

Ein Ansatz zu Graphischen Datenbanken und deren Verwendung in CAD-Systemen

Christoph Hübel, Bernhard Mitschang
Universität Kaiserslautern, Fachbereich Informatik
Erwin-Schrödinger-Straße
D-6750 Kaiserslautern

Kurzfassung

Die zentrale Aufgabe eines CAD-Systems ist die adäquate Unterstützung des Konstruktionsvorganges. Dazu müssen die Entwurfsobjekte sowohl graphisch repräsentiert als auch in einer sog. rechnerinternen Darstellung verwaltet werden. Heutzutage stehen für diese Aufgaben GKS-orientierte Graphiksysteme sowie Datenbanksysteme mit hierarchischer, netzwerkartiger bzw. relationaler Schnittstelle zur Verfügung. Am Beispiel des KUNICAD-Systems (ein datenbankbasierter 3D-Bauteilmodellierer) werden die Schnittstellen zwischen Graphik-, Anwendungs- und Datenhaltungskomponente aufgezeigt und analysiert. Dabei wird die Wichtigkeit der Kopplung von Graphik- und Datenhaltungskomponente erkannt. Als unmittelbare Konsequenz daraus, wird die Integration von Graphik- und DB-System zu einem Gesamtsystem, dem Graphischen Datenbanksystem, vorgeschlagen sowie dessen Architekturmerkmale und charakteristischen Eigenschaften vorgestellt.

1. Einleitung und Motivation

Die Datenverarbeitung im Bereich der sog. "nicht-kommerziellen" Anwendungen ist geprägt durch die Forderung nach Rechnerunterstützung beim Entwickeln, Entwerfen, Konstruieren und Zeichnen - kurz CAD (computer aided design) genannt. Als primäre Anwendungsgebiete sind die ingenieurwissenschaftlichen Disziplinen wie Maschinenbau, Elektrotechnik, Bauingenieurwesen und Architektur zu nennen. Aus Gründen der Kostenkontrolle, der Konstruktionsökonomie, der einheitlichen Erfassung und Darstellung der Entwurfsdaten sowie deren Austauschbarkeit wird in diesen Gebieten ein starker Zwang zur Einführung von CAD-Systemen ausgeübt.

Die Grobarchitektur eines CAD-Systems, wie sie in allen existierenden Systemen wiederzufinden ist, wird in Bild 1 skizziert. Die drei wesentlichen Komponenten werden repräsentiert durch das konkrete Anwendungssystem (AWS) sowie durch das Datenhaltungssystem (DHS) und das Graphiksystem (GS). Die wichtigsten Gründe für solch eine Subsystembildung sind Portabilität, Erweiterbarkeit, Entwicklungskosten, Integration mit anderen Anwendungen etc.

Durch diese Komponentenbildung werden gleichzeitig drei Subsystemschnittstellen definiert. Die Datenhaltungsschnittstelle ermöglicht die Verwaltung der Entwurfsobjekte in der rechnerinternen Darstellung. Kontrolliert von dem Anwendungssystem werden die Entwurfsobjekte gespeichert, gelöscht, geändert oder auf Verlangen bereitgestellt. Die Graphikschnittstelle erlaubt deren graphische Darstellung, ebenfalls gesteuert durch das Anwendungssystem. Die Schnittstelle zwischen DHS und GS vereinfacht die graphische Ausgabe der Entwurfsobjekte durch eine direkte Umsetzung von der rechnerinternen zur graphischen Darstellung. Die konkreten Auslegungen dieser Schnittstellen bestimmen die Eigenschaften des Gesamtsystems in erheblichem Maße. Wir unterscheiden daher zwischen einer direkten und einer indirekten Subsystemkopplung. Dabei bedeutet direkte Kopplung auch direkte Verarbeitung mittels der verfügbaren Schnittstellenfunktionen. Zusätzlich erlaubt die indirekte Kopplung das Herausholen der Daten, die dann lokal verarbeitet und nach Beendigung der Manipulation wieder zurückgegeben werden. Diese beiden unterschiedlichen Verarbeitungskonzepte haben unmittelbare Auswirkungen sowohl auf die Konsistenz der Daten als auch auf die Effizienz der Verarbeitung. Eine weitere charakteristische Schnittstelleneigenschaft ist die Abstraktionsmächtigkeit, d.h. die Mächtigkeit der angebotenen Funktionen auf den modellierten Objekten.

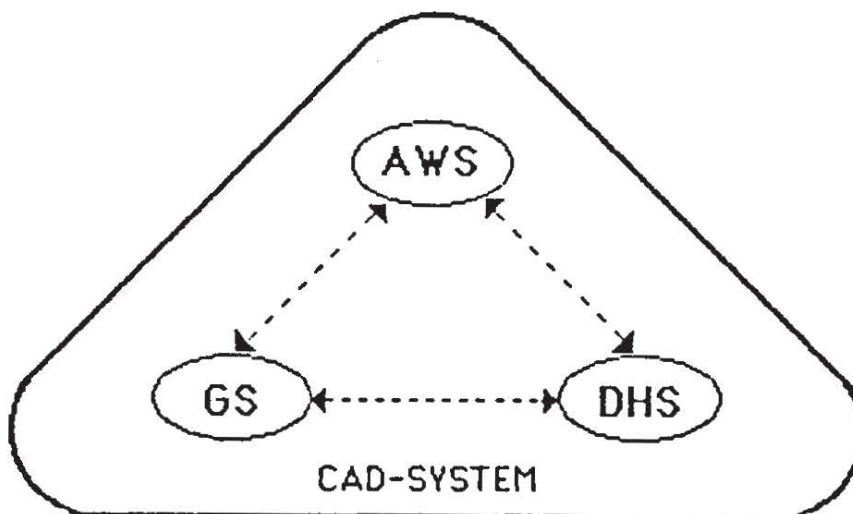


Bild 1: Grobarchitektur eines CAD-Systems

Versucht man die hier aufgezeigten Subsysteme und deren Schnittstellen in den existierenden CAD-Systemen wiederzufinden, so ist dies nur sehr schwer möglich, da die einzelnen Komponenten meistens zu einem komplexen Gesamtsystem verschmolzen sind. Die ersten kommerziell verfügbaren CAD-Systeme setzten sowohl bzgl. der graphischen Ein-/Ausgabe als auch bzgl. der Datenhaltung auf sehr niedrigen Schnittstellen auf. Mit der Entwicklung und Standardisierung des Graphischen Kernsystems (GKS, siehe [GKS83]) wurde zunächst die Portabilität bzgl. der graphischen Systemkomponente erreicht. Parallel zur Entwicklung von GKS wurde versucht, auch in bezug auf die Datenhaltung eine größere Unabhängigkeit zu erlangen. Der erste Schritt dazu war die Einführung einheitlicher Datenstrukturen und Dateiformate (z.B. IGES, siehe [En84]), die

zwar den Datenaustausch zwischen verschiedenen CAD-Systemen unterstützen, aber weiterhin keinerlei Konsistenzsicherungsmaßnahmen sowie nur eine geringe Unabhängigkeit zwischen der Datenhaltung und dem AWS gewährleisten.

Dies und die ständig fortschreitende Expansion und Integration der Anwendungen führt zum Einsatz von Datenbanksystemen (DBS). Damit können viele der Nachteile von dateiorientierten CAD-Systemen wie übermäßige Datenredundanz, fehlende Integritätskontrolle, mangelnde Datenstrukturierung usw. vermieden sowie Standardisierung, Austausch und Übertragung von komplexen Datenstrukturen verbessert werden. Zusätzlich werden ebenfalls wichtige Aufgaben wie Kontrolle des Mehrbenutzerbetriebs, Sicherung der Daten, Strukturierung der Verarbeitung mittels des Transaktionskonzeptes usw. vom DBS automatisch übernommen. Bisher sind DB-gestützte CAD-Systeme kaum verfügbar, da die auf dem Markt befindlichen kommerziellen DBS, für die erforderlichen Aufgaben denkbar schlecht gerüstet sind. Das liegt vor allem an ihrem mangelnden Anwendungsbezug bei der Datenmodellierung und an ihren für die Konstruktionsaufgaben überhaupt nicht geeigneten DB-Operationen. Besser geeignete DBS sollten daher eine mehr anwendungsbezogene, abstraktere und stärker objektorientierte Schnittstelle aufweisen [HR85]. Zusammen mit einer interaktiven Graphikunterstützung sind dann die Voraussetzungen gegeben die Akzeptanz solcher CAD-Systeme bei der Anwendung für kreative Entwurfsarbeiten zu erhöhen.

Da sich sowohl aus Aufwandsgründen als auch aus Mangel an anwendungsspezifischer DB-Erfahrung eine vollständige Neuentwicklung eines DB-gestützten CAD-Systems verbietet, wurde von uns eine Prototyp-Entwicklung durchgeführt, die sich bzgl. der Graphik- wie auch der DB-Komponente auf schon existierende Systeme abstützt. Dabei werden nicht die Leistungs-, sondern mehr die funktionalen Aspekte in den Vordergrund gestellt. Im nachfolgenden Kapitel 2 wird dieser Prototyp, ein datenbankgestütztes geometrisches Modellierungssystem für Werkstücke, beschrieben. Besonderes Augenmerk wird dabei auf die Systemarchitektur und auf die Schnittstellen zum DB- und zum Graphiksystem gelegt. Eine allgemeine Bewertung und erste Konsequenzen bzgl. unserer Prototyp-Implementierung werden dann in Kapitel 3 zusammengefaßt, bevor im letzten Kapitel die wichtigsten Schlußfolgerungen zusammen mit einem Ausblick auf weitere Arbeiten die Analyse abschließen.

2. Das KUNICAD-System

Das KUNICAD-System [HHL86] ist ein datenbankgestütztes, volumenorientiertes Modellierungssystem für Werkstücke, das als Kernalgorithmus für die geometrische Modellierung ein volumenorientiertes Verfahren (CSG, constructive solid geometry [RV82]) einsetzt. Dabei werden Bauteile als Polyeder dargestellt, die aus Basiskörpern unter Anwendung zugelassener Operationen gewonnen werden. Intern werden zusätzlich Strukturen nach dem Begrenzungsflächenmodell (BREP: boundary representation [RV82]) gehalten, die automatisch aus den CSG-Strukturen abgeleitet und, falls erforderlich, nachgeführt werden. Das KUNICAD-System stellt ein sog. körperorientiertes "dual

representation"-Modellierungssystem dar, da die rechnerinterne Darstellung der Konstruktionskörper einer CSG/BREP-Kombination entspricht. Die primäre, CSG-artige Darstellung erlaubt die direkte und einfache Abbildung des Anwendermodells, während sich die optionale BREP-Darstellung, wegen ihrer Nähe zur Graphik-Repräsentation, u.a. sehr gut für Ausgabezwecke eignet. Das KUNICAD-System besteht ungefähr aus 20000 Anweisungszeilen und wurde in den Programmiersprachen PL/I, FORTRAN 77 und PASCAL implementiert. An Software-Paketen werden das Datenbankverwaltungssystem UDS [UDS84] und das Graphikpaket PLOT10 [PL77] benutzt. Damit ist das KUNICAD-System auf allen BS2000-Rechenanlagen der Firma Siemens mit entsprechender Graphik-Hard- und Software lauffähig.

2.1 Die Systemarchitektur

Zu den Hauptkomponenten unseres datenbankbasierten 3D-Bauteilmodellierers gehört zum einen die Graphikkomponente GS, die Hilfsmittel zur Realisierung von ergonomischen Benutzerschnittstellen zur Verfügung stellt. Ein anderer Teil des Systems, die Datenhaltungskomponente DHS, verwaltet die rechner-

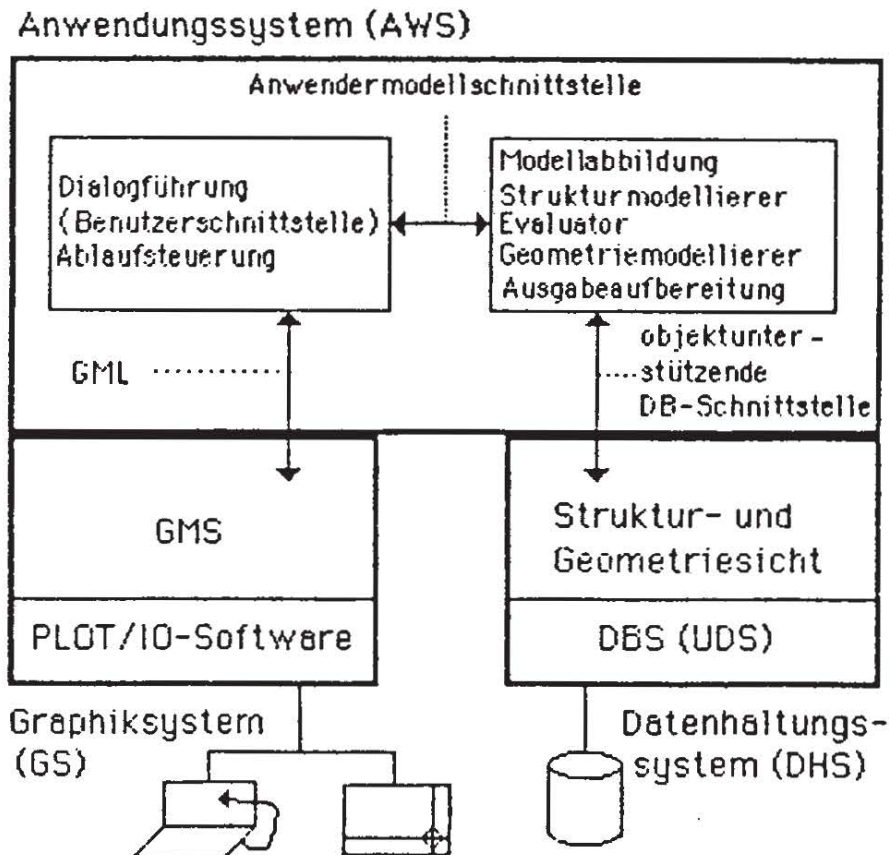


Bild 2: Architektur des KUNICAD-Systems

interne Darstellung der Konstruktionsdaten. Die dritte Schicht (AWS) bildet die "Anwendersicht" auf die rechnerinterne "Datensicht" ab (vgl. Bild 1). Die Architektur des KUNICAD-Systems wird in Bild 2 dargestellt.

Die Graphikkomponente (GMS, graphic manipulation system) erlaubt dem Anwendungssystem die Ein- und Ausgabe graphischer und textueller Information. Die Funktionalität der Graphikschnittstelle (GML, graphic manipulation language) entspricht der von GKS Level 2b. Der Modul Dialogführung ist Bestandteil der Anwendungskomponente. Seine Hauptaufgabe besteht in der Vorverarbeitung der Ein- und Ausgabe sowie in der Entkopplung der implementierten Benutzerschnittstelle von der darunterliegenden Anwendermodell-schnittstelle (auf der Ebene des Anwendermodelles spielt es keine Rolle, ob die Benutzerschnittstelle menü- oder komandogesteuert arbeitet bzw., ob eine Selektion durch eine "Pick"-Eingabe möglich ist oder nicht). Zusammen mit der Ablaufsteuerung werden die notwendigen Aktivitäten innerhalb der Modellabbildung angestoßen und kontrolliert. Die weiteren Moduln der Anwendungskomponente dienen der Modellabbildung, d.h. der Projektion der Anwendermodell-Objekte auf die Datenstrukturen und die damit assoziierten Operationen einer im System darunterliegenden Schnittstelle (Geometrie- und Struktursicht). Der Modul Strukturmodellierer verwaltet und aktualisiert die organisatorischen und strukturellen Konstruktionsdaten. Hierzu verwendet er Funktionen des Struktursichtmoduls an der Schnittstelle zur Datenhaltungskomponente. Der Modul Ausgabeaufbereitung übernimmt vorwiegend die Transformation der gespeicherten 3D-Information in eine 2D-Darstellung, die unmittelbar auf einem Graphik-Terminal sichtbar gemacht werden kann. In dem Modul Geometriemodellierer wird die eigentliche Verarbeitung der Geometriedaten durchgeführt. Geometrische Körperbeschreibungen können durch reguläre Mengenoperationen zu komplexeren Körperbeschreibungen aggregiert werden. Die Datenhaltungskomponente stellt im Modul Geometriesicht die hierzu erforderlichen Grundfunktionen bereit. Der Evaluator schließlich repräsentiert das Bindeglied zwischen der strukturellen (CSG) und der geometrischen (BREP) Interndarstellung eines Werkstücks. Dieser Programmbaustein steuert die Berechnung der BREP-Darstellung eines Körpers aus seiner CSG-Repräsentation. Hiermit ist es möglich, vom Aufbau des CSG-Baumes ausgehend, erst zu einem späteren Zeitpunkt, zu dem eine geometrische Beschreibung tatsächlich benötigt wird, diese auch herzuleiten. Die Moduln Struktur- und Geometriesicht der Datenhaltungskomponente repräsentieren die sogenannte objektunterstützende DB-Schnittstelle und stellen die Implementierung einer "Zusatzebene" [Mi84] auf dem zugrundeliegenden Datenbanksystem (UDS) dar. Sie stellen Objekte bereit, die von den Moduln der Modellabbildung auf vorteilhafte Art und Weise zu nutzen sind.

2.2 Die Anwendungskomponente

Die zentrale Aufgabe eines CAD-Systems ist die adäquate Unterstützung des Konstruktionsvorganges. Daher beinhaltet das Anwendermodell unseres KUNICAD-Systems allgemeine konstruktionsunterstützende Konzepte und erzielt zudem eine Abstraktion von der konkreten Benutzerschnittstelle bzw. von der vorhandenen Graphikkomponente. Durch die Unterscheidung von drei Körperarten

(Basis-, Bauteil- und Baugruppenkörper) im Anwendermodell wird eine geeignete Konstruktionssemantik ermöglicht. Die Typisierung/Instanziierung erlaubt den Übergang von der Ausprägung zum Typ bzw. umgekehrt vom Typ zur Ausprägung. Damit ist die Einführung von neuen und komplexeren Körperklassen möglich. Ebenso liefert das Konzept der Typen- bzw. Ausprägungsparametrisierung eine zusätzliche Flexibilisierung des Konstruktionsprozesses. Schließlich stellt die Konstruktionsumgebung einen benutzerkontrollierten organisatorischen Rahmen zur Verfügung, in dem Körperausprägungen und -typen nach bestimmten Verarbeitungsregeln entwickelt werden können.

Aufbauend auf diesem Anwendermodell lassen sich nun ergonomische Benutzerschnittstellen realisieren, die sich dem Arbeitsablauf und den Denkgewohnheiten im CAD-Bereich anpassen und damit die Akzeptanz solcher Systeme für kreative Entwurfsarbeiten entscheidend beeinflussen.

Graphikoperationen		Operationen auf Ausprägungen	
CLH	Clear löscht den Inhalt des Graphic-Windows	BI	BUILD Instance zur Durchführung der Zusammensetzung
PLT	Plot gibt den Inhalt des Graphic-Windows auf Plotter aus	ORA	ORange entspricht der kommandofolge OO-OO
CSZ	Set Character Size (für den Text auf dem Bildschirm)	ON	ONject erweitert eine Baugruppe
SGW	Set Graphic-Window	OPI	OPIate Instance zur Durchführung der Instanziierung
SPS	Set Plot Size	DDO	DISConnect reduziert eine Baugruppe
SVP	Set View Point für die Projektionsberechnung	DEI	DElete Instance
		LI	LIst local Instance
		MCI	MODify gleichbedeutend mit SEL, jedoch hier zusätzlich noch die Möglichkeit die gewählte Komponente zu ändern
Operationen zur Steuerung von Bearbeitungsphasen		MOI	MOve Instance
BOS	Begin Of Session	OPI	OPIrate Instance zur Durchführung von Durchschnitt, Vereinigung oder Differenz
EOS	End Of Session	REI	REname Instance
RCS	Reset Of Session	ROI	ROtate Instance
		SEL	SElect zum Auswählen einer Komponente einer Baugruppe
		SHI	SHOW Instance
Operationen einer Konstruktionsumgebung		sonstige Operationen	
BIC	Begin Of Construction	DES	DEscribe erlaubt die Beschreibung des Konstruktionsobjektes
EOC	End Of Construction	ED	ED beendet die Terminalsitzung
LCF	List Construction Environment	PARAM	define PARAMeter definiert formale Parameter
SCE	Show Construction Environment	VAR	define VARiable definiert Variablen die Schreibaufwand einsparen helfen
Operationen auf Typen			
DET	DElete Type		
DGT	DElete Global Type		
LGT	LIst Global Types		
LIT	LIst Local Types		
MAT	MAke Type zur Durchführung der Typisierung		
RET	REname Type		
SGT	SHOW Global Type		

Tabelle 1: Funktionen und Kommandos der Benutzerschnittstelle

Die Dialogführung implementiert mit Hilfe der Graphikkomponente die Benutzerschnittstelle des CAD-Systems, die letztendlich nur eine syntaktische Aufbereitung der abstrakten Anwendermodellschnittstelle mittels der konkret vorhandenen graphischen Hard- und Software darstellt. In Tabelle 1 sind die Funktionen und Kommandos dieser Benutzerschnittstelle tabellarisch aufgeführt.

2.3 Die Datenhaltungskomponente

Das Datenhaltungssystem besteht aus drei Teilen, dem Datenbanksystem (UDS), dem Modul Geometriesicht und dem Modul Struktursicht. Geometrie- und Struktursicht bilden die Schnittstelle zum Anwendungssystem. Wir wollen an dieser Stelle keine vollständige Beschreibung aller Operationen geben, sondern vielmehr anhand einiger Beispiele eine Charakterisierung dieser Schnittstelle und damit auch der Datenhaltungskomponente durchführen (eine detaillierte Beschreibung ist in [HFLM86] zu finden).

Datenbankschema unserer CAD-Anwendung

Die durch das Anwendermodell beschriebene Information muß mittels des vom Datenbanksystem UDS [UDS84] zur Verfügung gestellten netzwerkorientierten Datenmodells dargestellt werden. In Bild 3 sind die wichtigsten Satztypen des zugehörigen DB-Schemas, eingeteilt in zusammengehörige Informationseinheiten, aufgelistet.

Organisationsteil:	Konstruktionsumgebung Synonym	Geometrieteil:	BREP-Darstellungsschema Fläche Kante Punkt
Strukturteil:	Strukturknoten Position Beschreibung Parameter		UmgebungsInformation

Bild 3: Schemainformation des KUNICAD-Systems

Der Organisationsteil beschreibt den Zustand der einzelnen Konstruktionsumgebungen. Für jede definierte Umgebung existiert eine Satzausprägung vom Typ Konstruktionsumgebung, dem eine Menge von Synonym-Sätzen zugeordnet ist, die die Variablen- und Parameterinformation enthalten.

Der Strukturteil realisiert ein Netz von Knoten, durch das die "CSG"-Struktur des zugehörigen Typ- oder Ausprägungskörpers festgelegt ist. Ein Operatorattribut bestimmt, ob der Körper durch Vereinigung, Durchschnitt, Differenz oder Addition der assoziierten Körper entstanden ist. Die zugeordneten Satzausprägungen vom Typ Parameter repräsentieren die aktuellen Parameter einer Instanzierung oder Zusammensetzung. Ergänzend können noch Satzausprägungen vom Typ Beschreibung für erläuternden Text sowie vom Typ Position zugeordnet werden. Der Position-Satz enthält eine Transformationsmatrix, in die alle Verschiebungen, Rotationen und Skalierungen des Körpers eingearbeitet werden.

Der Geometrieteil benutzt zur Darstellung der geometrischen und topologischen Körperdaten das Begrenzungsflächenmodell, in dem die Körpergeometrie durch die Körpermantelflächen charakterisiert wird. Die Flächen selbst sind durch die begrenzenden Kanten und diese wiederum durch ihre Randpunkte bestimmt. Der Punkt-Satztyp besitzt die Punktkoordinaten, die die eigentliche geometrische Information beinhalten. Zur genaueren Beschreibung der geometrischen Umgebung eines Punktes bzw. einer Kante bezogen auf einen Körper werden Flächen-, Kanten- und Eckpunkt-Umgebungsinformationen explizit verwaltet. Diese dienen der einfacheren Verarbeitung von Körpern, die sich gegenseitig berühren.

Objektunterstützende DB-Schnittstelle

Als objektunterstützende Schnittstelle (OSS) bezeichnen wir die Schnittstelle zum Anwendungssystem, die mit Hilfe der Struktur- und Geometriesicht gebildet wird. Damit soll zum Ausdruck kommen, daß der "Gegenstand der Verarbeitung" (Verarbeitungseinheit, VE) an dieser Schnittstelle von komplexerer Natur sein kann als herkömmliche DB-Sätze. Alle OSS-Operationen nehmen Bezug auf sog. explizite Cursor. Unter einem expliziten Cursor verstehen wir eine spezifische Referenz auf eine Satzausprägung (im folgenden als Ankersatz bezeichnet) und deren "unmittelbaren Nachbarausprägungen" in der Datenbank, d.h. eine Referenz auf alle mit dem Ankersatz (über Set-Beziehungen) assoziierten Satzausprägungen. Er tritt quasi als "Objektrepräsentant" auf, da die Programme des Anwendungssystems nicht zwischen dem Cursor und der damit assoziierten Verarbeitungseinheit unterscheiden.

VE	Anker	konstituierende Satztypen
BREP	BREP	Fläche, Kante, Punkt, Umgebungs- information
Fläche	Fläche	Kante, Punkt, Umgebungsinformation
Kante	Kante	Punkt, Umgebungsinformation, Fläche
Punkt	Punkt	Umgebungsinformation, Fläche
Umgebung	Umgebung	Umgebungsinformation, Fläche
Konstruktions- umgebung	Konstruktions- umgebung	Synonym, zugehöriger Struktur- und Geometrieteil
Synonym	Synonym	_____
Instanz/Typ	Strukturknoten	Position, Beschreibung, Parameter, zugehöriger Geometrieteil

Bild 4: Verarbeitungseinheiten der objektunterstützenden DB-Schnittstelle

Die VE der objektunterstützenden Schnittstelle, zusammengestellt in Bild 4, korrespondieren mit dem zugrundeliegenden DB-Schema. So beinhaltet die Geometriesicht die BREP-VE, die sich auf der Schemaebene aus dem gesamten Geometrieteil zusammensetzt; auf Ausprägungsebene sind mit ihr alle diejenigen Satzausprägungen assoziiert, die mit einem einzigen BREP-Ankersatz über entsprechende Set-Beziehungen verbunden sind. Weitere VE sind Fläche-, Kanten-, Punkt- und die Umgebungs-VE. Sie bestehen aus Satzstrukturen, die im wesentlichen durch eine Reihe konstituierender Satztypen sowie einem

Ankersatz bestimmt sind. Die VE der Struktursicht umfassen jeweils größere Teile des DB-Schemas. Instanz- und Typ-VE sind repräsentiert durch die zugehörige Strukturbeschreibung und den assoziierten Geometrieteil. Die VE Konstruktionsumgebung enthält zusätzlich noch die Satzausprägung von Konstruktionsumgebung als Anker sowie die zugehörigen Synonym-Sätze. Lediglich die Synonym-VE ist einfach aufgebaut und besteht nur aus dem Ankersatz.

An der objektunterstützenden DB-Schnittstelle sind drei unterschiedliche Operationstypen zu erkennen:

- satzorientierte Operationen: Der Gegenstand der Verarbeitung ist genau ein Satz; Bsp.: "Gib Flächenattribute", "Setze Kantenattribute", "Gib nächste Kante der Fläche", etc.
- strukturorientierte Operationen: Der Gegenstand der Verarbeitung ist eine strukturierte Menge heterogener Sätze. Die Verarbeitung ist geprägt durch das "Herausziehen" solcher Satzmenge aus der bzw. ihr "Einbringen" in die DB; Bsp.: "Gib Punkt-Umgebungsinformation", "Setze Flächen- und Umgebungsinformation", etc.
- objektorientierte Operationen: Der Gegenstand der Verarbeitung ist ebenfalls eine strukturierte Menge heterogener Sätze. Allerdings ist die Verarbeitung durch eine "semantisch höherwertige" Aktivität bestimmt; Bsp.: "Mischen zweier BREP-Darstellungen", "Hinzufügen eines Durchdringungspunktes", etc.

Der Übergang zwischen den Operationstypen erscheint fließend. So gibt es Operationen, die Aspekte mehrerer Typen aufweisen. Demnach dienen die oben vorgestellten Operationstypen nicht zur vollständigen Klassifizierung, wohl aber zur Charakterisierung der Verarbeitung an dieser Schnittstelle.

2.4 Die Graphikkomponente

Unser Graphiksystem (GMS) basiert auf einem einfachen Vektorgraphik-System (PLOT/10-Software [P177] auf Tektronix-Graphikterminals der Serie 4010), welches im wesentlichen nur graphische linienorientierte Ausgabe sowie einfache textuelle Benutzerinteraktion unterstützt. Es bietet zusätzlich die Möglichkeit den Bildschirm mit Hilfe von Windows in mehrere Bereiche aufzuteilen. Jedes Window kann einzeln angesprochen werden und erlaubt verschiedene Abbildungsmaßstäbe vom Weltkoordinaten- auf das Bildschirmkoordinatensystem bereitzustellen. Es wird zwischen Graphik- und Text-Windows unterschieden, wobei für das Text-Window zusätzlich ein "Scrolling" vorgesehen ist. Als Initialisierungsaufwurf dient die Funktion OPENWORKSTATION. Für das weitere Arbeiten müssen dann ein oder auch mehrere Text- oder Graphik-Windows definiert werden. Danach stehen die übrigen in Tabelle 2 aufgeführten Funktionen zur Verfügung. Den Abschlußaufwurf bildet die Funktion CLOSEWORKSTATION, womit auch gleichzeitig alle Windows gelöscht werden. Die Funktionalität der Graphikschnittstelle entspricht in wesentlichen Teilen der GKS-Schnittstelle Level 2b. Dadurch ist der Austausch der hier beschriebenen Graphikkomponente durch ein GKS-orientiertes Graphiksystem unter vollständiger Beibehaltung der vorhandenen GML sehr leicht möglich. Außerdem wird der

Übergang zu einer mehr GKS-orientierten Graphikschnittstelle in gleichem Maß erleichtert.

Bildstrukturierungsfunktionen		Eingabefunktionen	
SETPGRAPHICWINDOW	Anlegen eines Graphic-Window	SETTEXT	Texteingabe
SETTEXTWINDOW	Anlegen eines Text-Window		
DELWINDOW	Löschen des spezifizierten Window		
CLEARWINDOW	Löschen des Inhalts des spez. Window		
CHANGIEWINDOW	Variation der Window-Parameter		
Ausgabefunktionen		Steuer- und Informationsfunktionen	
DRAW	Zeichnen einer Linie, bezogen auf die aktuelle Cursorposition	CHARSIZE	Bestimmt die Größe der Textzeichen
LINES	Zeichnen einer Linie zwischen zwei Koordinatenpunkten	GETLOCATION	Liefert die aktuellen Weltkoordinaten
BLOCK	Zeichnen einer "Linienfolge"	SETLOCATION	Positioniert den Cursor auf spezifizierte Weltkoordinaten
POINT	Zeichnen eines Punktes an spezifizierter Koordinatenposition	OPENWINDOWSTATION	Initialisierung
TEXT	Textausgabe an aktueller Cursorposition	CLOSEWINDOWSTATION	Löschen aller Windows und Zusatzinformationen
ORLFE	Positionieren des Cursors auf den Anfang der neuen Zeile		
		Plotterfunktion	
		PLOT	Ausgabe des spezifizierten Windows auf dem Plotter

Tabelle 2: Funktionen des Graphiksystems

3. Bewertung und Konsequenzen aus der KUNICAD-Entwicklung

Nachdem wir in Kapitel 2 eine knappe Beschreibung des KUNICAD-Systems gegeben haben, wollen wir nun eine Bewertung und Analyse der in der Einleitung skizzierten Schnittstellen zwischen den drei Subsystemen Anwendungskomponente, Datenhaltungskomponente und Graphikkomponente durchführen.

3.1 Die Schnittstelle zwischen Anwendungs- und Datenhaltungskomponente

Die Schnittstellenproblematik zwischen der Anwendungs- und Datenhaltungskomponente ist geprägt durch die Diskrepanz der Objekte des Anwendermodells zu denen des Datenmodells sowie den mit ihnen jeweils assoziierten Operationen. Modellabbildung und DBS-Zusatzebene dienen zur Überbrückung dieser nicht unerheblichen Modelldiskrepanz. Die Operationen auf den Objekten des Anwendermodells werden unmittelbar auf Operationen bzw. Operationsfolgen des Datenmodells abgebildet. Die Gangbarkeit des eingeschlagenen Weges, der direkten Kopplung einer CAD-Anwendung mit einem um eine Zusatzebene erweiterten konventionellen Datenbanksystem kann durch die Analyse der Aufrufhäufigkeiten an unterschiedlichen Modellebenen bewertet werden [Hü86]. Messungen ergaben für die Generierung der Begrenzungsflächendarstellung eines 52-flächigen Polyeders selbst nach verschiedenen Optimierungsschritten etwa 14800 Aufrufe von Datenmodelloperationen. Dieser Aufwand ist für realistische Anwendungen nicht zu tolerieren.

DB-Operationen für CAD-Anwendungen zeichnen sich durch häufiges Rereferenzieren innerhalb der jeweils aktuellen Verarbeitungseinheit aus. Die Pufferung der Verarbeitungseinheit erlaubt die Nutzung dieser Zugriffslokalität sehr "nahe" an ihrem "Entstehungsort". Eine weitere Verbesserung scheint erreichbar durch die Erweiterung bestehender Datenmodelle um mächtigere Strukturierungsmittel. Hierzu gibt es zahlreiche Vorschläge aus der Literatur [BB84, Da86, HM85, Mi85, Sch85]. Will man annähernd ein akzeptables Leistungsverhalten erzielen, so halten wir, einhergehend mit der Schaffung adäquater Datenmodellierungsmöglichkeiten, einen Wechsel des Verarbeitungs- und Darstellungsparadigmas für unumgänglich [HHL86]. Einmal sollte dem Anwendungsprogramm direkt eine heterogene, komplex strukturierte Satzmenge bereitgestellt werden können (dies impliziert eine Änderung des üblichen Subschema-Konzeptes). Daneben könnte die asynchrone bzw. parallele Verarbeitung der anfallenden DB-Operationen ermöglicht werden (dies impliziert eine veränderte Sicht auf die Transaktion als organisatorische Einheit der DB-Verarbeitung) [HHM85].

Zusammenfassend lassen sich nachstehende Forderungen an die Schnittstelle zwischen Anwendungs- und Datenhaltungskomponente formulieren:

- Schaffung angepaßter Datenmodelle zur Unterstützung der anwendungsgerechten Modellierung von Objekten des Anwendermodells.
- Bereitstellung heterogener und strukturierter Satzmenge.
- Bereitstellung satzorientierter sowie strukturorientierter und objektorientierter Operationen.
- Ausnutzen der Verarbeitungslokalität.

3.2 Die Schnittstelle zwischen Anwendungs- und Graphikkomponente

An der Schnittstelle der Anwendungskomponente zum Graphiksystem wurden bislang keine quantitativen Untersuchungen gemacht; daher ist nur eine qualitative Bewertung möglich. Die Schnittstelle orientiert sich an der Funktionalität des GKS-Standards. Gewisse Einschränkungen betreffen die Eingabemöglichkeiten (nur String-Eingabe, keine Pick-Funktion). GKS unterstützt bislang in erster Linie Anwendungen aus dem 2D-Bereich, so daß gewissen 3D-Anforderungen nur mühsam und in unnatürlicher Weise entsprochen werden kann. Insbesondere erscheint uns die 3D-Unterstützung durch einen räumlichen Cursor und ein räumliches Fenster angebracht.

Auf der Darstellungsseite halten wir eine über das Segment hinausgehende hierarchische Strukturierungseinheit für wünschenswert. 3D-Objekte können z.B. als Struktur aus Flächen-, Kanten- und Punktdarstellungen beschrieben werden und sollten auch als solche an der Graphikschnittstelle definierbar sein. Bei der KUNICAD-Systementwicklung führte dieses Strukturierungsbedürfnis zur Einführung einer BLOCK-Funktion, mit deren Hilfe ganze "Linienfolgen" als Übergabeeinheit der Graphikkomponente bereitgestellt werden. Auch bei 2D-Anwendungen sind im allgemeinen die Darstellungsobjekte meist stärker unterteilt, so daß die Forderung nach komplexeren Strukturierungseinheiten als generelle Anforderung aufgefaßt werden kann.

Zudem ist durch die Bereitstellung dieser mengenorientierten Schnittstelle auch ein positiver Einfluß auf das Leistungsverhalten der Graphikkomponente zu erwarten.

Die folgende Zusammenstellung zeigt nochmals die wichtigsten Forderungen auf:

- Gezielte Unterstützung von 3D-Anwendungen durch raumbezogene Eingabemöglichkeiten sowie die Integration von "hidden-line"-, Transformations- und Projektionsverfahren.
- Hierarchische Strukturierung von Ausgabelementen.
- Mengenorientierte Ein-/Ausgabeoperatoren.

3.3 Die Schnittstelle zwischen Datenhaltungs- und Graphikkomponente

Die Schnittstelle zwischen DBS und GS wird oftmals nur wenig berücksichtigt bzw. überhaupt nicht realisiert, so auch bei der Entwicklung des KUNICAD-Systems. Eine direkte Folge hiervon besteht nun darin, daß die Ausgabe von Daten des DBS bzw. deren Eingabe stets über das AWS abgewickelt werden muß. So bewirkt das Kommando "Zeige Instanz (SHow Instance, SHI) zunächst eine Reihe von Leseoperationen des DBS, mit deren Hilfe das AWS die geometrische Beschreibung einer entsprechenden Baugruppenausprägung als Hauptspeicherstruktur aufbaut. Anschließend führt das AWS standardmäßige Verfahren zur Ermittlung der verdeckten Kanten und zur Projektion durch und übergibt dem GS schließlich eine Folge von Linien (als Übergabeeinheit) für die 2D-Beschreibung der ausgewählten Baugruppe. Ebenso übernimmt das AWS die alleinige Verantwortung für die Übereinstimmung der im GS dargestellten Objekte mit der entsprechenden Repräsentation im DBS. Im KUNICAD-System führt dies zu einem in gewissem Sinne unnatürlichen Änderungsgeschehen. Jede Modifikationsoperation des Anwenders unterteilt sich in einen Änderungsvorgang und eine anschließende Neuausgabe an das GS. Die wiederholte Ausgabe muß vom AP explizit erzwungen werden. Dies gilt auch für die Teile, die nicht von einer Änderung betroffen sind. Die oben geforderte über das Segmentkonzept hinausgehende hierarchische Strukturierungseinheit innerhalb des GS würde hier erlauben, nur die zu ändernden Strukturteile selektiv dem GS weiterzugeben. Die Kopplung eines GS mit einem DBS über eine entsprechende, direkte Schnittstelle bildet eine wesentliche Voraussetzung für die Behebung dieses Mißstandes (siehe [CMMS84]). Die wichtigsten Argumente für eine möglichst enge Kopplung beider Komponenten sind:

- Entlastung des AWS durch direkten Datenaustausch.
- Vermeidung unterschiedlicher Zustände der graphischen und nicht-graphischen Objektrepräsentationen.

Im weiteren wollen wir einige Vorstellungen von der Art der Schnittstelle und den charakteristischen Eigenschaften einer solchen Kopplung erläutern.

Aus obiger Diskussion der Systemschnittstellen ergeben sich gemeinsame Anforderungen an GS und DBS. Diese betreffen die Möglichkeit zur adäquaten Objektstrukturierung in beiden Systemen. Unterstellt man eine derartige objektorientierte GS- bzw. DBS-Erweiterung, so kann eine gegenseitige Angleichung der Objektrepräsentationen erfolgen. Mit anderen Worten, das erweiterte GS kann dann unmittelbar die Objekte, die von einem erweiterten

DBS modelliert werden können, akzeptieren. Eine sinnvolle Kopplung bzw. eine Integration beider Systeme zu einem "Graphik-Datenbanksystem" (GDBS) kann nun durch einen gemeinsamen Objektpuffer erfolgen.

Bild 5 soll dies an einem Beispiel verdeutlichen. Durch einen Selektionsoperator wird eine bestimmte Baugruppe im Objektpuffer installiert (1). Die folgende Display-Anweisung kann sich direkt auf diese Repräsentation beziehen, da die Interpretation der entsprechenden Körperstruktur von der GS-Erweiterung durchgeführt werden kann (2). Bei den nun ausgeführten Operationen handelt es sich zum einen um einfache Umlagerungen von Bauteilen (dies kann unabhängig von einem AWS geschehen). Zum anderen werden wesentliche Bauteildaten direkt durch das AWS modifiziert (3). Da die Baugruppenrepräsentation im Objektpuffer stets den aktuellen Zustand markiert, d.h. denjenigen Zustand, den der Benutzer am Graphikbildschirm erzeugt, kann für die folgende Update-Operation die entsprechende Objektpufferdarstellung direkt verwendet werden (4).

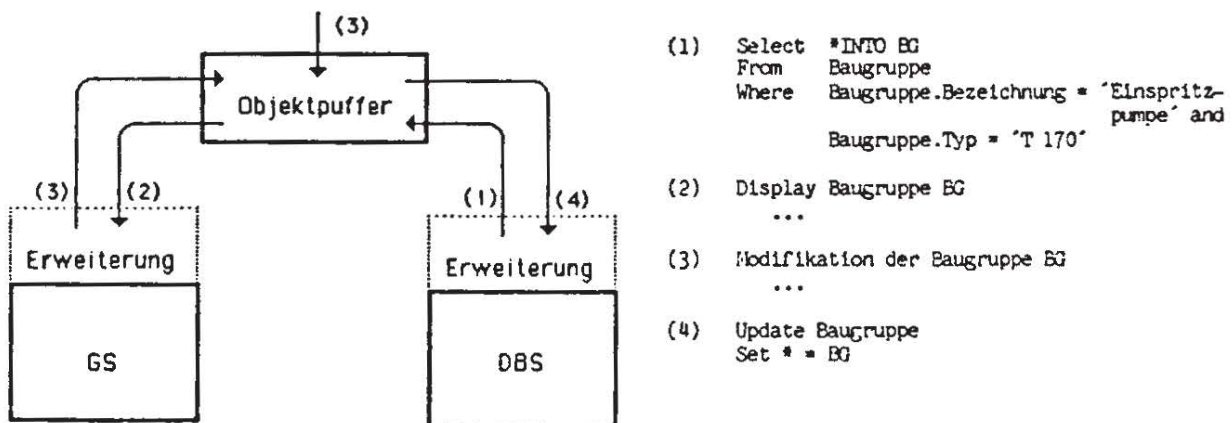


Bild 5: Kopplung GS-DBS über gemeinsamen Objektpuffer

Bild 6 skizziert unsere Vorstellungen von der Architektur eines GDBS-Kernsystems. Die wesentlichen Bausteine sind: erweitertes GS, erweitertes DBS, gemeinsamer Objektpuffer und eine übergreifende GDBS-Kern-Schnittstelle. Die beiden erweiterten Systembausteine stellen mengenorientierte Operatoren bereit, die eine Verarbeitung strukturierter heterogener Objekte erlauben. Dem GS sind Standard-Objekte aus dem 3D-Bereich (z.B. Begrenzungsflächendarstellung eines Körpers) bekannt. Typische Operationen wie Transformation, Projektion und die Ermittlung sichtbarer Kanten sind vollständig in das erweiterte GS integriert. Darüberhinaus könnten Objekttypen im Schema definierbar und ihre Beschreibung als Metadaten sowohl dem DBS als auch dem GS verfügbar sein. Die hier vorgeschlagene GS-Erweiterung stellt eine Verallgemeinerung der "anwendungsorientierten Schicht" aus [ARW83] dar.

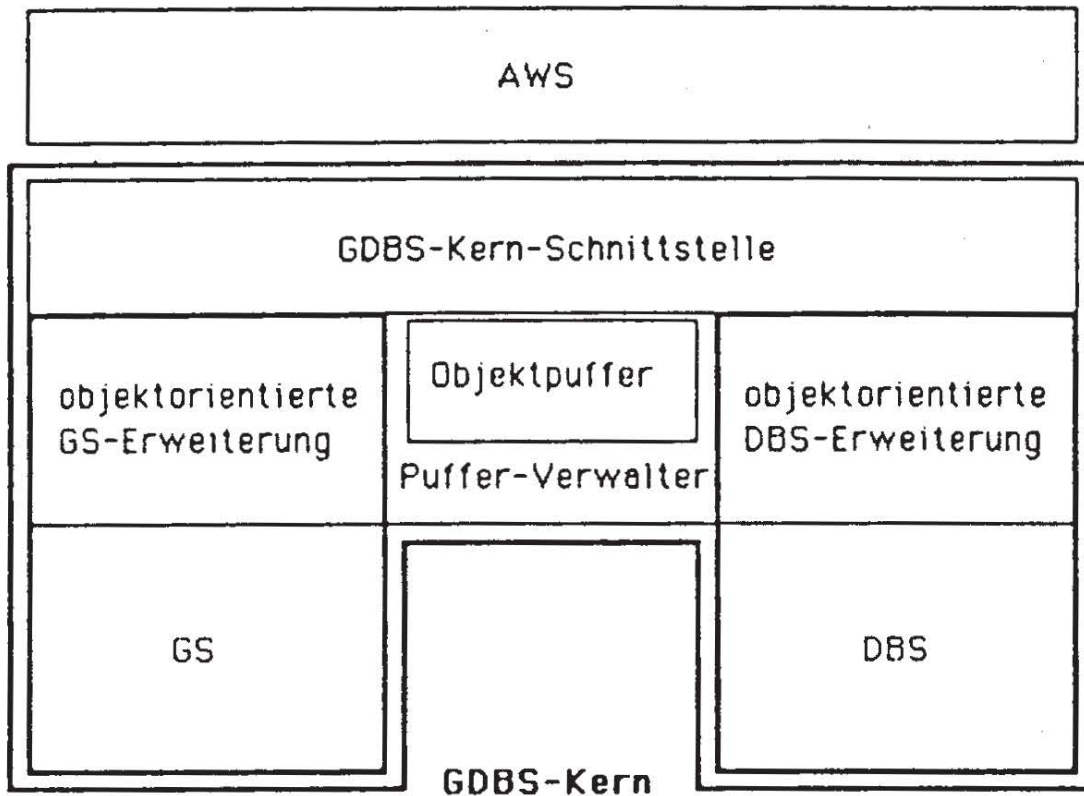


Bild 6: GDBS-Kern-Architektur

Die Kopplung der beiden Systeme erfolgt über einen gemeinsamen Objektpuffer. Der Zugriff auf den Puffer muß über einen Objektpuffer-Verwalter synchronisiert werden. Der übergeordnete Baustein, die GDBS-Kern-Schnittstelle, dient der zusätzlichen Entkopplung des AP von den darunterliegenden Systemkomponenten. Die Funktionen dieser Schnittstelle bewirken Aktionen des GS, des DBS bzw. beider Systeme.

Abschließend seien noch einmal die wichtigsten Vorteile der hier skizzierten Integrationsarchitektur vorgestellt:

- Einheitliche Schnittstelle zum AWS mit anwendungsbezogenen Objekten.
- Zentrale Verwaltung graphischer und nicht-graphischer Daten.
- Konsistenz zwischen graphischer und nicht-graphischer Objektrepräsentation.

Zusammenfassung

Die wesentlichen Eigenschaften eines CAD-Systems bestanden bislang hauptsächlich in der hohen Funktionalität, einer leichten Handhabbarkeit und einem hohen Maß an Benutzerfreundlichkeit. Zunehmend werden weitere Merkmale als wichtig erkannt. Sie erstrecken sich von der Portabilität über die leichte Erweiterbarkeit um benutzerspezifische Funktionen bis hin zu einer möglichen Kopplung mit anderen CAD- bzw. CAM-Systemen. Das wachsende Bedürfnis nach diesen Systemeigenschaften führt zu der Forderung nach einer Strukturierung der CAD-Software durch eine klare Abgrenzung in Anwendungs-, Datenhaltungs- und Graphikaspekte sowie zur Standardisierung der Schnittstellen zwischen den entsprechenden Subsystemen. Auf dem Graphiksektor sind durch die Normierung der GKS-Schnittstelle gewisse Vorgaben zu erkennen. Dies wird durch die wachsende Vermarktung entsprechender Graphiksysteme stark untermauert. Im Bereich der Datenhaltung gibt es bereits seit langem einheitliche Schnittstellennormen in Form von Datenmodell-Definitionen. Dennoch hat sich der Einsatz entsprechender, konventioneller Datenbanksysteme im CAD-Bereich bislang nicht durchgesetzt, da den zu erwartenden Vorteilen ein unerträglich schlechtes Leistungsverhalten gegenübersteht.

Zur Erforschung der allgemeinen CAD-Problematik und der Untersuchung der in diesem Zusammenhang interessierenden Datenbankaspekte dient die Prototypentwicklung des DB-gestützten 3D-Bauteilmodellierers KUNICAD. Neben der Bereitstellung eines den realen Konstruktionsvorgang unterstützenden Anwendermodells gilt der Charakterisierung der Subsystemschnittstellen eine besondere Aufmerksamkeit, da von ihnen ein dominierender Einfluß auf die Leistungsfähigkeit des Gesamtsystems ausgeht.

Durch die quantitative und qualitative Bewertung dieser Subsystemschnittstellen können Schwachstellen erkannt und aufgezeigt werden. Zunächst ergibt sich die Forderung nach einer Erweiterung bestehender Graphik- bzw. Datenbanksysteme um mächtigere Modellierungswerkzeuge zur Beschreibung komplex strukturierter Objekte. Dies führt zur Vereinheitlichung der Objektrepräsentationen im Datenbanksystem auf der einen und dem Graphiksystem auf der anderen Seite. Als Konsequenz hieraus ergibt sich eine Integrationsarchitektur, durch die ein erweitertes Graphik- und ein erweitertes Datenbanksystem über einen gemeinsamen Objektpuffer gekoppelt sind. Hiermit wird eine Vereinheitlichung der Datenhaltung innerhalb eines CAD-Systems ermöglicht. Sowohl graphische als auch nicht-graphische Daten werden durch ein einziges System verwaltet und durch eine übergreifende operationale Schnittstelle der Anwendung zur Verfügung gestellt. Ein solcher Ansatz verspricht neben der Erhöhung der Datenqualität eine Verbesserung der Leistungsfähigkeit des Gesamtsystems.

Wir danken Herrn Prof. Dr. T. Härder für die Anregung, uns mit diesem Thema zu befassen sowie für seine hilfreichen Anmerkungen während der Entstehungsphase dieser Arbeit. Ebenso möchten wir uns bei dem Programmkomitee und der Tagungsorganisation für die investierte Zeit und Mühe bedanken.

5. Literaturverzeichnis

- ARW83 Anderl, R., Rix, J., Wetzel, H.: GKS im Anwendungsbereich CAD, in: Informatik Spektrum, Band 6, Heft 2, Springer-Verlag, April 1983.
- BB84 Batory, D.S., Buchmann, A.P.: Molecular Objects, Abstract Data Types and Data Models: A Framework, in: Proc. of the 10th International Conf. on Very Large Data Bases, Singapore, August 1984, pp. 172-184.
- CMMS84 Christmann, H.-P., Meyer-Wegener, K., Mitschang, B., Sikeler, A.: Datenbankunterstützung für ein CAD-Arbeitsplatzsystem, in: Tagungsband des Fachgesprächs der GI-Fachgruppe 4.1.1 "Graphische Systeme", Bonn-Bad Godesberg, November 1984.
- Da86 Dadam, P., et al.: A DBMS Prototype to Support Extended NF² Relations: An Integrated View on Flat Tables and Hierarchies, in: Proc. ACM-SIGMOD Conf., Washington, D.C., May 1986.
- En83 Encarnacao, J.: Interfaces of CAD-Systems with its Application Environment - Requirements, Conflicts, Evaluation Procedure, in: Proc. of IFIP WG 5.2 Working Conference on CAD Systems Framework, Roeros Norway, June 1982, pp. 23-39.
- En84 Enderle, G.: IGES (Das aktuelle Schlagwort), Informatik-Spektrum, Band 7, Heft 1, Febr. 1984, S. 45.
- GKS83 mehrere Artikel in: Informatik Spektrum, Band 6, heft 2, Springer Verlag, April 1983.
- HHLM86 Härder, T., Hübel, Ch., Langenfeld, S., Mitschang, B.: KUNICAD - Ein datenbankgestütztes geometrisches Modellierungssystem für Werkstücke, erscheint in der Zeitschriftenreihe Forschung und Entwicklung, Springer Verlag, 1986
- HHM85 Härder, T., Hübel, Ch., Mitschang, B.: Decomposition and parallel execution of complex database operations, Forschungsbericht Nr. 25/85, SFB 124, Universität Kaiserslautern, Dezember 1985.
- HM85 Hartzband, D.J., Maryanski, F.J.: Enhancing Knowledge Representation in Engineering Databases, in: IEEE Computer, September 1985, pp. 39-48.
- HR85 Härder, T., Reuter, A.: Architektur von Datenbanksystemen für Non-Standard-Anwendungen, in: Proc. GI-Fachtagung "Datenbanksysteme in Büro, Technik und Wissenschaft", März 1985, IFB 94, Springer-Verlag Berlin, Seite 253-286 (eingeladener Vortrag).
- M184 Mitschang, B.: Überlegungen zur Architektur von DB-Systemen für Ingenieur Anwendungen, in: Tagungsband der 14. GI-Jahrestagung in Braunschweig, Informatik-Fachberichte 88, Springer Verlag Berlin, 1984, S. 318-334.
- M185 Mitschang, B.: Ein Datenmodell zur Abbildung von komplexen Objekten, Forschungsbericht des SFB 124, Nr. 20/85, Universität Kaiserslautern, 1985.
- PL77 TEKTRONICS: Plot/10 Terminal Control System, User Manual, TEKTRONIX Inc. Beaverton, Oregon, 1977.
- Sch85 Schek, H.-J.: Towards a Basic Relational NF² Algebra Processor, in: Proc. 2nd Int. Conf. on FODO, Kyoto, 1985.
- UDS84 UDS, verschiedene Handbücher zum Datenbanksystem der Firma Siemens, München, 1984.

Diese Arbeit entstand im Rahmen eines Projektes innerhalb des von der Deutschen Forschungsgemeinschaft geförderten Sonderforschungsbereiches 124.