik, München, 1989
15. Haabena, J., Steinmüller, B.: The NMP-CADLAB Framework - a Common Framework for Tool Integration and Development. In: Proceedings European Design Automation Conference, March 1990
16. Hübel, Ch.: Ein Verarbeitungsmodell für datenbankgestützte Ingenieur Anwendungen in einer arbeitsplatzrechner-orientierten Ablaufumgebung, Dissertation, Kaiserslautern, 1992
17. Iochpe, C., Castro-Livi, M.-A.: Cooperation Support in Computer-Aided Design Environments, Research Report, CPGCC-UFRGS, Porto Allegre, Brazil, 1991 (submitted for publication)

18. N. N.: Initial Graphics Exchange Specification (IGES), Version 4.0, US Department of Commerce, National Bureau of Standard, June 1988
19. Käfer, W.: A Framework for Version-based Cooperation Control, in Proc. of 2nd Int. Symposium on Database Systems for Advanced Applications (DASFAA), Tokyo, Japan, April 1991
20. Käfer, W.: Geschichts- und Versionsmodellierung komplexer Objekte - Anforderungen und Realisierungsmöglichkeiten am Beispiel des NDBS PRIMA, Dissertation, Universität Kaiserslautern, 1992
21. Lockemann, P.C., Schmidt, J.W.: Datenbank-Handbuch, Berlin-Heidelberg- München, Springer, 1987
22. Mitschang, B.: Eine Molekül-Atom-Datenmodell für Non-Standard-Anwendungen - Anwendungsanalyse, Datenmodellentwurf, Implementierung, Informatik-Fachberichte, Band 185, Berlin, Springer, 1988
23. Maier-Rothe, Ch.: Computer Integrated Manufacturing: Bedeutung, Stand und Ausblick. In: Elektronische Rechenanlagen, 27. Jahrgang, Heft 3, S. 174-180, 1985
24. Phillips, R.W.: State Change Architecture: A Protocol for Executable Process Models. In: 22th Proceedings of the Annual Hawaii International Conference of System Sciences, Vol. III, Hawaii, 1989
25. Reisig, W.: Petrinetze, Springer-Verlag, Berlin, 1982
26. Scheer, A.-W.: Modellierung betriebswirtschaftlicher Informationssysteme. In: Zeitschrift Wirtschaftsinformatik, 32. Jahrgang, Heft 5, Oktober 1990
27. Scholz, B.: CIM-Schnittstellen; Konzepte, Standards und Probleme der Verknüpfung von Systemkomponenten in der rechnerintegrierten Produktion, München, Oldenburg, 1988
28. Sriram, D., Logcher, R., Fukuda, S. (eds): Computer Aided Cooperative Product Development, MIT-JSME Workshop, Proceedings, Springer-Verlag, Cambridge, Nov. 1989
29. Sutter, B.: Ansätze zur Integration in technischen Entwurfsanwendungen - angepasste Modellierungswerkzeuge, durchgängige Entwurfsunterstützung, datenorientierte Integration, Dissertation, Kaiserslautern, 1992
30. Wilson, P.R., Kennicott, P.R.: ISO STEP Baseline Requirements Document (IPIM), ISO TC184/SC4/WG1 N 284, ISO Draft Proposal No. 10103, 1988
31. VDI-Richtlinie 2222: Konstruktionsmethodik - Konzipieren technischer Produkte, Blatt 1 VDI-Verlag, Düsseldorf, 1977
32. van der Wolf, P., Sloof, G.W., Bingley, P., Dewilde, P.: Meta Data Management in the NELSI CAD Framework. In: 27th Proceedings of the ACM/IEEE Design Automation Conference, Orlando, 1990
33. Wächter, H., Reuter, A.: Grundkonzepte und Realisierungsstrategien des ConTract-Modells. In: Informatik Forschung und Entwicklung 5, 202-212 (1990)