

Universität Stuttgart  
Fakultät Informatik



---

# **ASN und die relationalen Datenbanksysteme (Realisierungsaspekte)**

*Dr.-Ing. Marian Mešina*

E-mail: Prof Roller <roller@informatik.uni-stuttgart.de>

CR-Klassifikation: C.2.4; J6; H.2.4; H.5.3

Institut für Informatik  
Fakultät Informatik  
Universität Stuttgart  
Breitwiesenstraße 20-22  
D-70565 Stuttgart

## **ASN und die relationalen Datenbanksysteme (Realisierungsaspekte)**

*Dr.-Ing. Marian Mešina*

Bericht Nr. 2002/03  
Juli 2002

## Zusammenfassung

In diesem Report wird eine mögliche Weiterentwicklung des ASN-Datenmodells präsentiert. Das ASN wurde zuerst mit dem objektorientierten Datenbanksystem "ObjectStore" realisiert. Das Ziel dieses Berichtes ist es, die Möglichkeiten der Realisierung von ASN mit relationalen Datenbanken zu untersuchen. Es wurde ein Datenmodell vorgeschlagen, der die Integration von neuen Objektarten, von neuen Eigenschaften der Objekte und von neuen Beziehungen zwischen den Objekten ohne Änderung der Datenstrukturen in der Datenbank erlaubt. Die Lösung von wichtigsten Aufgaben, wie z.B. die Beschreibung von Constraints, die Benutzung von hierarchischen Datenstrukturen, die Beschreibung der Projekte, die Beschreibung von Beziehungen in relationalen Datenbanken wurde beschrieben. Es wurde gezeigt, wie die Effizienz von ASN durch die Verfeinerung des ASN-Datenmodells verbessert werden kann. Für die erste Implementierung von ASN mit relationalen Datenbanken wurde als Testentwicklungsumgebung MS Access vorgesehen.

# Inhaltsverzeichnis

1.	Zielsetzung: .....	4
2	Die wichtigsten Komponenten des ASN-Datenmodells .....	5
2.1	Das Kern des ASN-Datenmodells .....	5
2.2	Objektarten.....	6
2.3	Die Handhabung der Objektmerkmale.....	7
2.4	Vorläufige Festlegung der Datenfelder in der Tabelle Merkmale. ....	9
2.5	Beispiele für objektspezifische Tabellen.....	12
2.6	Beziehungen zwischen Objekten, die Tabelle Netz.....	12
2.7	Beziehungsarten.....	13
2.8	Vorschläge für Namenskonventionen und die Versionsverwaltung. ....	15
3	Die Handhabung von Hierarchien im ASN Modell .....	16
4	Die Beschreibung der Prozessabläufe .....	20
5	Die Beschreibung von Constraints im ASN-Datenmodell.....	25
5.1	Zusammenfassung:.....	33
6	ECA-Regeln und das ASN-Datenmodell .....	34
7	Effizienzsteigerung des ASN durch die Verfeinerung des ASN-Datmodells .....	43
8	ASN-Datenmodellierung mit MS Access .....	47
8.1	Datenschnittstellen und Programmschnittstellen des Systems MS Access .....	50
8.2	Heterogene Netze und das ASN.....	52
8.3	Zusammenfassung.....	53
9	Zeitplan für die 1. Testphase (Vorschlag) .....	54

## 1. Zielsetzung:

Für die Realisierung des SFB 374 Projektes benötigen wir ein gemeinsames Datenmodell. Eine der wichtigsten Aufgaben der ASN-Datenbank besteht darin, die Zugriffe auf alle, für RPD notwendigen Objekte zu kontrollieren, zu koordinieren und alle nicht kompatiblen Änderungen der Objekte zu vermeiden. Ohne eine gemeinsame ASN-Datenbank kann die wichtigste ASN-Unterstützung für RPD nicht vorgeführt werden.

Das ASN-Datenmodell muss Datenbankunabhängig sein und die Funktionalität, die im SFB 374 Forschungsprojekt vorgesehen wurde, voll enthalten. In der Konstruktion des vorgelegten Datenmodells wurden nur solchen Elemente vorgesehen, dass das Modell in objektorientierten, aber auch in relationalen Datenbanksystemen implementiert werden kann.

Es wird angestrebt, die ASN-Datenbank als Überbau, als Zusatz zu dem bereits vorhandenen EDV-Umfeld der SFB Arbeitsgruppen aufzubauen.

Die ASN-Datenbank enthält und verwaltet hauptsächlich nur Referenzen auf die Objekte der realen Welt und die Objekte der EDV-Welt (Betriessystemfiles in lokalen Netzwerk und im Internet).

Des weiteren werden diese Referenzen auch als Objekte bezeichnet.

Die ASN-Benutzer arbeiten weiter mit ihren fachspezifischen Anwendungen, bearbeiten und verwalten ihre Dateien. Die ASN-Datenbank enthält Informationen über den Speicherort der Dateien, über den Bearbeitungsstand, über die Zugriffsrechte und die erlaubten Zugriffsmethoden. Die Änderungen auf einem Objekt führen automatisch zur Statusänderung der abhängigen Objekte und zur Benachrichtigung der zuständigen Mitarbeitern.

Die Benutzung der relationalen Datenbanksysteme für ASN ist möglich. Die objektbezogene Information muss meistens über mehrere Tabellen verteilt werden. Allerdings ist die Anzahl der notwendigen Tabellen relativ klein. Die objektbezogenen Daten können aus den Tabellen relativ leicht und schnell zusammengestellt werden. Die Beziehungen zwischen den Objekten liegen explizit als Tabelle vor und können leicht ausgewertet werden. Für die Erfassung von neuen Objekten, neuen Objekteigenschaften und neuen Beziehungen zwischen Objekten ist die Änderung der Datenbankstruktur nicht notwendig. Die feste Datenstruktur der Datenbank erleichtert und beschleunigt den Zugriff auf die Information.

Das gemeinsame ASN-Datenmodell muss relativ schnell aufgebaut und in einer Testdatenbank implementiert werden. Diese Datenbank muss für alle im SFB 374 Projekt beteiligten Personen zugänglich sein.

Die Abbildung der Realität in einem Datenmodell ist nicht eindeutig. Hier handelt es sich nur um mehrere Vorschläge, um eine Diskussionsgrundlage. In Absprache mit allen Beteiligten muss das Datenmodell endgültig festgelegt werden. Erst danach werden in der ASN-Datenbank die Testdaten gespeichert. Letztlich müssen Voraussetzungen geschaffen werden, damit alle Beteiligten mit der Testdatenbank im Dialog arbeiten können.

Wenn uns gelingt, auf der Basis der hier vorgelegten Konzeption und in Absprache mit allen Beteiligten ein ASN-Datenmodell aufzubauen, wären die zukünftigen Programmentwicklungen weitgehend von der Auswahl des Datenbanksystems unabhängig. Die externen Programme, die auf die ASN-Datenbank zugreifen, müssten für unterschiedliche Datenbanksysteme nur die entsprechenden Treiber verwenden.

## 2 Die wichtigsten Komponenten des ASN-Datenmodells

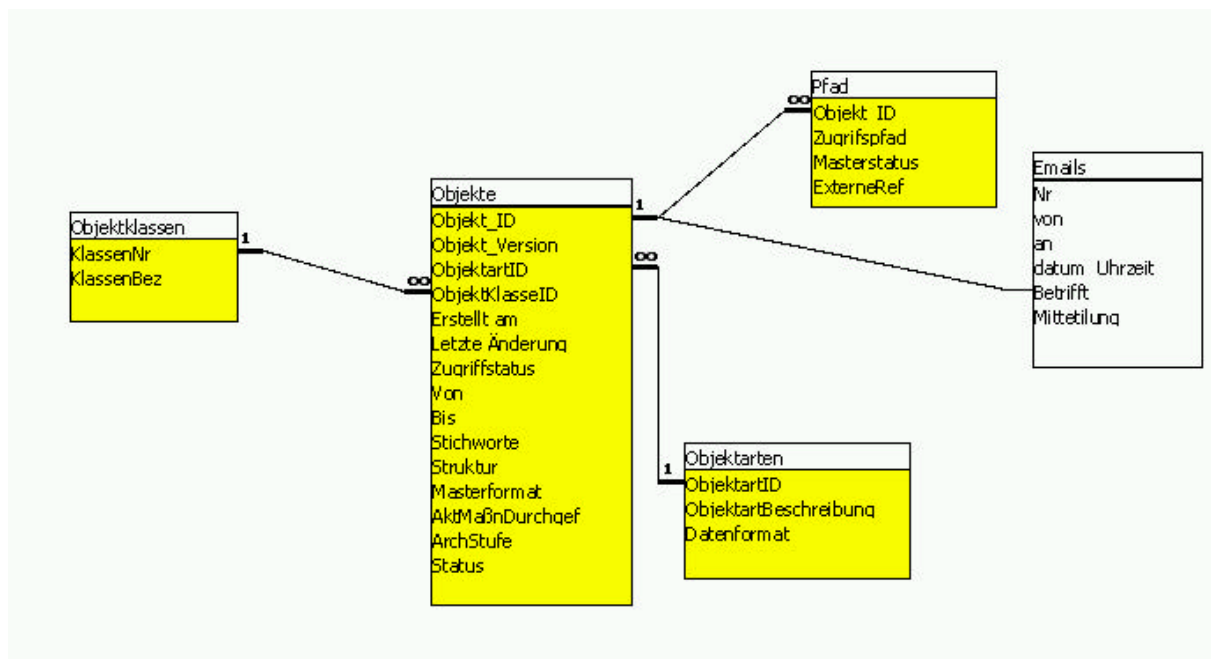


Abbildung 1 Das Kern des ASN-Datenmodells

### 2.1 Das Kern des ASN-Datenmodells

Die Tabellen "Objekte" und "Pfad" bilden das Kernstück des ASN-Datenmodells. Um die Objekte unter verschiedenen Gesichtspunkten unterscheiden zu können, gehören zu dem ASN-Kern auch die Tabellen "Objektklassen" und "Objektarten"

Die Tabelle "Objekte" enthält Referenzen auf die Objekte der realen Welt, aber auch auf die Objekte der ASN-Datenbank. Es ist zunächst offen, welche Objekte der RPD-Prozesskette direkt und welche nur als Referenzen in der ASN-Datenbank erfasst werden sollen. Es bietet sich an, z.B. die CAD-Modelle nur als Referenzen zu verwalten. Es ist aber auch denkbar, einen Teil der Information über die Merkmale der realen Objekte in der ASN-Datenbank und ein Teil außerhalb der ASN-Datenbank, als Betriebssystemdateien oder als Einträge in anderen Datenbanken zu speichern und verwalten.

Feldname	Datentyp	Beschreibung
Objekt_ID	Text	Schlüssel der Tabelle, eindeutige Objektbezeichnung
Objekt_Version	Text	Zahl oder Text, Versionsnummer (Optional)
ObjektartID	Zahl	s. Tabelle "Objektarten"
ObjektKlasseID	Zahl	s. Tabelle Objektklassen
Erstellt am	Datum	wichtig als Zugriffskriterium
Letzte Änderung	Datum	wichtig als Zugriffskriterium
Zugriffstatus	Text	frei, im Lesezugriff, im Schreibzugriff, wird geändert, im Lesezugriff und Schreibzugriff, wurde geändert
Von	Datum	Existenzdauer (z.B. für Projekte)
Bis	Datum	Existenzdauer
Stichworte	Text	die wichtigsten Stichworte als Grundlage für Suchfunktion
Struktur	Boolean	ja/nein < - > zusammengesetzt/einfach

Feldname	Datentyp	Beschreibung
Masterformat	Text	bevorzugter Speicherungsformat
AktMaßnDurchgef	Boolean	wichtig für ECA-Regeln, betrifft die abhängigen Objekte
ArchStufe	Text	wichtig für die Suchfunktion/Zugriff und die Archivierung
Status	Text	"normal", "soll überprüft/geändert werden"

**Tabelle 1** Die Struktur der Tabelle "Objekte"

Der Speicherungsart der referierten Objekte wird in der Tabelle "Pfad" erfasst. Von einem Objekt können im Netzwerk mehrere Kopien existieren, aber nur eine Kopie darf den Masterstatus haben. Nur der Inhalt einer solchen Kopie gilt als "richtig" und nur eine solche Kopie wird permanent gepflegt und auf dem neuesten Stand gehalten.

Feldname	Datentyp	Groesse
Objekt_ID	Text	255
Zugriffspfad	Text	Pfad im Netzwerk, ftp-Adresse, Internetadresse
Masterstatus	Boolean	ja-master status, nein – nur einfache Kopie
ExterneRef	Hyperlink	Direktverknüpfung zum Objekt im Netzwerk oder im Internet und auch zu der Anwendung, mit der das Objekt bearbeitet wird

**Tabelle 2** Die Struktur der Tabelle "Pfad"

Der Doppelklick auf den Hyperlinkeintrag in der Tabelle "Pfad" öffnet die referenzierte Datei mit dem Programm, das zu der Dateiart unter Windows zugewiesen wurde. So z.B. wird eine DXF-Datei mit dem Programm AutoCAD geöffnet.

## 2.2 Objektarten

ObjektartID	ObjektartBeschreibung	Datenformat
1	Anforderungsdatei/Vorgabendatei	
2	Beschreibungsdatei (mehrere Objekte)	
3	CAD 2D-Modell	
4	CAD 3D-Modell	
5	CAD-Geschichtedatei	
6	Kompatibilitätsdatei	
7	Constraintsdatei	
8	Kostenberechnungsdatei	
9	Model_Parameterdatei	
10	NC-Datei	
11	Sonstige	
12	OptKomponentenliste	
13	Person	
14	Projekt	
15	Projektbeschreibung	
16	Rasterbild	
17	Resource	
18	Simulationsdatei	

ObjektartID	ObjektartBeschreibung	Datenformat
19	Systemneutrale CAD-Modelldatei	
20	Team, Gruppe	
21	Objektbeschreibungsdatei	
22	Eingabedatei für die Berechnungen	
23	Ausgabedatei für die Berechnungen	
24	Fertigungsdatei	
25	Prüfungsbericht	
26	Beschreibungsdatei, bezogen auf 1 Objekt	
27	Forderungen	
28	Fertigungsverfahren	
29	Prototyp	
30	Firma/Unternehmen/Werkstadt	
31	Unternehmensbereich	
32	Aufgabe	
33	Virtueller Prototyp	
34	Aktion (als Komponente in der Aufgabe)	
35	Plan (bezieht sich auf eine Aktion, Aufgabe, Projekt)	
36	Prozess	

**Tabelle 3** Auszug aus der Tabelle "ObjektArten"

Wir müssen vorläufig ca. 36 Objektarten unterscheiden. Die Unterscheidung ist für die Handhabung der ECA-Regeln sehr wichtig.

### 2.3 Die Handhabung der Objektmerkmale

In der Tabelle "Objekte" wurden nur solche Merkmale erfasst, die man für alle vorgesehenen Objektarten verwenden kann.

Um die restlichen Merkmale zu erfassen, sind im Prinzip zwei Wege denkbar.

A.) Objekt spezifische Merkmaltabelle einzuführen, wie z.B. Tabelle "Personal":

Vorname	Nachname	Strasse	PLZ	Ort
Johann	Keppler	Milchstrasse 7	77777	Sonnenstadt
...	...	...		

**Tabelle 4**

Nachteile dieser Vorgehensweise:

1. Wir hätten im Datenmodell sofort mindestens 35 neuen Tabellen. Wenn wir nur die Vielfalt der CAD-Objekte bedenken, sogar noch deutlich mehr
2. Bei Einführung einer neuen Objektart müsste jeweils eine neue Tabelle angelegt werden
3. Bei Einführung einer neuen Eigenschaft müsste die Struktur der Tabelle geändert werden

4. Die gleiche Eigenschaft/Merkmal befindet sich in verschiedenen Objekttabellen in verschiedenen Spalten und deswegen wäre die Beschreibung von Constraints sehr schwierig.
- B.) Einführung von nur einer Tabelle "Merkmale" für alle Objekte mit folgender Grundstruktur:

Objekt_ID	Merkmal_ID bzw. Merkmalbezeichnung	Wert	...weitere Merkmaleigenschaften...
Produkt_A24_V1	Farbe	rot	....
MitarbPersNr_40	Geburtstag	31.12.1954	...

**Tabelle 5**

Vorteile dieser Vorgehensweise:

1. Für neue Objektarten müssen nicht unbedingt neue objektspezifische Tabellen angelegt werden.
2. Bei Einführung einer neuen Eigenschaft muss die Struktur der vorhandenen Tabellen nicht geändert werden. Also können neue Merkmale ohne Änderungen der Tabellen erfasst werden.
3. Die Handhabung von Constraints wird sehr erleichtert.

In der Praxis es ist empfehlenswert, beide Merkmalbeschreibungsmethoden zu kombinieren. Merkmale, die mit Sicherheit in den Constraints benutzt werden, sollten unbedingt mit der zweiten Methode erfasst werden.

Die Benutzung der vorgeschlagenen Tabelle "Merkmale" bringt u. a. folgende Vorteile:

1. Für ein vorgesehenes Merkmal können die Ist-Werte und auch die Soll-Werte mit Toleranzgrenzen erfasst werden.
2. Die Handhabung des ungenauen und unsicheren Wissens ist möglich.
3. Die Merkmalliste ist offen. Es können jederzeit neuen Merkmale erfasst werden.
4. Die zeitlich begrenzte Gültigkeit der Merkmale ist vorgesehen



## 2.4 Vorläufige Festlegung der Datenfelder in der Tabelle Merkmale.

Auch die Merkmale der Objekte haben ihre Eigenschaften. Die Eigenschaften sollen für alle Merkmale und Merkmalarten möglichst einheitlich festgelegt werden. Die vorläufige Festlegung ist in der folgenden Tabelle dargestellt.

Objet_ID	MerkmalID	Merkmalart	Wert	Obere Grenze	Untere Grenze	Einheit	Datentyp	Wahrscheinlichkeit?	Gültigkeit von?	Gültigkeit bis?	Vorgabe Wert	Vorgabe UGr	Vorgabe OGr	Status
DemoTeil1	Biegefestigkeit	1	300	0	0	N		1						
DemoTeil1	Kosten	1	45	0	0	EUR		1						
NC_Maschine_01	Zustand_bereit	1	ja	0	0			1						
Objekt1	Breite	1	24	0	0			1						
Objekt1	Hoehe	1	2	0	0			1						
Objekt1	Laenge	1	3	0	0			1						
Objekt1	Volumen	1	144	0	0			1						
Objekt1	Oberflaeche	1	252	0	0			1						
Objekt1	Eigengewicht	1	2	0	0			1						
Objekt1	Gesamtgewicht (mit Bauelementen)	1	4	0	0			1						
Objekt2	Breite	1	5	0	0			1						
Objekt2	Hoehe	1	6	0	0			1						
Objekt2	Laenge	1	7	0	0			1						
Objekt2	Volumen	1	210	0	0			1						
Objekt2	Oberflaeche	1	214	0	0			1						
Objekt2	Eigengewicht	1	5	0	0			1						
Objekt2	Gesamtgewicht (mit Bauelementen)	1	5	0	0			1						
Objekt3	Eigengewicht	1	7	0	0			1						
Objekt3	Gesamtgewicht (mit Bauelementen)	1	7	0	0			1						

**Tabelle 6** Auszug aus der Tabelle "Merkmale"

## Kurzerklärung der Datenfelder in der Tabelle Merkmale

Objet_ID	Kennzeichnung des Objekts, zu dem das Merkmal gehört
MerkmalID	Merkmalbezeichnung oder MerkmalID (Nummer)
Merkmalart	um den Zugriff auf bestimmten Merkmalarten zu erreichen
Wert	Ist-Wert des Merkmals
ObereGr	Betr. unsicheres Wissen, wenn der Wert z.B. von Fertigungsverfahren abhängig ist
UntereGr	Betr. unsicheres Wissen, wenn der Wert z.B. von Fertigungsverfahren abhängig ist
Einheit	Betr. quantitative Merkmale
Datentyp	Information eventuell für Programmentwicklungen verwendbar
Wahrscheinlichkeit?	Betr. unsicheres Wissen
Gültigkeit von?	Um Zeitfaktoren bei dem Produkteinsatz zu berücksichtigen
Gültigkeit bis?	Um Zeitfaktoren bei dem Produkteinsatz zu berücksichtigen
Vorgabe	Soll Wert des Merkmals, Vorgabe
VorgabeOGr	Soll-Wert des Merkmals, obere Toleranzgrenze
VorgabeUGr	Soll-Wert des Merkmals, untere Toleranzgrenze
Status	"Normal" oder "wurde geändert" (Bedeutung für die ECA -Regeln)

Tabelle 7 Datenfelder in der Tabelle "Merkmale"

MerkmalID	Merkmal
-2	Objekt Zugriffsstatus
-1	Objekt Status
0	Nothing
1	ProduktforderungsID
2	Bezeichnung1
3	Wichtigkeit
4	Beschreibung
5	Biegefestigkeit
6	Bruchfestigkeit
7	E-Modul
8	Hitzebestaendigkeit
60	Wie hoch ist der angestrebte Zielpreis?
61	Wie hoch sind die Materialeinzelkosten?
62	Wie hoch sind die Einzelkosten der Fertigung?
63	Wieviele Tage benötigt die Entwicklung des Prototyps?
64	Wie hoch ist der Stundenssatz der Entwicklung?
65	Das Design ist aufwendig (=1) oder standard (=0)?
66	Wieviele verschiedene Teile werden verwendet?
67	Wieviele dieser Teile sind neu entwickelt?
68	Wieviele der verwendeten Teile sind Wiederhohlteile?
69	Wieviele Varianten werden produziert?
70	Fertigungsgemeinkosten (Personal)
71	Fertigungskomplexitätsfaktor (Design)
72	Fertigungskomplexitätskosten
73	Herstellkosten
74	Herstellkomplexitätsfaktor (Anzahl Teile)
75	Herstellkomplexitätskosten (Anzahl Teile)
76	Herstellkomplexitätsfaktor (Teile Neuentwicklung)

<b>MerkmalID</b>	<b>Merkmal</b>
77	Herstellkomplexitätskosten (Teile Neuentw.)
78	Herstellkomplexitätsfaktor (Teile Wiederholung)
79	Herstellkomplexitätskosten (Teile Wiederh.)
80	Verwaltungskomplexitätsfaktor (Varianten)
81	Verwaltungskomplexitätskosten
82	Drifting Costs
103	Rauhigkeit der Oberfläche
104	Materialeinzelkosten
105	Fertigungseinzelkosten
106	Prototyphersteller
107	Maschine
108	Prototypwerkstadt
109	Qualität
110	Sollwert
111	GefObereToleranz
112	GefUntereToleranz
113	Zeit
114	Kosten
115	Aktionstermin
116	Standort
117	Anschaffungskosten
134	Zustand_bereit
135	Prototypprüfungen erfolgreich

**Tabelle 8** Auszug aus der Tabelle Merkmalenliste

Es wäre denkbar, in der Tabelle "Merkmale" direkt mit den Merkmalbezeichnungen zu arbeiten. Um Speicherplatz zu sparen wäre es jedoch günstiger, anstelle der Merkmalbezeichnungen nur die Merkmal\_ID Nummern zu verwenden.

Die Benutzung dieser Konzeption wird nochmals im Kapitel "Constraints" angesprochen.

## 2.5 Beispiele für objektspezifische Tabellen

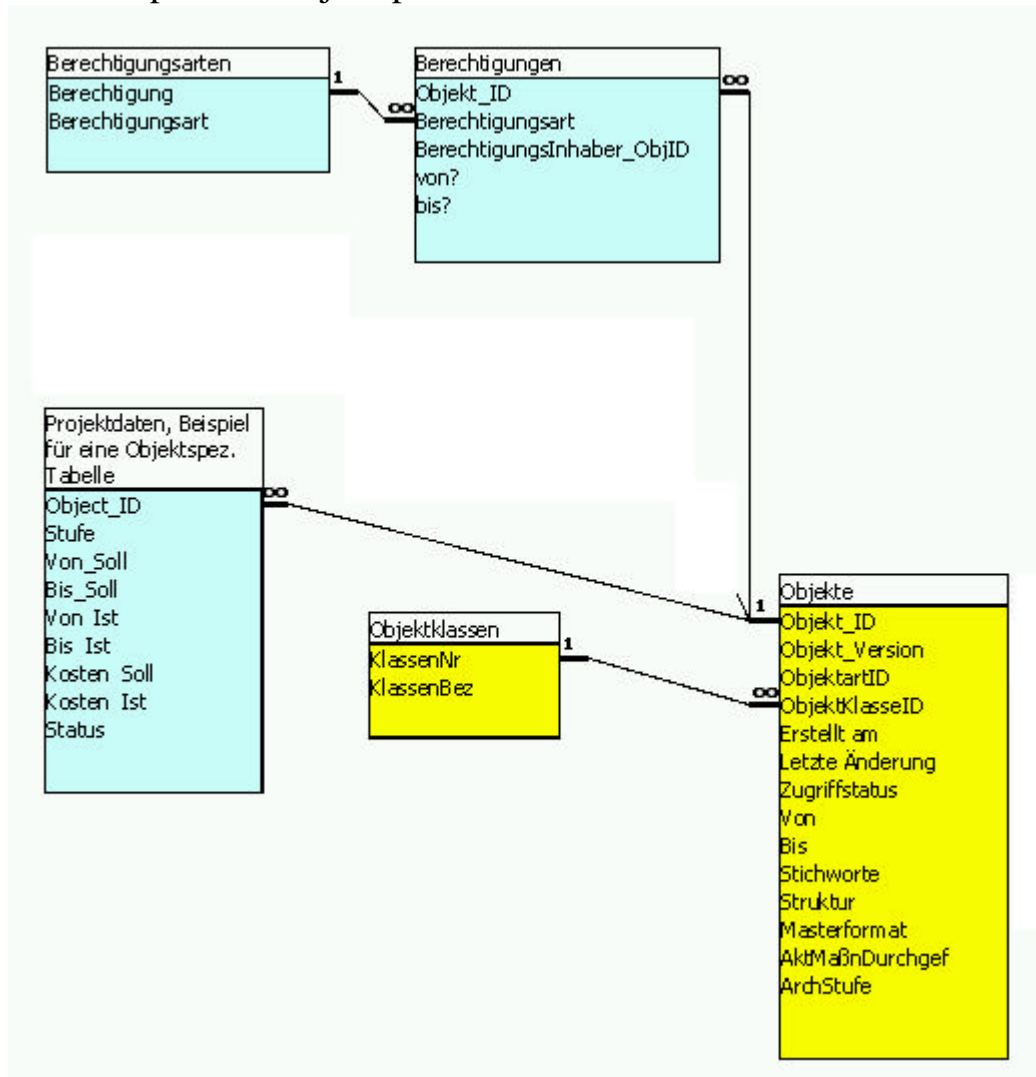


Abbildung 2 Beispiel für einige objektspezifischen Tabellen

Als Beispiel für objektspezifischen Tabellen kommen u.a. Berechtigungstabellen, Projektdaten und Prozessbeschreibungstabellen in Frage.

Alle Datenbanken haben ihre eigene Sicherheitsphilosophie. Meisten werden den Benutzern die Zugriffsrechte auf Objekte der Datenbank fest zugewiesen. Die Tabelle "Berechtigungen" ist Grundlage dafür, dass die Zugriffsrechte durch die Änderung der Einträge in dieser Tabelle sehr flexibel geändert werden können. Parallel zu der dem Datenbanksystem eigenen Sicherheitskonzeption kann anhand der Tabellen "Berechtigungen" und "Berechtigungsarten" eine eigene Sicherheitskonzeption aufgebaut werden, die auch beim Zugriff der externen Programme auf die Datenbank genutzt werden kann.

## 2.6 Beziehungen zwischen Objekten, die Tabelle Netz

Zwischen Objekten der realen Welt existieren verschiedene Beziehungen. Diese Beziehungen können geometrisch als Netz dargestellt werden. Die Knotenpunkte des Netzes sind Objekte. Wenn die Änderungen der Knotenpunkte automatisch zu semantisch notwendigen Änderungen der anderen Objekte führen, spricht man von aktiven semantischen Netzwerken.

In objektorientierten Datenbanksystemen ist die Existenz der Beziehungen auf der Konzeptebene vorgesehen und auf der Instanzebene durch die Beziehungsslots realisiert. Das Einführen und Löschen von Beziehungen bedeutet Änderungen in der Netzstruktur.

In hier vorgelegten Datenmodell werden die Beziehungen zwischen den Objekten durch die Tabelle "Netz" beschrieben. Die Tabelle "Netz" hat eine sehr einfache Struktur. Das Löschen oder Einfügen von Beziehungen bedeutet nur das Löschen oder Einfügen der Datensätze in der Tabelle "Netz". Die Information über Beziehungen liegt explizit als Tabelle vor. Dadurch kann sie schnell abgefragt und verwaltet werden.

Feldname	Datentyp	Beschreibung
Vorgänger	Text	Objekt_ID, z.B. PersonalNr
Nachfolger	Text	Objekt_ID, z.B. ZeichnungName
Beziehung	Zahl	Zahl, z. B. 37 – entspricht ist zuständig
BeziehungsDetails	Text	z. B. zuständig als Prüfer (Bemerkung: auch anders realisierbar)
von	Datum	Zeitgrenze für die Gültigkeit der Beziehung
bis	Datum	Zeitgrenze für die Gültigkeit der Beziehung

Tabelle 9 Die Struktur der Tabelle "Netz"

In der Tabelle Netz können nur sogenannten Dualbeziehungen erfasst werden. Die Constraints gelten als Beziehungen ganz anderer Art und müssen deswegen anders gehandhabt werden.

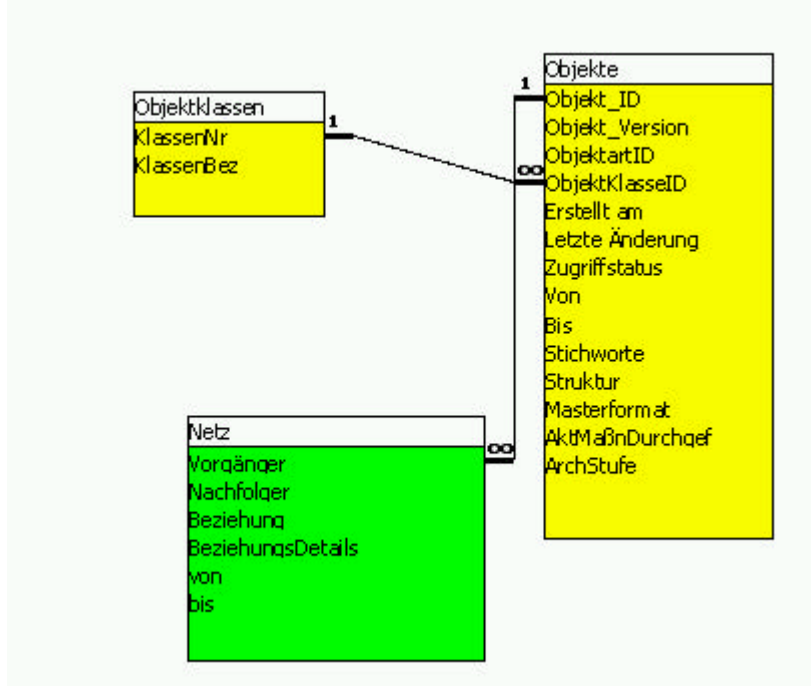


Abbildung 3 Beziehungen zwischen den Tabellen "Objekte" und "Netz"

Die bisherige Analyse der RPD-Problematik zeigt, dass ca. 38 Beziehungsarten zu unterscheiden sind.

## 2.7 Beziehungsarten

BeziehungID	Beschreibung
0	keine Beziehung
1	A enthält B als Kopie (Objekt-Objekt), (Zeichnung, Zeichnung)
2	A enthält B als Referenz (Objekt-Objekt)

BeziehungID	Beschreibung
3	A enthält Teile von B als Kopie (Objekt-Objekt)
4	A enthält Teile von B als Referenz (Objekt-Objekt)
5	B ist abgeleitet von A (Objekt-Objekt)
6	A und B enthalten Teile voneinander (Objekt-Objekt)
7	A und B sind ähnlich (Objekt-Objekt)
8	A kann den B enthalten (Objekt-Objekt)
9	A kann den B enthalten als ??? (wenn B in A mehrmals vorkommen kann)
10	A kooperiert mit B (Personen, Teams), (Person,Person)
11	A ist untergeordnet B (Personen, Teams), (Person,Person)
12	A ist übergeordnet B (Personen, Teams) ???
13	A ist Mitglied in B (Personen, Teams)
14	A ist equivalent mit B (z.B. CAD-Modelle, untersch. Formate)
15	A hat als Leiter1 B (Person-Person)
16	A hat als Leiter2 B (Person-Person)
17	A enthält B (Physikalisch, Auto-Motor)
18	A Betrifft B (Dokumente, Produkte)
19	A enthält den B als ? (wenn B in A mehrmals vorkommen kann)
20	Objekt A passt zu Objekt B
21	A und B haben ein gemeinsames Merkmal (Ähnlichkeit)
22	A und B haben ein gemeinsames Merkmal und Merkmalwert (Ähnlichkeit)
23	A braucht B (Personen-Ressourcen) Unterscheidung??
24	A braucht B (Projekte-Ressourcen) Unterscheidung??
25	A braucht B (Teams-Ressourcen) Unterscheidung??
26	A braucht B (Objekte-Ressourcen) Unterscheidung??
27	A bearbeitet B (Person-Objekt)
28	A ist verantwortlich für B (Person-Objekt,Resource)
29	A ist entstanden durch Synchronisierung von B
30	A erfüllt Forderungen B
31	A gehört zu B (Produkt, Projekt)
32	A besitzt B (Firma, Ressourcen)
33	A ist Mitarbeiter von B (Person, Firma, Unternehmensbereich)
34	A herstellt B (Firma, Produkt)
35	A ist beteiligt am B (Firma, Projekt)
36	A ist Stellvertreter von B (Person, Person)
37	A ist zuständig für B
38	B ist Version von A

**Tabelle 10** Die Auflistung der Beziehungsarten

Die Handhabung von symmetrischen Beziehungen muss noch per Vereinbarung festgelegt werden.

Je feiner ist die Unterscheidung der Beziehungen, um so genauer ist die Abbildung des RPD-Wissens in der ASN-Datenbank.

Unter Berücksichtigung der internen Struktur der Objekte könnte ein viel feineres semantisches Beziehungsnetz aufgebaut werden. In der Tabelle 10 wurde die interne Struktur der Objekte nicht berücksichtigt.

**Die Benutzung der Tabelle "Netz" könnte auch in objektorientierten Datenbanken als Alternative zu der Benutzung von Beziehungsslots verwendet werden.**

## 2.8 Vorschläge für Namenskonventionen und die Versionsverwaltung.

Wenn ein Objekt (z.B. ein CAD-Modell) in verschiedenen Datenformaten gespeichert wurde, entstehen in der Datenbank verschiedene Objekte.

Der Name der Objekte soll (aber muss nicht unbedingt) den Hinweis auf die Gemeinsamkeit enthalten (z. B. BaugruppeX25\_IGES, BaugruppeX25\_DXF).

Die Gemeinsamkeit der Objekte muss unbedingt in der Tabelle "Netz" abgebildet werden. Dort steht ein Datensatz, der die Information enthält, dass die Objekte BaugruppeX25\_IGES und BaugruppeX25\_DXF äquivalent sind (s. Beziehung Nr. 14 in der Tabelle 10)

Die Versionen eines Objektes werden in der Datenbank als selbständige Objekte, mit unterschiedlichen Objekt\_IDs gehandhabt. Der Name sollte, muss aber nicht unbedingt den Hinweis auf das Ursprungsobjekt Objekt enthalten, wie z. B. BaugruppeX25\_V1, BaugruppeX25\_V2 usw.

Viel wichtiger sind die Einträge in der Tabelle "Netz". Dort steht die Beziehungen Nr. 38 (B ist Version von A). Dadurch lassen sich ganze Versionsbäume und Versionsnetze leicht erfassen und im Bedarfsfall leicht rekonstruieren.

### 3 Die Handhabung von Hierarchien im ASN Modell

Die Objekte der Datenbank werden häufig nach verschiedenen Gesichtspunkten in verschiedenen Hierarchien angeordnet. Im Prinzip wäre es denkbar, auch die Zugehörigkeit zu Hierarchien in der Tabelle "Netz" zu erfassen. Für die Beschleunigung der hierarchiebezogenen Zugriffe zu den Objekten wird hier noch eine alternative Vorgehensweise vorgeschlagen.

Nehmen wir als Beispiel zwei Hierarchien, "Organisationseinheiten" und "Produktkategorien".

Ein Kurzwagen KWTHK gehört in die Produktkategorie Kompaktführungen SR, Typ SR-V (s. Pfeil, Abbildung 5) und wurde bei der Firma A, in der Niederlassung Berlin, im Fachbereich Entwicklung, in der Abteilung Konstruktion, Arbeitsgruppe B entwickelt (s. Pfeil, Abbildung 4). Somit muss der Kurzwagen KWTHK in 2 logisch unabhängigen Hierarchien eingeordnet werden.

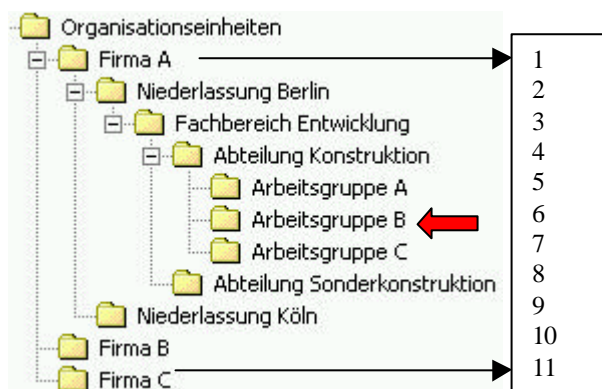


Abbildung 4 Nummerzuweisung zum Pfad (Position) in der Hierarchie

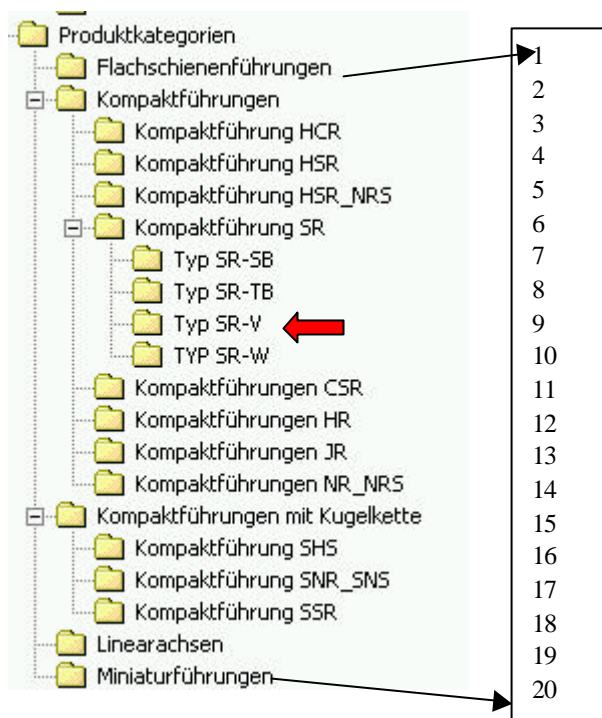


Abbildung 5 Nummerzuweisung zum Pfad (Position) in der Hierarchie

In den folgenden Tabellen wird gezeigt, wie man die Hierarchien und die Zugehörigkeit des Objektes in beiden Hierarchien beschreiben kann.



Jeder Pfad in der Hierarchie bekommt eine Nummer (s. Abbildungen 4 und 5), die in einer Hierarchie nur einmal benutzt werden darf. Die Auswahl der Nummer ist ohne Bedeutung. Auch Lücken in der Nummerierung sind erlaubt.

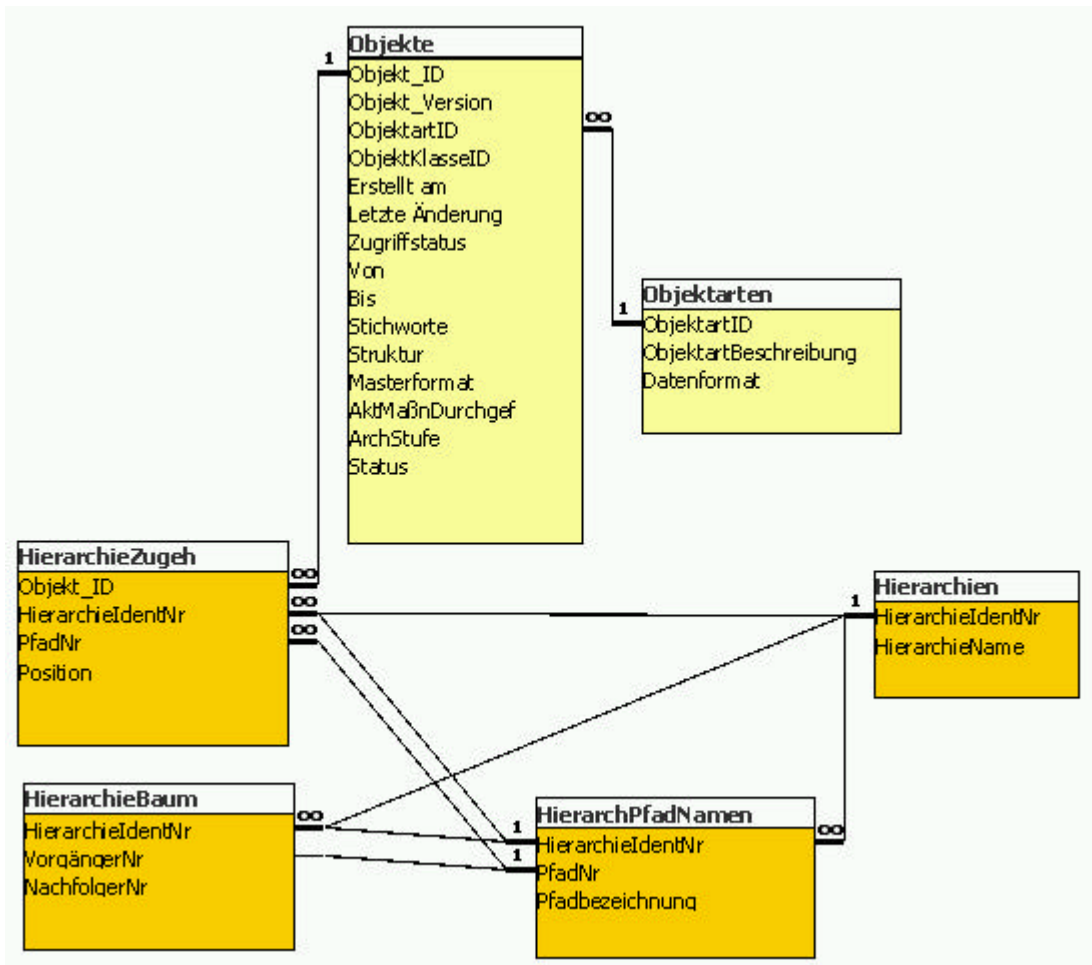


Abbildung 6 Die Beschreibung der Hierarchien im ASN-Modell

HierarchieIdentNr	HierarchieName
1	Firma xyz
2	Klassen
3	Produktkategorien
4	Organisationseinheiten

Tabelle 11 Beispiel für die Inhalte der Tabelle Hierarchien

In der Tabelle "Hierarchien" werden Hierarchien ganz verschiedener Art eingetragen.

Es ist denkbar, auf die Tabelle "Hierarchien" zu verzichten und auch die Hierarchien als Objekte in der Tabelle "Objekte" zu führen. Dann wäre es notwendig, das Datenfeld "HierarchieIdentNr" als Datentyp Text zu deklarieren und dort nicht nur die Zahlen, sondern auch die Texte zur eindeutigen Identifizierung von Hierarchien zu verwenden.

Darüber muss noch eine Vereinbarung getroffen werden.

In der Tabelle "HierarchPfadNamen" wird die Zuweisung der Nummern zu den Pfaden in den einzelnen Hierarchien erfasst.

HierarchidentNr	PfadNr	Pfadbezeichnung
3	1	Flachschienenführungen
3	2	Kompaktführungen
3	3	Kompaktführungen HCR
3	4	Kompaktführungen HSR
3	5	Kompaktführungen HSR_NRS
3	6	Kompaktführung SR
3	7	Typ SR-SB
3	8	Typ SR-TB
3	9	Typ SR-V
3	10	Typ SR-W
3	11	Kompaktführungen CSR
3	12	Kompaktführungen HR
3	13	Kompaktführungen JR
3	14	Kompaktführungen NR_NRS
3	15	Kompaktführungen mit Kugelkette
3	16	Kompaktführungen SHS
3	17	Kompaktführungen SNR_SNS
3	18	Kompaktführungen SSR
3	19	Linearachsen
3	20	Miniaturführungen
4	1	Firma A
4	2	Niederlassung Berlin
4	3	Fachbereich Entwicklung
4	4	Abteilung Konstruktion
4	5	Arbeitsgruppe A
4	6	Arbeitsgruppe B
4	7	Arbeitsgruppe C
4	8	Abteilung Sonderkonstruktion
4	9	Niederlassung Köln
4	10	Firma B
4	11	Firma C

**Tabelle 12** Beispiel für die Inhalte der Tabelle "HierarchPfadNamen"

Die Struktur der Hierarchien wird in der Tabelle "HierarchieBaum" beschrieben "

HierarchidentNr	VorgängerNr	NachfolgerNr
3	1	2
3	2	3
3	3	4
3	4	5
3	4	6
3	4	7
3	3	8
3	1	9
3	10	0
3	11	0
4	1	0
4	2	3
4	2	4

HierarchidentNr	VorgängerNr	NachfolgerNr
4	2	5
4	2	6
4	6	7
4	6	8
4	6	9
4	6	10
4	11	0
4	12	0
4	13	0
4	14	0
4	15	16
4	15	17
4	15	18
4	19	0
4	20	0

**Tabelle 13** Beispiel, Auszug aus der Tabelle "HierarchieBaum"

**Festlegung:** Pfade, die keinen Nachfolger haben, müssen auch in der Tabelle "HierarchieBaum" eingetragen werden. Dort steht im Datenfeld "NachfolgerNr" Wert 0. So kann man die "Blattobjekte" in der Hierarchie sehr leicht erkennen.

Objekt_ID	HierarchidentNr	PfadNr	Position
Wagen KWTHK	3	9	0
Wagen KWTHK	4	6	0

**Tabelle 14** Auszug aus der Tabelle "HierarchieZueh"

Die Zuordnung der Objekte zu verschiedenen Hierarchien wird in der Tabelle "HierarchieZueh" eingetragen.

Ein Objekt kann in der gleichen Hierarchie mehrmals eingetragen werden. Ein Mitarbeiter ist zum Beispiel in der Abteilung Sonderkonstruktion als Leiter und in der Arbeitsgruppe C als Berater eingetragen. Diese Information kann im Datenfeld "Position" gespeichert werden.

Die hier vorgelegte Konzeption der Handhabung der Hierarchien hat folgende Vorteile:

1. Objekte können leicht in die vorhandenen Hierarchienstrukturen eingeordnet oder entfernt werden. (Datensätze in der Tabelle "HierarchieZueh" einfügen und löschen)
2. Beliebige neue Hierarchien können aufgebaut werden (Neue Datensätze in der Tabellen "Hierarchien", "HierarchieBaum" und "HierarchPfadNamen").
3. Die vorhandenen Hierarchien können leicht modifiziert werden.

In den Tabelle "HierarchieZueh" dürfen nur solche Wertkombination ("HierarchieIdentNr"- "PfadNr") benutzt werden, die vorher in der Tabelle "HierarchPfadNamen" als Kombination ("HierarchieIdentNr"- "PfadNr") eingetragen wurden.

In den Tabelle "HierarchieBaum" dürfen nur solche Wertkombinationen. ("HierarchieIdentNr"- "VorgängerNr") und ("HierarchieIdentNr"- "NachfolgerNr") benutzt werden, die vorher in der Tabelle "HierarchPfadNamen" als Kombinationen ("HierarchieIdentNr" – "PfadNr") eingetragen wurden.

Es können automatische Prüfmechanismen aufgebaut werden, die die Erfüllung dieser Forderungen überprüfen.

## 4 Die Beschreibung der Prozessabläufe

### Testbeispiel: Prototyperstellung

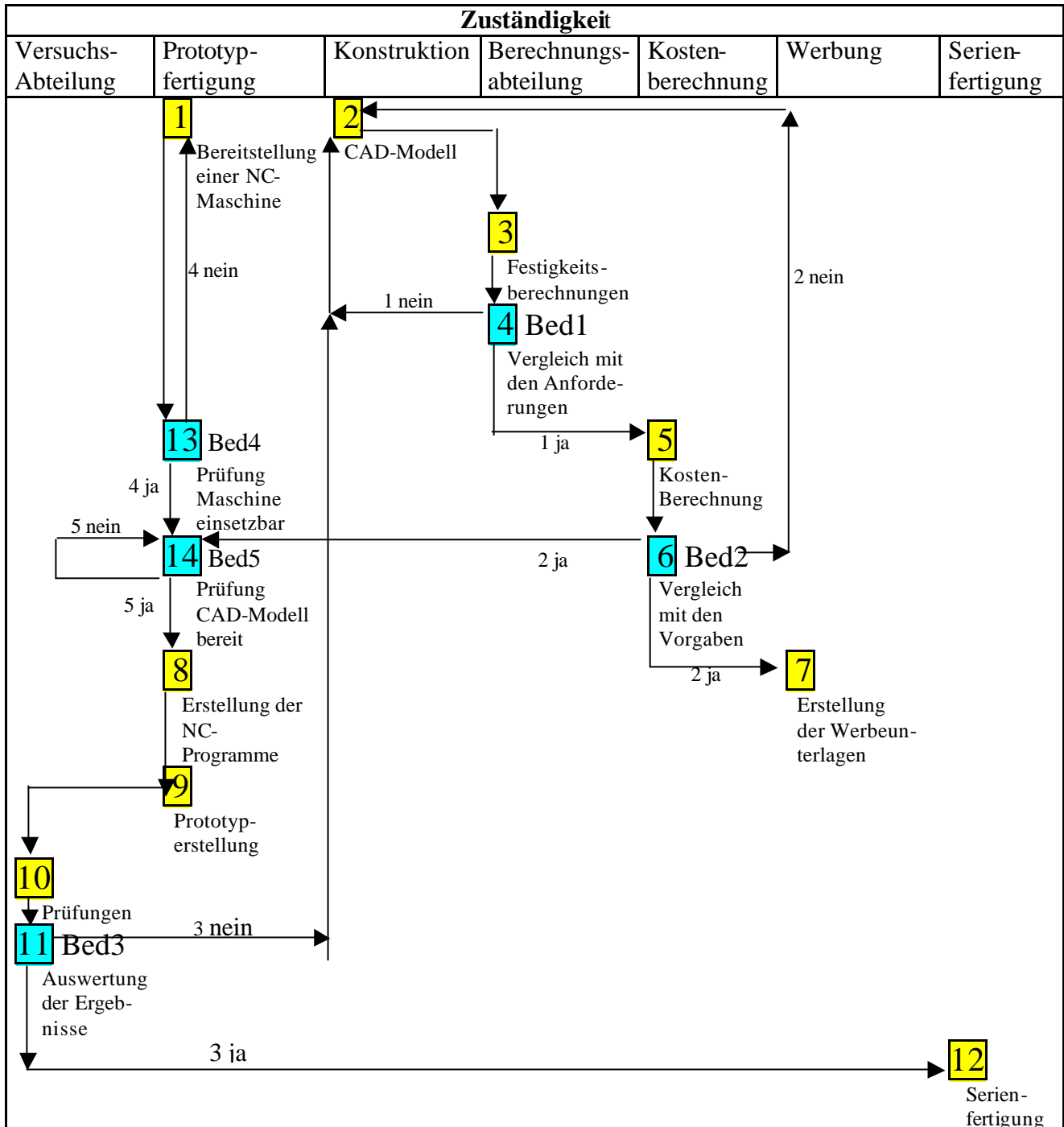


Abbildung 7 Darstellung des Prozessverlaufs als orientierter Graph.

Um einen Prototyp eines Produktes zu erstellen, muss eine NC-Maschine bereitgestellt werden. Das CAD-Modell wird in der Abteilung Konstruktion entwickelt und dann an die Rechenabteilung weitergegeben. Dort werden FEM Festigkeitsüberprüfungen des mit CAD modellierten Objektes durchgeführt. Wenn die Überprüfung negativ ausfällt, muss das CAD-Modell geändert werden. Sonst wird die Kostenberechnung durchgeführt. Wenn die Kosten im vorgesehenen Rahmen liegen, können Werbeunterlagen und die NC-Programme hergestellt werden. Danach folgt die Prototyp-erstellung. Der Prototyp wird in der Versuchsabteilung verschiedenen Prüfungen unterworfen. Wenn

die Prüfungen negativ ausfallen würden, muss das CAD-Modell geändert werden. Sonst wird die Serienfertigung gestartet. Weiterhin gehen wir davon aus, dass die Werbeunterlagen dynamisch mit CAD-Modell verknüpft sind. Somit ist die Änderung der Werbeunterlagen nach der Änderung des CAD-Modells nicht mehr notwendig.

Im orientierten Graph (s. Abbildung 7) können wir Knotenpunkte finden, die nur eine Tätigkeit/Aktivität beschreiben (1,2,3,5,7,8,9,10,12) und Verzweigungspunkte die Prüfbedingungen enthalten (4,6,13,11,13,14). In unserem Beispiel (s. Abbildung 7) haben wir zwei Eingangspunkte (1,2) und 2 Ausgangspunkte (7,12). Die Prozessbeschreibung führt zu einer netzartiger Struktur.

Zwei Eingangspunkte zu haben, bedeutet, dass 2 Aktivitäten parallel und zunächst unabhängig voneinander anfangen. Der Start der beiden Aktivitäten muss nicht gleichzeitig sein. Das Prozess endet mit zwei Aktivitäten, die ebenfalls nicht synchron sein müssen.

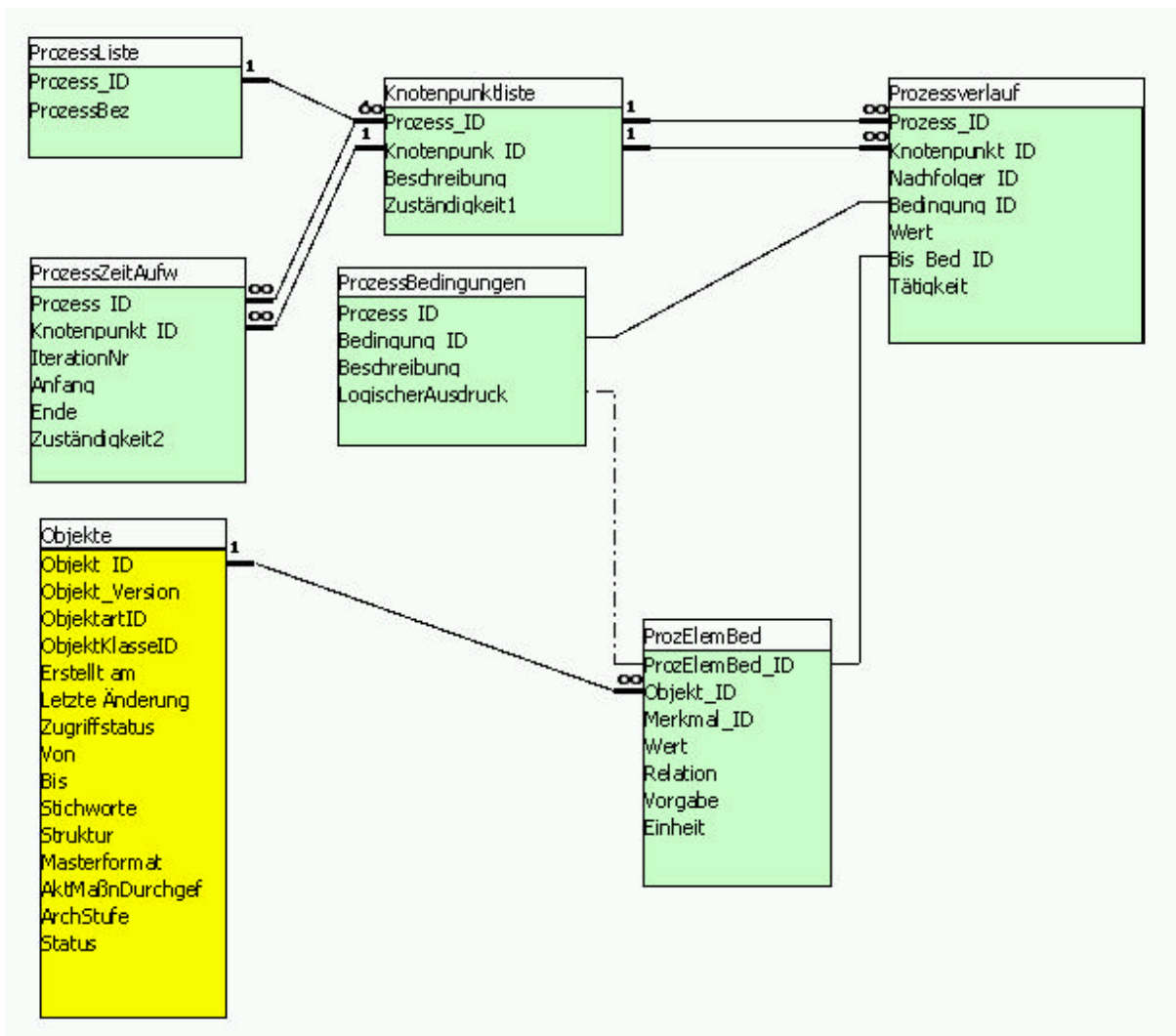


Abbildung 8 Beschreibung der Prozesse im ASN-Datenmodell.

Für die Prozessbeschreibung brauchen wir im ASN-Modell mehrere Tabellen. Alle Prozesse werden in der Tabelle "ProzessListe" eingetragen.

Es wäre denkbar, anstelle der "ProzessListe" auch die Prozesse als Objekte in der Tabelle "Objekte" zu führen. Diese Entscheidung muss in Absprache mit den Arbeitsgruppen getroffen werden.

Prozess_ID	ProzessBez
1	PrototypBau

**Tabelle 15** Beispiel, Auszug aus der Tabelle ProcessListe

In der Tabelle "ProcessListe" wurde zuerst nur 1 Prozess erfasst.

Die Tabelle "Knotenpunktliste" enthält alle Knotenpunkte von allen Prozessen, mit Beschreibung der Tätigkeit.

Es ist vorgesehen, dass im Prozessverlauf einige Knotenpunkte mehrmals erreicht werden. Um die Bearbeitungszeiten erfassen zu können, muss noch die Tabelle "ProzessZeitAufw" angelegt werden. Diese Zusatztable hat vorläufig folgende Struktur:

Prozess_ID	Knotenpunkt_ID	IterationNr	Anfang	Ende	Zuständigkeit2
1	1	1	01.07.2002	02.07.2002	
1	2	1	02.07.2002	04.07.2002	
1	3	1	04.07.2002	05.07.2002	
1	2	2	05.07.2002	06.07.2002	
1	3	2	06.07.2002	07.07.2002	

**Tabelle 16** Auszug aus der Tabelle "ProzessZeitAufw"

In den Datenfeldern "Anfang" und "Ende" kann nicht nur das Datum, sondern auch die Uhrzeit eingetragen werden.

Prozess_ID	Knotenpunkt_ID	Beschreibung	Zeitaufwand	Zuständigkeit1
1	1	Bereitstellung einer NC-Maschine		
1	2	Erstellung/Anpassung des CAD-Modells		
1	3	Festigkeitsberechnung		
1	4	Vergleich mit Festigkeitsanforderungen (Bed1)		
1	5	Kostenberechnung		
1	6	Vergleich mit Kostenvorgaben (Bed2)		
1	7	Erstellung der Werbeunterlagen		
1	8	Erstellung der NC-Programme		
1	9	Prototyperstellung		
1	10	Prototypprüfungen in der Versuchsabteilung		
1	11	Auswertung der Ergebnisse (Bed3)		
1	12	Freigabe für Serienfertigung		
1	13	Prüfung der NC-Maschine (Bed4)		
1	14	Prüfung CAD-Modell vorhanden (Bed5)		

**Tabelle 17** Auszug aus der Tabelle "Knotenpunktliste"

Da im Prozessablauf verschiedene Kriterien abgefragt werden und die Prozesse eventuell auch anders als geplant ablaufen können, kann die Tabelle "ProzessZeitAufw" erst schrittweise, nachdem in den einzelnen Knotenpunkten die Bearbeitung beendet wurde mit Daten gefüllt werden. Somit stellt sie eine nachträgliche Prozessdokumentation dar.

Die Tabellen mit Datenstruktur wie "ProzessZeitAufw" könnte man auch als Planungstabellen verwendet werden.

Die Tabelle "ProzessZeitAufw" kann als Grundlage für die grafische Darstellung des Zeitverlaufs der Prozesse benutzt werden. Für die Lösung dieser Aufgabe könnte man z. B. MS Excel verwenden.

Wenn die Zuständigkeit für die Knotenpunkte bei einzelnen Iterationen des Prozessablaufs bei verschiedenen Personen bzw. Organisationseinheiten liegt, kann sie in der Tabelle "ProzessZeitAufw" im Datenfeld "Zuständigkeit2" eingetragen werden.

Wenn die Zuständigkeit für die Knotenpunkte beim ganzen Prozessablauf bei den gleichen Personen bzw. Organisationseinheiten bleibt, kann sie in der Tabelle "Knotenpunkliste" im Datenfeld "Zuständigkeit1" eingetragen werden.

Für die Beschreibung von Entscheidungskriterien werden die Tabellen "Prozessbedingungen" und "ProzElemBed" benutzt.

ProzElemBed_ID	Objekt_ID	Merkmal_ID	Wert	Relation	Vorgabe	Einheit
1	DemoTeil1	5		>=	45	N
2	DemoTeil1	114		<=	90	EUR
3	DemoTeil1	135		=	ja	
4	NC_Maschine_01	134		=	Ja	
5	DemoTeil1_CAD	-1		=	normal	
6			Iter(1,7)	<	1	

**Tabelle 18** Auszug aus der Tabelle "ProzElemBed"

In der Tabelle 18 bezeichnen die Merkmal\_ID Nummer folgende Merkmale: 114 die Kosten, 5 die Biegefestigkeit, 134 den Zustand "Maschine ist bereit", 135 – Prototypprüfung bestanden, -1 – Objektstatus "normal" (die Objektbearbeitung und Folgemaßnahmen wurden abgeschlossen).

Die Zuweisung der Zahlenwerte zu den Merkmalen wird in der Tabelle "Merkmallenliste" beschrieben.

Die Spalte "Wert" wird nicht immer benutzt.

In der Tabelle "ProzElemBed" werden einfache Bedingungen erfasst, die sich direkt auf die in der Tabelle "Merkmale" gespeicherten Merkmalwerte beziehen.

Es gibt auch Bedingungen, die sich nicht direkt auf die Tabelle "Merkmale" beziehen. Dann würden in der Tabelle "ProzElemBed" die Felder "Objekt\_ID" und "Merkmal\_ID" leer bleiben. Im Datenfeld "Wert" kann z. B. die aktuelle Zeit eingetragen und mit der Vorgabe verglichen werden. In der Tabelle 18 bedeutet die Funktion Iter(1,7) die Anzahl der bereits erreichten Iterationen (Bearbeitungen) des Knotenpunktes 7 im Prozess 1. Die Werbeunterlagen werden nur dann erstellt, wenn sie noch nicht existieren. (s. Annahme – die Änderung der Werbungsunterlagen nach der Änderung des CAD-Modells ist nicht notwendig)

Die Kriterien für die Prozessverzweigungspunkte werden aus der Tabelle "Prozessbedingungen" übernommen. Die Zeitabfragen können auch in den Prozessbedingungen eingebaut werden.

Prozess_ID	Bedingung_ID	Beschreibung	LogischerAusdruck
1	1	Bed1	=B1
1	2	Bed2	=B2
1	3	Bed3	=B3
1	4	Bed4	=B4
1	5	Bed5	=B5
1	6	Bed6	=B2 & B6

**Tabelle 19** Auszug aus der Tabelle "Prozessbedingungen"

In der Spalte "LogischerAusdruck" wurden (mit Ausnahme der letzten Zeile) einfach die Prozelem-  
Bed\_ID Nummer aus der Tabelle "ProzElemBed" übernommen und nur um den Buchstabe "B"  
ergänzt. Allerdings ist vorgesehen, in dieser Spalte komplexe Ausdrücke (UND, ODER, XOR,  
NICHT usw.) aus den einfachen Kriterien aus der Tabelle "ProzElemBed" zu bilden.

Prozess_ID	Knotenpunkt_ID	Nachfolger_ID	Bedingung_ID	Wert	Bis_Bed_ID	Tätigkeit
1	1	13		0		0
1	2	3		0		0
1	3	4		0		0
1	4	2		1 nein		0
1	4	5		1 ja		0
1	5	6		0		0
1	6	2		2 nein		0
1	6	14		2 ja		0
1	6	7		6 ja		0
1	8	9		0		0
1	9	10		0		0
1	10	11		0		0
1	11	12		3 ja		0
1	11	2		3 nein		0
1	13	14		4 ja		0
1	13	1		4 nein		0
1	14	14		5 nein		0
1	14	8		5 ja		0

**Tabelle 20** Auszug aus der Tabelle "Prozessverlauf"

In der Tabelle 20 haben wir nur die Werte für das Prozess Nr. 1 übernommen. Die Segmente des orientierten Graphs sind durch die Knotenpunktnummer eines Knotens und die Knotenpunktnummer seines Nachfolgers beschrieben.

Ein Knotenpunkt kann mehrere Nachfolger haben.

Wenn ein Übergang immer, ohne Überprüfung einer Bedingung stattfindet, befindet sich in der Spalte "Bedingung\_ID" der Wert 0.

Es gibt 2 Möglichkeiten:

In einem Verzweigungsknoten darf immer nur eine Bedingung aus der Tabelle "Prozessbedingungen" benutzt werden. Je nachdem, welche Werte die Bedingung liefert, werden die Nachfolgerknotenpunkte gewählt. Also können in der Bedingung mehrere Werte (Zahlen oder Texte) abgefragt werden.

Als Alternative wäre es möglich, bei jedem aus dem Knotenpunkt ausgehenden Pfad eine separate Bedingung zu verwenden, die nur mit "ja" oder "nein" beantwortet werden muss. Die Tabelle "Prozessverlauf" wurde so konzipiert, dass man mit beiden Optionen arbeiten kann..

Die Datenspalte "Bis\_Bed\_ID" wird vorläufig nicht benutzt. Die Spalte "Tätigkeit" kann Zusatzinformationen enthalten, die sich auf die Übergänge zwischen den Knotenpunkten beziehen.

Die Nummerierung der Knotenpunkte hat für die Reihenfolge der Bearbeitung keine Bedeutung



## 5 Die Beschreibung von Constraints im ASN-Datenmodell

### Aufgabe:

Die Datenstrukturen für die Constraintbeschreibung wurden so angelegt, dass man verschiedene quantitative Merkmale von verschiedenen Objekten durch mehrere Formeln koppeln kann. Dabei soll es möglich sein, frei zu wählen, welche Merkmale neu berechnet werden sollen und welche als Vorgabegrößen vorgesehen wurden. Des weiteren wird in der Datenstruktur gespeichert, ob die gewünschten Constraint-Berechnungen nach der Änderung der Merkmalwerte bereits durchgeführt wurden oder noch nicht.

Die Berechnungen werden mit einem externen Programm durchgeführt (z. B. MS Excel/Solver, Mapple usw.) und die Ergebnisse in die ASN-Datenbank automatisch eingetragen.

Im folgenden Beispiel wurden 6 Objekten insgesamt 7 Constraints zugewiesen. Die Constraints 1,2,3,4,6,7 legen Beziehungen zwischen den Merkmalen einzelner Objekte fest. Nur das Constraint Nr. 5 bringt die Merkmale von mehreren Objekte zueinander in Beziehung. Das Gesamtgewicht des Objektes 5 ist gleich dem Eigengewicht plus Gewicht der benutzten Bauteile von Objekt1 und Objekt3

Objekt_ID	Merkmal	Constraint_ID						
		1	2	3	4	5	6	7
Objekt1	Breite	a						
Objekt1	Hoehe	b						
Objekt1	Laenge	c						
Objekt1	Volumen	v						
Objekt2	Breite		a					
Objekt2	Gesamtgewicht (mit Bauelementen)					G1		
Objekt2	Hoehe		b					
Objekt2	Laenge		c					
Objekt2	Oberflaeche		F					
Objekt3	Gesamtgewicht (mit Bauelementen)					G2		
Objekt3	Hoehe			h				
Objekt3	Kante_A			r				
Objekt3	Volumen			v				
Objekt4	Hoehe				h			
Objekt4	Kante_A				r			
Objekt4	Oberflaeche				F			
Objekt5	Eigengewicht					G0		
Objekt5	Gesamtgewicht (mit Bauelementen)					G		
Objekt6	Hoehe						h	h
Objekt6	Kante_A						r	r
Objekt6	Oberflaeche							F
Objekt6	Volumen						v	

**Tabelle 21** Beispiel für die Festlegung von Constraints (keine ASN-Tabelle, nur View)

Diese Information muss im ASN-Modell in mehreren Tabellen abgebildet werden. In der Tabelle 21 ist noch nicht erkennbar, welche Formeln für die Constraints benutzt wurden.

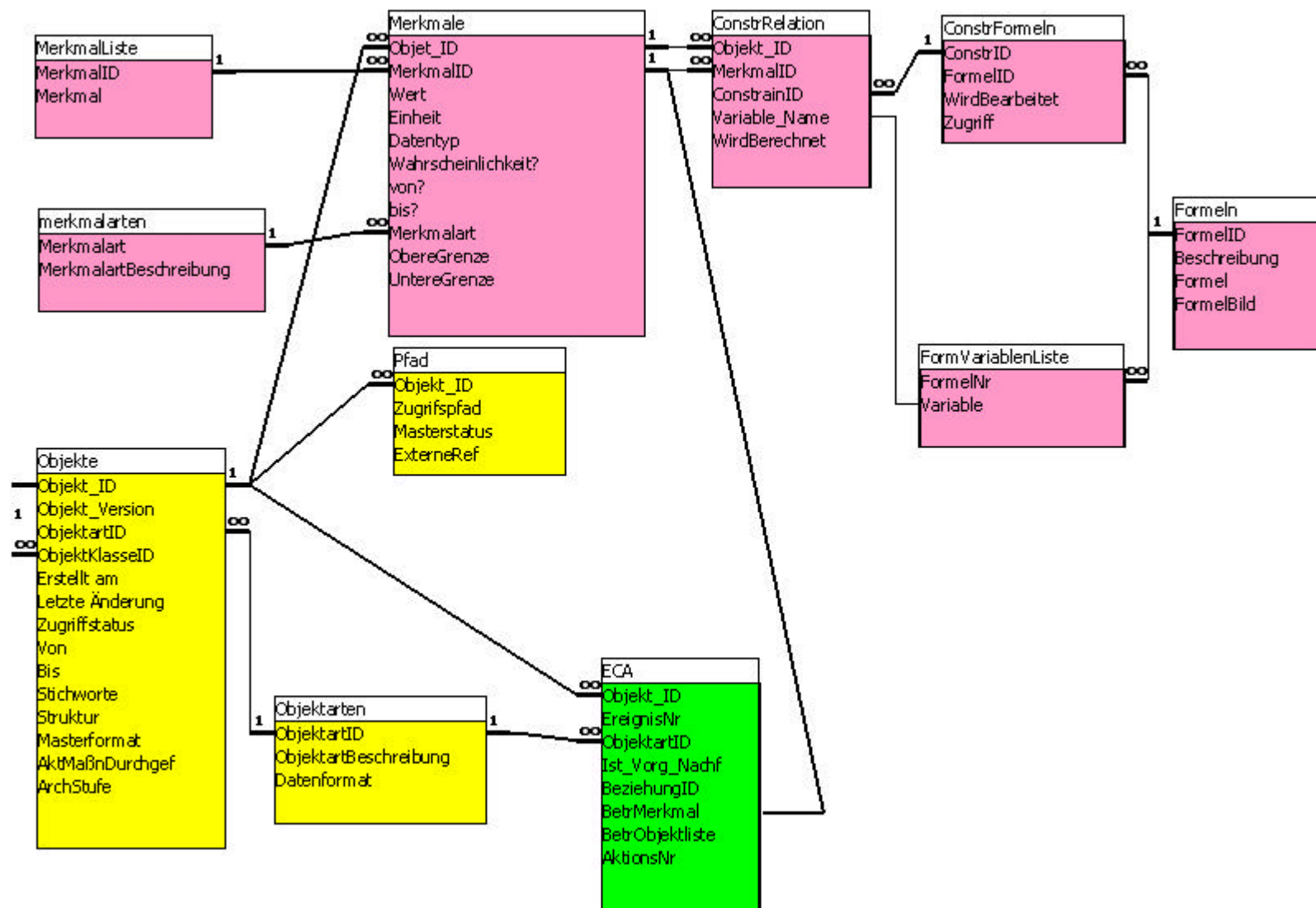


Abbildung 9 Tabellen für die Handhabung der Merkmale und Constraints

Objekt_ID	Merkmal	1	2	3	4	5	6	7
Objekt1	Breite	$V=a*b*c$						
Objekt1	Hoehe	$V=a*b*c$						
Objekt1	Laenge	$V=a*b*c$						
Objekt1	Volumen	$V=a*b*c$						
Objekt2	Breite		$F=2*(a*b+b*c+c*a)$					
Objekt2	Gesamtgewicht (mit Bauelementen)					$G=G0+G1+G2$		
Objekt2	Hoehe		$F=2*(a*b+b*c+c*a)$					
Objekt2	Laenge		$F=2*(a*b+b*c+c*a)$					
Objekt2	Oberflaeche		$F=2*(a*b+b*c+c*a)$					
Objekt3	Gesamtgewicht (mit Bauelementen)					$G=G0+G1+G2$		
Objekt3	Hoehe			$V=PI*R*R*H$				
Objekt3	Kante_A			$V=PI*R*R*H$				
Objekt3	Volumen			$V=PI*R*R*H$				
Objekt4	Hoehe				$F=2*PI*R*(R+H)$			
Objekt4	Kante_A				$F=2*PI*R*(R+H)$			
Objekt4	Oberflaeche				$F=2*PI*R*(R+H)$			
Objekt5	Eigengewicht					$G=G0+G1+G2$		
Objekt5	Gesamtgewicht (mit Bauelementen)					$G=G0+G1+G2$		
Objekt6	Hoehe						$V=PI*R*R*H$	$F=2*PI*R*(R+H)$
Objekt6	Kante_A						$V=PI*R*R*H$	$F=2*PI*R*(R+H)$
Objekt6	Oberflaeche							$F=2*PI*R*(R+H)$
Objekt6	Volumen						$V=PI*R*R*H$	

Tabelle 22 Beispiel für die Festlegung von Constraints (keine ASN-Tabelle, nur View)

Die Tabelle 22 zeigt eine andere Sicht von benutzten Constraints als die Tabelle 21. Intern, im ASN Datenmodell werden die Constraints ganz anders gespeichert, als in den Tabellen 21 und 22 dargestellt. Allerdings lassen sich solche Ansichten der Constraints, wie in den Tabellen 21 und 22 dargestellt, vom ASN-Datenmodell ganz leicht ableiten.

Für die Beschreibung von Constraints brauchen wir zuerst die Tabelle "Formeln":

Tabellenname:	Formeln	Beschreibung
<b>Feldname</b>	<b>Datentyp</b>	
FormelID	Zahl	Identnummer der Formel
Beschreibung	Text	Kurzbeschreibung/Bezeichnung der Formel
Formel	Memo	Formelausdruck, entsprechend den Konventionen für VBA
FormelBild	OLE	Grafische Darstellung der Formel

**Tabelle 23 - Die Struktur der Tabelle "Formeln"**

FormelID	Beschreibung	Formel	FormelBild
1	VolumenQuader	$V=a*b*c$	
2	GesamtgewichtGruppe	SUM(Objekt;relation)	
3	FlaecheQuader	$F=2*(a*b+b*c+c*a)$	
4	Algorithmus 1	Sub Test(P1;P2;P3)	
5	Algorithmus 2	$F(x1;x2;x3;...)=0$	
6	Gesamtgewicht2	$G=G0+G1+G2$	
7	VolZyl	$V=PI*R*R*H$	
8	FlaecheZyl	$F=2*PI*R*(R+H)$	
21	Gleich	$a=b$	
22	Ohmsche Gesetz	$u=r*i$	
23	Int1	$e=int(f(x);a;b)$	
24			

**Tabelle 24 Beispiel für die Inhalte der Tabelle "Formeln"**

Jeder Formel entspricht ein Datensatz in der Tabelle "Formeln". Jede Formel wird nur einmal beschrieben und kann dann beliebig oft in der ASN-Datenbank benutzt werden. Für die Bearbeitung von Constraints ist die Spalte "Formel" sehr wichtig. Die Spalte enthält einen arithmetischen Ausdruck, den Aufruf einer Funktion oder den Aufruf einer Subroutine. Jede Formel kann in verschiedene Constraints referiert werden.

Für komplexe Formeln sind in der Tabelle "Formeln" auch Bilder vorgesehen.

Zu jeder Formel gehört eine Variablenliste. Diese Liste ist in der Tabelle "FormVariablen" gespeichert.

Tabellenname:	FormVariablen	Beschreibung
<b>Feldname</b>	<b>Datentyp</b>	
FormelNr	Zahl	Identnummer der Formel
Variable	Text	Die Bezeichnung der variablen

**Tabelle 25 – Die Struktur der Tabelle "FormVariablen"**

FormelNr	Variable
1	a
1	b
1	c
1	v
3	a
3	b
3	c
3	F
6	G
6	G0
6	G1
6	G2
7	h
7	r
7	v
8	F
8	h
8	r

Tabelle 26 – Auszug aus der Tabelle "FormVariablen"

Bei der Constraints-Beschreibung müssen die Variablen einer Formel den Merkmalen der Objekte zugewiesen werden. Die Tabelle "FormVariablen" dient nur zur automatischen Übernahme der Variablenbezeichnungen in die Tabelle "ConstrRelation"

Tabellenname:	ConstrKlassen	Beschreibung
Feldname	Datentyp	
ConstrKlassID	Text	Bezeichnung/IdentNr der Constraintsklasse
Objekt_ID	Text	Identnummer des Objekts
Merkmal_ID	Text	Identnummer des Merkmals
Formel_ID	Zahl	Identnummer der Formel
Variable	Text	Bezeichnung der Variablen

Tabelle 27 – Die Struktur der Tabelle "ConstrKlassen"

Die Tabelle "ConstrKlassen" erlaubt es, die einzelnen Constraints oder eine Ansammlung von Constraints nur einmal zu beschreiben und dann beliebig oft zu verwenden. In dieser Tabelle werden nicht direkt die Merkmalbezeichnungen, sondern deren Identnummer eingetragen (s. Tabelle "MerkmalListe")

ConstrKlassID	Objekt_ID	Merkmal_ID	Formel_ID	Variable
FlaecheQuad	Dummy1	23	3	c
FlaecheQuad	Dummy1	38	3	F
FlaecheQuad	Dummy1	21	3	a
FlaecheQuad	Dummy1	22	3	v
FlaecheZyl	Dummy1	87	8	r
FlaecheZyl	Dummy1	22	8	h
FlaecheZyl	Dummy1	38	8	F

ConstrKlassID	Objekt_ID	Merkmal_ID	Formel_ID	Variable
Gesamtgewicht2	Dummy1	86	6	G1
Gesamtgewicht2	Dummy2	86	6	G2
Gesamtgewicht2	Dummy3	85	6	G0
Gesamtgewicht2	Dummy3	86	6	G
VolQuad	Dummy1	21	1	a
VolQuad	Dummy1	24	1	b
VolQuad	Dummy1	22	1	v
VolQuad	Dummy1	23	1	c
VolZyl	Dummy1	22	7	h
VolZyl	Dummy1	87	7	r
VolZyl	Dummy1	24	7	v

Tabelle 28 – Auszug aus der Tabelle ConstrKlassen

Dummy1,2,3 – Platzhalter für beliebige Objekte

Die Constraintsklassen dienen nur zur Unterstützung der automatischen Erstellung von Constraints

Alle Objektmerkmale, die für die Constraintsbeschreibung notwendig sind, werden in der Tabelle "MerkmalListe" gespeichert.

Tabellenname:	MerkmalListe	Beschreibung
Feldname	Datentyp	
Merkmal	Text	
MerkmalID	Zahl	

Tabelle 29 Die Struktur der Tabelle "MerkmalListe"

Beispiel Auszug aus der Tabelle "MerkmalListe:"

MerkmalID	Merkmal
0	Nothing
1	ProduktforderungsID
2	Bezeichnung1
3	Wichtigkeit
4	Beschreibung
5	Biegefestigkeit
6	Bruchfestigkeit
7	E-Modul
8	Hitzebestaendigkeit
9	Lackierbarkeit
10	TPFForderungIGD
11	Saeurebestaendigkeit
12	Oelbestaendigkeit
13	Kaeltebestaendigkeit
14	Wasserbestendigkeit
...	

Tabelle 30 – Auszug aus der Tabelle "MerkmalListe"

Nur die Merkmale, die in der "MerkmalListe" eingeführt wurden dürfen in der Tabelle "Merkmale" benutzt werden.

Tabellenname:	Merkmale	Beschreibung
Feldname	Datentyp	
Objet_ID	Text	
MerkmalID	Zahl	
Wert	Text	
Einheit	Text	
Datentyp	Text	
Wahrscheinlichkeit?	Double	
von?	Datum	
bis?	Datum	
Merkmalart	Zahl	
ObereGrenze	Double	
UntereGrenze	Double	

**Tabelle 31 – Die Struktur der Tabelle Merkmale**

Die hier vorgelegte Beschreibung der Objektmerkmale zeichnet sich durch eine sehr hohe Flexibilität aus. Es können den Objekten beliebige neue Merkmale zugewiesen werden, ohne die Datenstrukturen der ASN-Datenbank ändern zu müssen. Auch die Handhabung des unsicheren, des ungenauen oder des unvollständigen Wissens ist möglich.

	Objet ID	MerkmalID	Wert	Einheit	Datentyp	Wahrscheinlichkeit?	von?	bis?	Merkmalart	ObereGrenze	UntereGrenze
▶	Objekt1	Breite	24			1				0	0
	Objekt1	Hoehe	2			1			1	0	0
	Objekt1	Laenge	3			1			1	0	0
	Objekt1	Oberflaeche	4			1			1	0	0
	Objekt1	Volumen	4			1			1	0	0
	Objekt2	Breite	5			1			1	0	0
	Objekt2	Gesamtgewicht	33			1			1	0	0
	Objekt2	Hoehe	6			1			1	0	0
	Objekt2	Laenge	7			1			1	0	0
	Objekt2	Oberflaeche	9			1			1	0	0
	Objekt3	Gesamtgewicht	21			1			1	0	0
	Objekt3	Hoehe	8			1			1	0	0
	Objekt3	Oberflaeche	7			1			1	0	0
	Objekt3	Radius	5			1			1	0	0
	Objekt3	Volumen	20			1			1	0	0
	Objekt4	Hoehe	4			1			1	0	0
	Objekt4	Oberflaeche	7			1			1	0	0
	Objekt4	Radius	9			1			1	0	0
	Objekt5	Gesamtgewicht	13			1			1	0	0
	Objekt5	Gewicht ohne Baute	11			1			1	0	0
	Objekt6	Hoehe	5			1			1	0	0
	Objekt6	Oberflaeche	7			1			1	0	0
	Objekt6	Radius	22			1			1	0	0
	Objekt6	Volumen	23			1			1	0	0
*		Nothing				1			1	0	0

**Tabelle 32 Beispiel –Auszug aus der Tabelle "Merkmale"**

In der Tabelle 32 wurden die Werte für Volumen und die Oberfläche noch nicht neu berechnet.

Nur die Merkmale, die in der Tabelle "Merkmale" eingetragen wurden, dürfen für die Beschreibung von Constraints benutzt werden.

Alle Constraints bekommen in der Tabelle "ConstrFormeln" eine eindeutige Nummer, ConstrID.

Tabellenname:	ConstrFormeln	Beschreibung
Feldname	Datentyp	
ConstrID	Zahl	Identnummer des Constraint
ConstrainKlasse	Text	Zugehörigkeit des Constraints zu einer Klasse
FormellID	Zahl	Identnummer der Formel
WirdBearbeitet	Boolean	Ja - das Constraint muß bearbeitet werden, die abhängigen Variablen müssen neu berechnet werden Nein – das Constraint wird bei der Aktualisierung nicht berücksichtigt
Zugriff	Boolean	ja - die Bearbeitung gerade läuft, nein - das Constraint wird momentan nicht bearbeitet

Tabelle 33 Struktur der Tabelle "ConstrFormeln"

In der Tabelle "ConstrFormeln" werden zu den Constraints die Formeln zugewiesen.

ConstrID	ConstrainKlasse	FormellID	WirdBearbeitet	Zugriff
1	VolQuad	1	Ja	Nein
2	FlaecheQuad	3	Nein	ja
3	VolZyl	7	Nein	ja
4	FlaecheZyl	8	Nein	ja
5	Gesamtgewicht2	6	Nein	ja
6	VolZyl	7	Ja	nein
7	FlaecheZyl	8	Ja	nein

Tabelle 34 Beispiel, Auszug aus der Tabelle "ConstrFormeln"

Jedem Constraint wird nur eine Formel zugewiesen. Allerdings können die Merkmale eines Objektes in mehreren Constraints einbezogen werden. Im Datenfeld "WirdBearbeitet" wird eingetragen, ob das Constraint bearbeitet werden soll oder nicht (Bedeutung für die ECA-Regeln)

Tabellenname:	ConstrRelation	Beschreibung
Feldname	Datentyp	
Objekt_ID	Text	Identnummer/Bezeichnung des Objekts
MerkmalID	Zahl	Identnummer des Merkmals
ConstrainID	Zahl	Identnummer des Constraints
Variable_Name	Text	Name der Variablen/Bezeichnung, so wie in den Formeln benutzt wurde
WirdBerechnet	Boolean	entscheidet, ob die Variable vorgegeben ist, oder ob sie berechnet werden muss

Tabelle 35 Die Struktur der Tabelle "ConstrRelation"

Objekt_ID	MerkmalID	ConstrainID	Variable_Name	WirdBerechnet
Objekt1	21	1	a	Ja
Objekt1	22	1	b	Nein
Objekt1	23	1	c	Nein
Objekt1	24	1	v	Nein
Objekt2	21	2	a	Nein
Objekt2	22	2	b	Ja
Objekt2	23	2	c	Nein
Objekt2	38	2	F	Nein



Objekt_ID	MerkmalID	ConstrainID	Variable_Name	WirdBerechnet
Objekt2	86	5	G1	Nein
Objekt3	22	3	h	Nein
Objekt3	24	3	v	Ja
Objekt3	86	5	G2	Nein
Objekt3	87	3	r	Nein
Objekt4	22	4	h	Ja
Objekt4	38	4	F	Nein
Objekt4	87	4	r	Nein
Objekt5	85	5	G0	Nein
Objekt5	86	5	G	Ja
Objekt6	22	6	h	Ja
Objekt6	22	7	h	Ja
Objekt6	24	6	v	Nein
Objekt6	38	7	F	Nein
Objekt6	87	6	r	Ja
Objekt6	87	7	r	Ja

**Tabelle 36** Beispiel, Auszug aus der Tabelle "ConstrRelation"

Für das Objekt6 wurden mehrere Constraints benutzt.

In der Tabelle "ConstrRelation" werden den Objekten und ihren Merkmalen die Constraints und die Variablen aus den Formeln zugewiesen

Das Constraint Nr. 5 koppelt die Eigenschaften der Objekte Objekt2, Objekt3 und Objekt5.

Im Datenfeld "WirdBerechnet" wird eingetragen, welches Merkmal bei der Auswertung von Constraints berechnet werden soll. Der Wert =Ja bedeutet, dass das Merkmal berechnet werden muss,

Der Wert =nein bedeutet, dass das Merkmal als Vorgabegröße betrachtet wird.

## 5.1 Zusammenfassung:

Die hier vorgeschlagene Erfassung der Merkmale und Constraints zeichnet sich durch eine sehr hohe Flexibilität aus.

Die Tabelle "Merkmale" kann noch um weitere Spalten erweitert werden, die z. B. die statistischen Verteilungen der nicht exakten Parameterwerte beschreiben. Um eine hohe Flexibilität zu erreichen, mussten die Constraintsbeschreibungen in drei Tabellen "ConstrRelation", "ConstrFormel" und "Formeln" aufgeteilt werden.

Es ist vorgesehen, Hilfsformulare zu entwickeln, die die Dateneingabe in alle 3 Tabellen gleichzeitig erleichtern sollen.

## 6 ECA-Regeln und das ASN-Datenmodell

Für die Erfassung und Handhabung von ECA-Regeln sind im ASN-Datenmodell mehrere Tabellen (s. Abbildung 10) notwendig

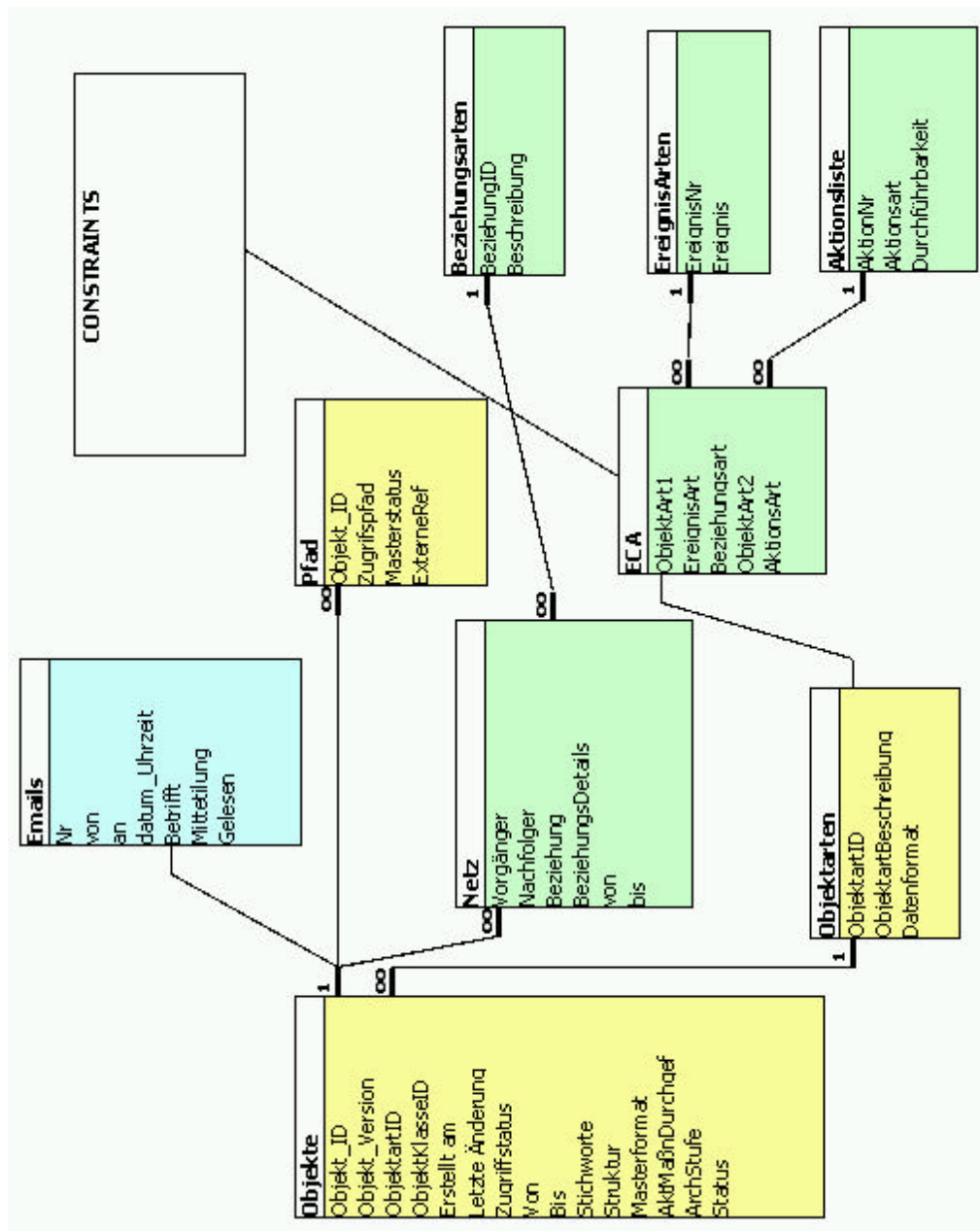


Abbildung 10 Die wichtigsten Tabellen für die Handhabung von ECA-Regeln

AktionNr	Aktionsart	Durchführbarkeit
1	Lesesperre setzen	d
2	Schreibsperre setzen	d
3	Objekt ausblenden	d
4	Objekt löschen	d
5	Nachricht senden an	d,m
6	Änderungszugriff beendet	m
7	Objekt im Dialog prüfen/aktualisieren	d
8	Statuswerte setzen/ändern	m
9	Objektmerkmale im Dialog prüfen/aktualisieren	d
10	Objektmerkmale automatisch prüfen/aktualisieren	d
11	Objekt automatisch prüfen/aktualisieren	d,m
12	Constraints bearbeiten	m
13	Beziehungen festlegen	m
14	Objekt erstellen	m
15	Beziehungstabelle prüfen/ändern	d
16	Im Datenfeld "AktMaßnDurchgef" Wert "ja" setzen	d
17	Im Datenfeld "AktMaßnDurchgef" Wert "nein" setzen	d
18	Datenfeld "Status" Wert "normal" setzen	d
19	Datenfeld "Status" Wert "soll überprüft bzw. geändert werden" setzen	d
20	Im Datenfeld "Zugriffstatus" Wert "frei" setzen	d
21	Im Datenfeld "Zugriffstatus" Wert "im Lesezugriff" setzen	d
22	Im Datenfeld "Zugriffstatus" Wert "in Schreibzugriff" setzen	d
23	Im Datenfeld "Zugriffstatus" Wert "wird geändert" setzen	d
24	Im Datenfeld "Zugriffstatus" Wert "im Lesezugriff und im Schreibzugriff" setzen	d
25	Im Datenfeld "Zugriffstatus" Wert "wurde geändert" setzen	d

Tabelle 37      Tabelle "Aktionsarten"

## Erklärungen zur Durchführbarkeit

d      direkt, automatisch mit einem Programm

d,m    zum Teil automatisch, zum Teil im Dialog (manuell)

m      nur im Dialog, manuell

EreignisNr	Ereignis
1	ObjektLöschen
2	Objekt neu Erstellen
3	Objekt in Lesezugriff
4	Objekt in Änderungszugriff
5	Lesezugriff beendet
6	Änderungszugriff beendet
7	Objektmerkmal geändert
8	Objektmerkmal gelöscht
9	Objektmerkmal neu eingeführt
10	Objekt ersetzt durch ein anderer Objekt
11	Eine Kopie von Objekt wurde erstellt

Tabelle 38      Tabelle "Ereignisarten"

Die Tabellen "Aktionsarten" und "Ereignisarten" müssen noch in Absprache mit SFB 374 Arbeitsgruppen modifiziert und erweitert werden.

Die Unterscheidung zwischen den Begriffen "Ereignis" und "Aktion" ist manchmal schwierig. Was als Aktion am Ende einer Kausalkette gilt, kann als ein Ereignis am Anfang einer anderen Kausalkette gesehen werden.

Die Ereignisse können in der Tabelle "Ereignisliste" registriert werden.

Objekt_ID	Ereignisart	MerkmalNr	Zeitpunkt
1	1	4	11.04.2002 12:30:00

**Tabelle 39** Beispiel, Auszug aus der Tabelle "Ereignisliste"

Bei der praktischen Anwendung von ASN ist auch die Kommunikation zwischen den Teilnehmer sehr wichtig. Die Ausgetauschten Textnachrichten werden automatisch in der Tabelle "emails" gespeichert.

Nr	von	an	datum_Uhrzeit	Betrifft	Mitteilung	Gelesen
1	Objekt1	Objekt2	01.07.2002 08:00:00	Test	Text bis zu 6000 Buchstaben	ja

**Tabelle 40** Beispiel, Auszug aus der Tabelle "emails"

Die wichtigste Komponente in der Handhabung von ECA-Regeln ist die Tabelle "ECA". In dieser Tabelle wird das "ECA-Wissen" so erfasst, dass man die ECA-Regeln mit möglichst kleinem Programmierungsaufwand realisieren kann.

Bei "ECA"-Regeln werden wir mit einer sehr hohen kombinatorischen Vielfalt konfrontiert.

Für die Auswahl/Festlegung der Aktion müssen folgende Aspekte mitberücksichtigt werden:

1. **Objektart** – welche Objektart hat das Objekt, auf das das Ereignis eingewirkt hat?  
Es gibt ca. 36 Objektarten.
2. **Ereignisart** – was ist bei dem Ereignis geschehen? Wurde die in der Tabelle "Objekte" referierte Datei geändert (z. B. ein CAD-Modell) oder wurden die Werte in der Tabelle "Merkmale" geändert?  
Es gibt ca. 11 Ereignisarten und mehrere hundert Merkmale.
3. **Welche Objekte sind mitbetroffen?** In welchen Beziehungen steht das Objekt zu den anderen Objekten? Sind die Beziehungen nur durch die Tabelle "Netz" und/oder auch durch die Constraints beschrieben?  
Hier müssen jeweils ganze Objektmengen (Listen) ermittelt werden. Die Bedingungen ("Condition") werden in den Beziehungen gespeichert.  
Es gibt ca. 38 Beziehungsarten und theoretisch beliebig viele Beziehungen, die implizit durch die Constraints definiert werden können.
4. **Die Auswahl der Aktionsart.** Bei einem Ereignis kommen auch mehrere Aktionen in Frage. Die Aktion muss zu den, durch die Beziehungen mit dem Ereignis-Objekt ver-

bundenen Objekten passen.  
Es gibt ca. 25 Aktionsarten.

Auch abgesehen von der Vielfalt, die durch die Anwendung von Constraints entstehen kann, müssen wir hier mindestens 15048 Situationen unterscheiden und für jede Situation aus ca. 25 Aktionsarten eine oder mehrere passende Aktionen auswählen.

Erst wenn diese Auswahl und die Durchführung der Aktionen in der ANS-Datenbank automatisch abläuft, kann man sagen, dass die ECA-Regeln in der ASN-Datenbank erfolgreich und vollständig erfasst wurden.

Die Unterscheidung von ca. 15048 x 25 Optionen in einem oder mehreren Programmen kaum denkbar ist. Es ist nicht möglich, solche Programme mit einem vertretbaren Zeitaufwand zu entwickeln

Ein Teil der RPD-Semantikinformation legt die zulässigen Kombinationen fest. Dabei müssen folgende zulässigen Kombinationen ermittelt werden:

1. Objektart – Ereignisart
2. Objektart – Beziehungsart – Objektart
3. Objektart Aktionsart
4. Ereignisart Aktionsart

Wenn die vorgesehenen Objektarten, Beziehungsarten und Aktionsarten in den SFB-Arbeitsgruppen endgültig festgelegt werden, können die zulässigen Kombinationen ermittelt und in den Tabellen "OE", "OBO", "OA" und "EA" abgespeichert werden.

Tabelle	Bedeutung
"OE"	Die zulässigen Kombinationen Objektart und Ereignisart
"OBO"	Die zulässigen Kombinationen Objektart, Beziehungsart, Objektart
"OA"	Die zulässigen Kombinationen Objektart und Aktionsart
"EA"	Die zulässigen Kombinationen Ereignisart und Aktionsart

**Tabelle 41 Die Bedeutung der Tabellen "OE", "OBO", "OA" und "EA"**

Objektart	Ereignisart
CAD 3D-Modell	Objekt in Änderungszugriff öffnen
CAD 3D-Modell	ObjektLöschen
NC-Datei	Objekt in Lesezugriff öffnen

**Tabelle 42 Beispiel, Auszug aus der Tabelle "OE"**

ObjektArt1	Beziehungsart	Objektart2
CAD 3D-Modell	B ist abgeleitet von A (Objekt-Objekt)	CAD 3D-Modell
Person	A ist zuständig für B	CAD 3D-Modell

**Tabelle 43 Beispiel, Auszug aus der Tabelle "OBO"**

Objektart	Aktionsart
CAD 3D-Modell	Beziehungen festlegen
CAD 3D-Modell	Datenfeld "AktMaßnDurchgef" Wert "nein" setzen
CAD 3D-Modell	Datenfeld "Status" Wert "normal" setzen

Objektart	Aktionsart
CAD 3D-Modell	Beziehungen festlegen
CAD 3D-Modell	Beziehungstabelle prüfen/ändern

Tabelle 44 Beispiel, Auszug aus der Tabelle "OA"

Ereignisart	Aktionsart
ObjektLöschen	Beziehungstabelle prüfen/ändern
Objekt in Änderungszugriff öffnen	Constraints bearbeiten
Objekt in Änderungszugriff öffnen	Datenfeld "AktMaßnDurchgef" Wert "ja" setzen
Objekt in Änderungszugriff öffnen	Datenfeld "Status" Wert "normal" setzen

Tabelle 45 Beispiel, Auszug aus der Tabelle "EA"

Diese Hilfstabellen erlauben, die Tabelle "ECA" automatisch zu erstellen.

Beispiel: ein SQL-Befehl für die Erstellung der Tabelle "ECA":

```
SELECT DISTINCT OE.Objektart, OE.Ereignisart, OBO.Beziehungsart, OA.Objektart,
OA.Aktionsart INTO ECA
FROM EA INNER JOIN ((OE INNER JOIN OBO ON OE.Objektart = OBO.ObjektArt1)
INNER JOIN OA ON OBO.Objektart2 = OA.Objektart) ON (EA.Aktionsart =
OA.Aktionsart) AND (EA.Ereignisart = OE.Ereignisart)
ORDER BY OE.Objektart, OE.Ereignisart, OBO.Beziehungsart, OA.Objektart,
OA.Aktionsart;
```

Für den Aufbau der Tabelle "ECA" sind die Tabellen "OE", "OBO", "OA" und "EA" nicht unbedingt notwendig, aber sehr hilfreich.

ObjektArt1	EreignisArt	Beziehungsart	ObjektArt2	AktionsArt
CAD 3D-Modell	Objekt in Änderungszugriff öffnen	B ist abgeleitet von A (Objekt-Objekt)	NC-Datei	Objekt im Dialog prüfen/aktualisieren
CAD 3D-Modell	Objekt in Änderungszugriff öffnen	B ist abgeleitet von A (Objekt-Objekt)	NC-Datei	Nachricht senden an
Firma/Unternehmen/Werkstadt	ObjektLöschen	A herstellt B (Firma, Produkt)	Virtueller Prototyp	Beziehungstabelle prüfen/ändern
Firma/Unternehmen/Werkstadt	ObjektLöschen	A kooperiert mit B (Personen, Teams), (Person, Person)	Person	Nachricht senden an
Firma/Unternehmen/Werkstadt	ObjektLöschen	A ist übergeordnet B (Personen, Teams)	Team, Gruppe	Beziehungstabelle prüfen/ändern
Person	Objekt neu Erstellen	A ist Mitglied in B (Personen, Teams)	Team, Gruppe	Beziehungen festlegen

Tabelle 46 Beispiel, Auszug aus der Tabelle "ECA"

ObjektArt1 bezieht sich auf die Objekte, bei denen das Ereignis gewirkt hat.

ObjektArt2 sind die Objekte, die auf der Basis der Beziehungen zum Ereignisobjekt durch das Ereignis mitbetroffen sind. Dabei handelt es sich um die Objekte, an denen eine Aktion durchgeführt werden muss.

Um Speicherplatz zu sparen, werden in der Tabelle "ECA" nur Zahlen gespeichert, die die einzelnen Objektarten, Ereignisarten und Aktionsarten eindeutig identifizieren (s. Tabellen "Objektarten", "Ereignisarten" und "Aktionsarten")

In der Tabelle "ECA" wird nur festgelegt, auf welche Objektart (s. Objektart2) welche Aktionsart angewendet werden soll. Für eine Objektart kommen eventuell mehrere Aktionsarten in Frage.

Dort sind auch Aktionsarten eingetragen, die sich widersprechen (in dem Sinne, dass sie nicht alle angewendet werden können) und aus denen eine Auswahl getroffen werden muss!

Wenn auf einem konkreten ObjektX ein Ereignis stattgefunden hat, können mit Hilfe der Tabelle "Netz" und der Tabelle "ECA" alle mitbetroffenen Objekte ermittelt werden.

In der Tabelle "Netz" werden zum ObjektX anhand der Beziehung Nr. j alle seine Nachfolger, dann die Nachfolger des Nachfolgers usw. ermittelt. Für verschiedene Beziehungsarten (eine andere Nr. j) finden wir jeweils eine andere Menge von direkten und nichtdirekten Nachfolgern.

Diese Mengen müssen nicht unbedingt überschneidungsfrei sein.

Die Ermittlung aller direkten wie auch indirekten Nachfolger eines Objektes wird in der Datenbank mit einem Programm "Aktionsmenge1" durchgeführt. Das Makro erzeugt eine Tabelle, die die Objekt\_ID's aller zu bearbeitenden Objekte enthält.

Bei der Änderung der Objektmerkmale muss nicht nur die Tabelle "Netz" überprüft werden, sondern auch alle Constraints. Nur so kann ermittelt werden, welche Objekte durch die Merkmaländerung mitbeeinflusst sind. Auch für die Ermittlung solcher Objektmengen muss ein Hilfsprogramm ("Aktionsmenge2") geschrieben werden.

Beide Programme "Aktionsmenge1" und "Aktionsmenge" können entweder direkt in der Datenbank, oder in einem externen Zugriffsprogramm integriert werden.

Bei der direkten Integration der Programme in die ASN-Datenbank, wären diese Programme zum Teil datenbankabhängig. Jedoch können die Algorithmen der Programme immer von dem benutzten DB-System unabhängig sein, solange sie sich auf die vorgesehene Datenstruktur für die Beschreibung von Beziehungen und Constraints beziehen.

Wenn die Programme "Aktionsmenge1" und "Aktionsmenge" in einem externen DB-Zugriffsprogramm integriert wären, könnte man sie von der Datenbank unabhängig schreiben.

Nicht alle Aktionen in der Datenbank können sofort und automatisch durchgeführt werden (s. Tabelle 37 - "Aktionsarten"). In der ASN-Datenbank wird oft zuerst nur ein Eintrag gemacht

Die Tabelle "Objekte" enthält für die Handhabung der ECA-Regeln 3 wichtigen Datenfelder:

Das Datenfeld "**Zugriffstatus**"

Merkmalwert:	
Frei	ein neuer Zugriff jeder Art ist möglich
im Lesezugriff	ein weiterer Lesezugriff ist möglich. Ein Schreibzugriff wäre theoretisch denkbar, aber es besteht das

	Risiko, dass der Lesende dann nicht aktuelle Information bekommt.
im Schreibzugriff	weiterer Schreibzugriff ist nicht möglich. Ein Lesezugriff wäre denkbar, aber es besteht das Risiko, dass der Lesende nicht aktuelle Information bekommt.
wird geändert	Das Objekt ist nicht geöffnet. Der Lesezugriff und Schreibzugriff ist möglich, aber es wäre sicherer, zuerst auf die vorgesehene Änderung zu warten. Es werden eventuell auch alle abhängigen Objekte durch die Änderung mitbeeinflusst.
im Lesezugriff und im Schreibzugriff	Situation wie beim Lesezugriff und Schreibzugriff
wurde geändert	Ein Objekt wurde geändert, aber seine Eigenschaften in der Tabelle "Merkmale" wurden noch nicht angepasst. Z.B.: Ein CAD-Modell wurde geändert, aber das Merkmal "Gewicht" noch nicht aktualisiert. Dies gilt auch umgekehrt: Ein Parameter für die Variantenkonstruktion wurde geändert, aber das CAD-Modell wurde noch nicht neu konstruiert.

Tabelle 47

Die Änderungen im Datenfeld "Zugriffstatus" können zum Teil als Ereignisse in Sinne von ECA-Regeln betrachtet werden.

#### Das Datenfeld "**AktMaßnDurchgef**"

Auf dem Objekt hat sich ein Ereignis abgespielt. Dadurch ist die Notwendigkeit entstanden, auch andere Objekte zu aktualisieren (ECA)

Merkmalwert:	
ja	alle Aktualisierungsmaßnahmen auf abhängigen Objekten wurden schon durchgeführt
nein	die Aktualisierungsmaßnahmen müssen noch durchgeführt werden

Tabelle 48

#### Das Datenfeld "**Status**"

Hier werden die Auswirkungen von Änderungen an anderen Objekten, von denen das Objekt abhängig ist, berücksichtigt

Merkmalwert:	
"normal"	Die Auswirkungen der Änderungen anderer Objekte wurden berücksichtigt
"soll überprüft bzw. geändert werden"	Die Auswirkungen der Änderungen anderer Objekte wurden noch nicht berücksichtigt

Tabelle 49

Beispiel für den Ablauf der Vorgänge in der ASN-Datenbank



Die Lage: Objekt1 soll modifiziert werden. Die Information in der ASN-Datenbank soll dabei helfen, alle nicht kompatiblen Änderungen der anderen abhängigen Objekte zu vermeiden und die Verwendung nicht mehr aktueller Informationen über Objekte zu verhindern.

Zeit ↓	Objekt1 z. B. CAD-Model	Objekt2 (ein von den mehreren mitbetroffenen Objekten) z. B. NC-Datei
Ereignis:	Objekt öffnen/ Option Ändern	
Aktion1	Im Datenfeld " <b>Zugriffstatus</b> " Wert "im Schreibzugriff" setzen	
Aktion2	Im Datenfeld " <b>AktMaß-Durchgef</b> " Wert: "nein" setzen	
Aktion3	alle von dem Objekt1 abhängigen Objekte ermitteln (s. Programme "Auswahlmenge1" und "Auswahlmenge2") um die notwendigen Maßnahmen durchführen zu können. (Maßnahmen s. Objekt2)	
Aktion4		Im Datenfeld " <b>Zugriffstatus</b> " Wert "wird geändert" setzen
Aktion5		Im Datenfeld " <b>AktMaß-Durchgef</b> " Wert "nein" setzen
Aktion6		Datenfeld " <b>Status</b> " setzen Wert "soll überprüft bzw. geändert werden"
Aktion7	Nachricht schicken an den Inhaber/Betreuer des Objektes Objekt2	
Aktion8	Modelbearbeitung durchführen	
Aktion9	Im Datenfeld " <b>Zugriffstatus</b> " Wert "wurde geändert" setzen	
Aktion10	CAD-Modell und seine Merkmale zueinander anpassen (z. B. Variantenprogramm für die CAD-Modellgenerierung starten)	
Aktion11	Im Datenfeld " <b>Zugriffstatus</b> " Wert "frei" setzen	
Aktion11+x		Alle vorgesehenen Aktionen (Überprüfungen und Ände-

		rungen) durchführen
Aktion 12+x	Im Datenfeld " <b>AktMaßn-Durchgef</b> " Wert: "ja" setzen	
Aktion 13+x		Im Datenfeld " <b>Zugriffsta-tus</b> " Wert "frei" setzen
Aktion 14+ x		Im Datenfeld " <b>Status</b> " Wert "normal" setzen
Aktion 15+x		Im Datenfeld " <b>AktMaßnDurchgef</b> " Wert "ja" setzen

**Tabelle 50      Aktionen und die Datenfelder "Status", "Zugriffstatus" und "AktMaßnDurchgef"**

Bemerkung zur Aktion3:

Die Tabelle "Bearbliste", die eine Liste der von dem Objekt1 abhängigen Objekte enthält, wird automatisch mit den Programmen "Aktionsliste1" und "Aktionsliste2" erstellt. Eine Aktualisierungsabfrage, die auf der Basis der Tabellen "Objekte" und "Bearbliste" aufgebaut wird, führt die Aktionen 4,5 und 6 automatisch durch.

Die Liste der zuständigen Personen für die Objekte aus der Tabelle "Bearbliste" kann relativ leicht automatisch ermittelt werden. In der Weiterentwicklung des ASN-Projektes ist vorgesehen, die Zuständigen automatisch per e-mail zu benachrichtigen (Aktion7). Die Nachrichten werden automatisch in der Tabelle "emails" archiviert.

**Die vorläufige Analyse der Aufgaben deutet darauf hin, dass durch die vorgeschlagene Erfassung der ECA-Regeln in Tabellen, der Programmierungsaufwand wesentlich reduziert werden kann.**

Es ist notwendig, pro Aktionsart ein Programm, die Programme "Aktionsliste1", "Aktionsliste2", und für die Auswertung von Constraints ein Programm zu schreiben. Diese Programme können dann aus den Datenbankformularen oder von externen Programmen aus gestartet werden.

Die Tabelle "ECA" beschreibt alle möglichen Aktionen, die durch ein Ereignis auf einem Objekt, auf den von ihm abhängigen Objekten durchgeführt werden sollten.

Es gibt Ereignisse, die auf einem Objekt durchgeführt werden und die zu Aktionen auf dem gleichen Objekt führen. Auch in diesem Fall müssen die Optionen systematisch erfasst werden. In der Tabelle "OEA" werden die zulässigen Kombinationen für die Objektart, Ereignisart und Aktionsart zusammengefasst. Diese Tabelle muss noch aufgebaut werden.

## 7 Effizienzsteigerung des ASN durch die Verfeinerung des ASN-Datmodells

Es fällt auf, dass in der vorgelegten Konzeption keine "strengen" Sperren auf die Objekte gesetzt werden. In der Tabelle "Objekte" werden nur Hinweise gespeichert, z. B. steht im Datenfeld "Status" als Wert "soll überprüft bzw. geändert werden". Der potenzielle Bearbeiter des Objektes muss mit anderen Bearbeitern klären, ob und welche Information er jetzt aus dem Objekt übernehmen kann.

Die bisherige Abbildung der Semantik in das Datenmodell ist für eine voll automatisierte Sperrung von Dateien nicht präzise genug!

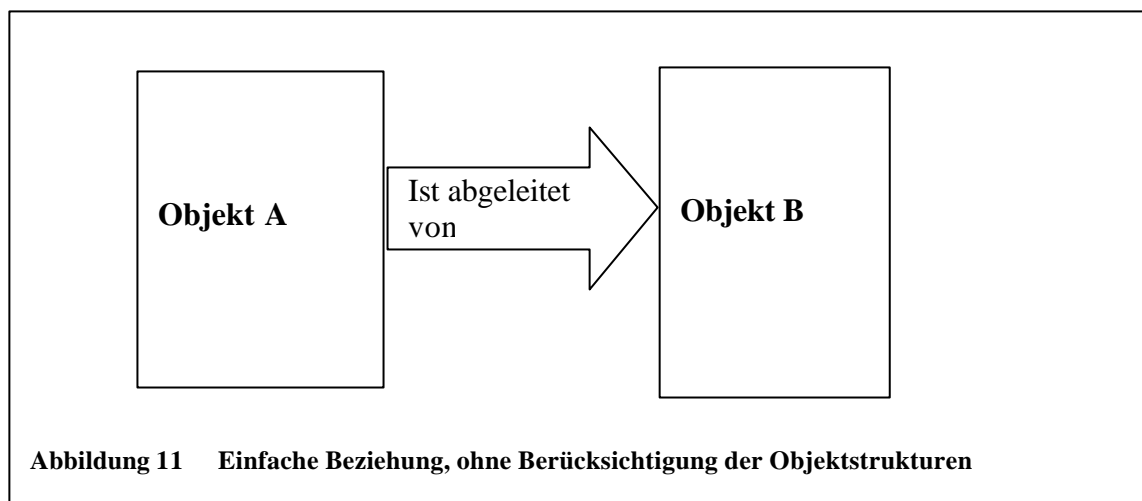
Beispiel:

Objekt B (eine NC-Datei) ist abgeleitet vom Objekt A (ein CAD-Modell). Die Beziehung "ist abgeleitet" lässt erwarten, dass die Änderung des Objekt A auch Änderungen des Objekt B zur Folge haben könnte. Jetzt kommt es darauf an, welche Änderungen tatsächlich durchgeführt werden. Wenn im CAD-Modell nur die Bemaßungen oder die Beschriftungen geändert wurden, ist die abgeleitete NC-Datei ohne Änderung weiter verwendbar und muss nicht zuerst gesperrt werden. Wenn im Objekt A die Geometrie geändert wurde, muss auch die NC-Datei angepasst werden. Vor der Anpassung musste sie für die Weiterverwendung gesperrt werden.

Für die automatische Handhabung von Sperren müssten die Begriffe "Beziehung", "Lesen der Objekte" und "Änderung der Objekte" präzisiert werden.

Statt der Beziehung "ist abgeleitet von" müsste man die Beziehung "ist abgeleitet von Geometriedaten von" einführen.

In folgenden Abbildungen wird als Beispiel die Präzisierung der Beziehung "Ist abgeleitet von" grafisch veranschaulicht.



In der Tabelle "Netz" wird die Beziehung wie folgt eingetragen:

Vorgänger	Nachfolger	Beziehung	BeziehungsDetails	von	bis
Objekt A	Objekt B	5			
...	...	...	...	...	...

**Tabelle 51 Auszug aus der Tabelle "Netz"**

Die innere Struktur der Objekte kann mit der Tabelle "ObjKomponenten" beschreiben werden.

Objekt_ID	Komponente
Objekt A	A1
Objekt A	A2
Objekt A	A3
Objekt A	Geometrie
Objekt A	Bemaßung
Objekt A	Texte
Objekt B	Programmzeilen
Objekt B	Kommentare
Objekt B	Technologiedaten

Tabelle 52 Auszug aus der Tabelle "ObjKomponenten"

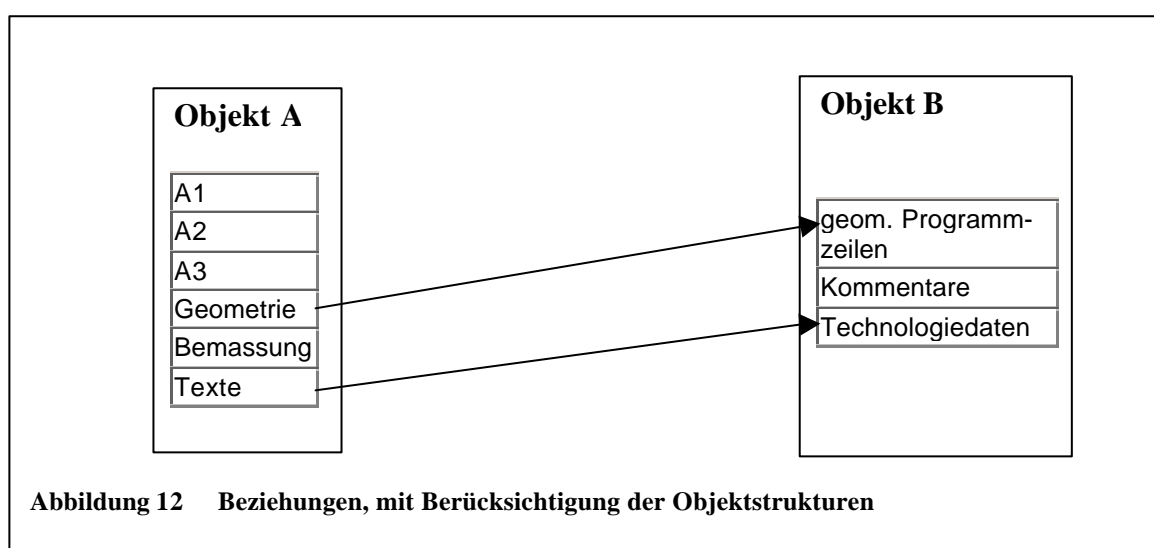


Abbildung 12 Beziehungen, mit Berücksichtigung der Objektstrukturen

Die Abbildung 12 zeigt genauer als die Abbildung 11 wie Objekt B von Objekt A abgeleitet wurde. Diese genauere Information kann in der Tabelle "NetzDet" problemlos beschrieben werden.

Vorgänger	Nachfolger	Beziehung	BeziehungsDetails	von	bis	Komp1	Komp2
Objekt A	ObjektB	5				Geometrie	Programmzeilen
Objekt A	Objekt B	5				Texte	Technologiedaten

Tabelle 53 Auszug aus der Tabelle "NetzDet"

Als Strukturkomponenten betrachten wir hier nur solche Bestandteile des Objektes, die selbst keinen Status als Objekte haben. So kann z.B. der Aufbau einer Baugruppe aus einzelnen Bauteilen bereits mit der Tabelle "Netz" beschrieben werden.

Die Sachlage kann sehr unterschiedlich sein. Manchmal können die Technologiedaten auch von den Geometriedaten abhängig sein.

Es muss noch festgelegt werden, ob man eventuell nur mit der Tabelle "NetzDet" arbeiten wird. Theoretisch wäre es denkbar, auf die Tabelle "Netz" zu verzichten. Die Nachteile und Vorteile dieser Entscheidung werden noch diskutiert.

Die innere Struktur der Objekte muss auch bei der Handhabung der Lese- und Änderungszugriffe berücksichtigt werden. Der Lesezugriff und Schreibzugriff muss durch die Auswahl der Komponenten des Objektes, die man lesen oder ändern will beschrieben werden. Nicht immer braucht man alles.

Anbei einige Beispiele:

Vorgesehener Änderungsvorgang am Objekt A:

Komponente:	wird geändert
A1	x
A2	x
A3	
Geometrie	
Bemaßung	x
Texte	

**Tabelle 54** Änderungsvorgang, Festlegung FA1

Vorgesehener Lesevorgang auf dem abgeleiteten Objekt B:

Komponente:	wird gelesen:
Programmzeilen	x
Kommentare	x
Technologiedaten	x

**Tabelle 55** Lesevorgang, Festlegung FB1

Der Änderungsvorgang FA1 erlaubt auf dem abgeleiteten Objekt B nicht nur den Lesevorgang FB1, sondern auch beliebigen Änderungszugriff. Die Sperrung des Objektes B auf Grund der Information in der Tabelle "Netz" wäre zwar als Vorsichtsmaßnahme richtig, aber es würde die parallele Bearbeitung der Objekte A und B verhindern.

Somit kann man durch die genauere Beschreibung der Beziehungen zwischen den Objekten und die Analyse der Lese- und Schreibzugriffe die Parallelität der Objektbearbeitung erhöhen und dadurch die Prototypenentwicklung beschleunigen.

Die Analyse der Änderungsvorgänge ist auch Grundlage für die parallele Bearbeitung von mehreren identischen Kopien eines Objektes. Nehmen wir als Beispiel eine CAD-Zeichnung.

	Kopie 1	Kopie2	Kopie 3	Kopie 4
Komponente:	wird geändert	wird geändert	wird geändert	wird geändert
A1	x			
A2	x			
A3		x		
Geometrie, Bereich1		x		
Geometrie, Bereich2			x	
Geometrie, Bereich3				x
Bemaßung	x			
Texte		x		
Zeichnungskopf				x

**Tabelle 56 Beispiel für die parallele Bearbeitung identischen Kopien einer CAD-Zeichnung**

Eine sinnvoll festgelegte Aufgabenteilung bei der Bearbeitung der Zeichnungskomponenten erlaubt es, die Kopien der Zeichnung automatisch zu synchronisieren. In Rahmen einer Diplomarbeit ist vorgesehen, ein Programm entwickeln zu lassen, das die geänderten Kopien einer Zeichnung, die in DXF-Format gespeichert wurden, automatisch synchronisieren kann.

Wenn in einer ASN-Datenbank die in den Tabellen 52-56 beschriebene Detailinformation fehlt, muss dieses Informationsdefizit durch intensive Kommunikation zwischen den beteiligten Personen kompensiert werden.

## 8 ASN-Datenmodellierung mit MS Access

Die MS Access Datenbank kann als eine **"Zwischenstation"** gesehen werden. Die Datenübertragung und die Modellübertragung in andere Datenbanksysteme ist jederzeit möglich.

Was spricht in der Testphase für MS Access?

- Die MS Access Software ist praktisch an jedem Arbeitsplatz vorhanden
- Die notwendige Einarbeitungszeit in MS Access Benutzung ist kurz (ca. 3 Stunden)
- Das Datenmodell kann leicht aufgebaut und modifiziert werden
- Die Handhabung der Daten in MS Access Tabellen ist sehr einfach. Die Daten können direkt in den Tabellen bearbeitet werden. Es können sehr leicht beliebig komfortable, benutzerspezifische Formulare erstellt werden.
- Für die Modifikationen des Datenmodells und für die Datenerfassung/Datenverarbeitung sind keine Programmierungskennnisse notwendig
- Die benutzerspezifischen Sichten (Views) der Datenbank lassen sich sehr leicht erstellen
- Alles, was mit MS Access machbar ist, lässt sich mit besseren Systemen auch realisieren.
- MS Access kennt die sog. "verknüpfen Tabellen". Dadurch ist die Aufteilung der Daten in physikalisch unterschiedliche Datenbanken und sogar Datenbanken verschiedener Art möglich. Die ODBC Treiber erlauben z.B., die Daten in einer Dbase IV Datenbank zu speichern, aber in MS Access zu sehen und zu bearbeiten. Die Leistung der Treiber für verschiedene Datenbanksysteme ist sehr unterschiedlich
- Die ASN-Testbenutzer können Ihre eigenen Datenbanken benutzen und einen Teil der Informationen als verknüpfte Tabellen in der ASN-Datenbank bereitstellen.
- Umgekehrt ist es auch möglich, einen Teil der ASN-Daten als verknüpfte Tabellen in einer externen Datenbank zu verwenden. Die Abbildung 13 zeigt, dass alle Clients auf die Information in beiden Datenbanken zugreifen können
- Eine direkte Integration der Daten und Methoden, wie bei den objektorientierten Datenbanken ist nicht möglich. Dies ist jedoch prinzipiell kein Nachteil. Die Daten können durch Formulare bearbeitet werden. In die Formulare und die Datenfelder der Formulare können beliebige Methoden integriert werden.
- Die integrierten Methoden können zum Teil in VBA (Visual Basic for Applications), zum Teil in SQL und auch in der Kombination von VBA und SQL geschrieben werden.
- Die VBA-Programme wären MS Access spezifisch und nur als Algorithmen in anderen Datenbanksystemen verwendbar. Hier bietet sich an, die Algorithmen in UML zu dokumentieren, um sie dann in anderen Datenbanksystemen in Programme umzusetzen

- Die SQL-Programme könnte man leicht modifiziert in andere Datenbanksysteme übernehmen.

Theoretisch wäre es denkbar, die ASN-Funktionalität komplett mit Hilfe der intelligenten MS Access-Formulare zu realisieren.

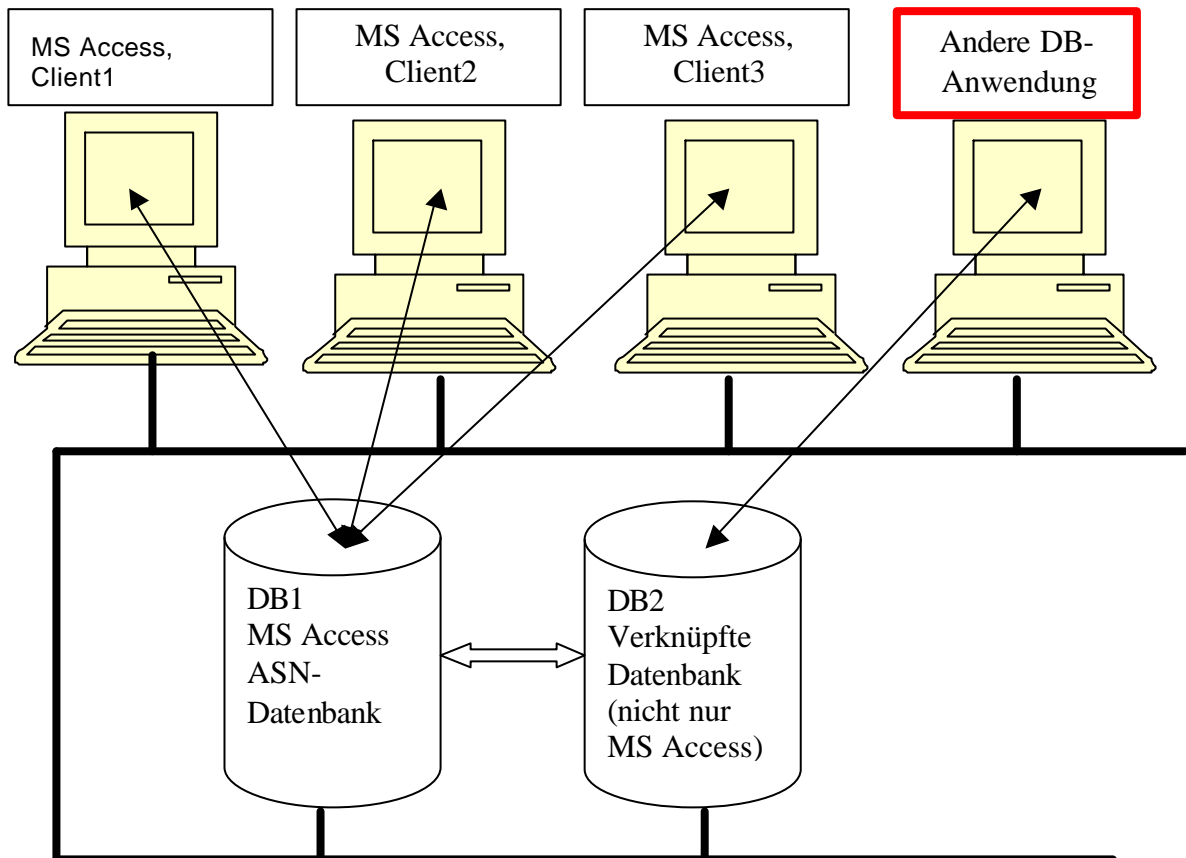


Abbildung 13 Zugriff auf die MS Access Datenbank mit Hilfe der MS Access Software

Die ASN-Benutzer könnten als Client-Software MS Access benutzen und in der vernetzten Umgebung auf eine zentrale MS Access ASN-Datenbank zugreifen. In den benutzerspezifischen Zugriffsformularen wird entschieden, ob und welche Dateien der Benutzer öffnen und bearbeiten kann. Von dort aus werden auch die notwendigen Anwendungen gestartet.

Die Formularereignisse (z. B. Wertänderungen in Datentextfeldern) können im Sinne der ECA-Regeln weitere notwendigen Aktionen automatisch starten.

Allerdings sind unabhängig von der Benutzung der Formularen auch andere Zugriffsmöglichkeiten und Nutzungsmöglichkeiten der ASN-Datenbank möglich!

Die Benutzung von verknüpften MS Access-Datenbanken erlaubt, den Zugriff auf die ASN-Datenbank benutzerspezifisch zu gestalten. Jeder Benutzer hat seine eigene Access-Datenbank, die seine spezifischen Abfragen, Formulare und Berichte enthält (Sicht der Datenbank). Von dieser Datenbank aus wird auf die verknüpften Tabellen der gemeinsamen ASN-Datenbank zugegriffen. Jedoch lassen sich die benutzerspezifischen Zugriffe auf die zentrale ASN-Datenbank auch anders realisieren.



Für die Verbesserung der Performance und Transaktionsicherheit kommt anstelle von MS Access z.B. die Benutzung des MS SQL-Servers in Frage.

## 8.1 Datenschnittstellen und Programmschnittstellen des Systems MS Access

Das Programm MS Access kann die Datentabellen in verschiedenen Formaten exportieren und importieren, z.B. auch als reine Textdateien. Es ist aber auch möglich, Texttabellen, die als selbständige Betriebssystemfiles existieren, mit Tabellen in einer Access-Datenbank zu verknüpfen. So könnte eine Anwendung, die keine Access-Programmschnittstelle besitzt, die Daten in eine verknüpfte Texttabelle (Betriebssystemdatei) schreiben und lesen und dadurch die Information in der ASN-Datenbank modifizieren und nutzen. Diese Tabelle kann auch programmtechnisch in der ASN-Datenbank ausgewertet werden. Die direkte Benutzung von verknüpften Text-Tabellen oder HTML-Tabellen ist ohne Zusatzsoftware zu wenig komfortabel.

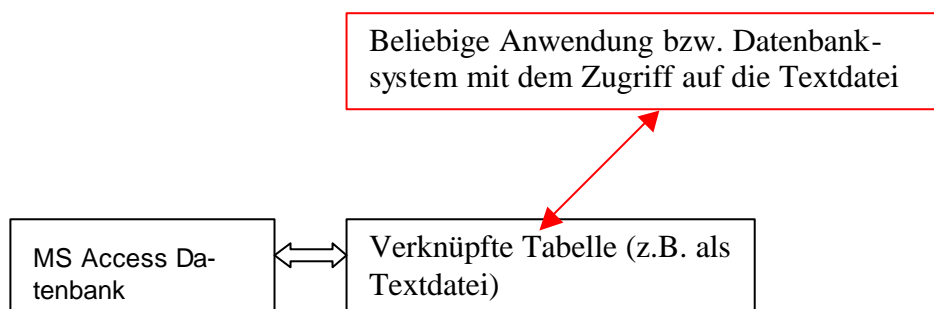


Abbildung 14 Die Kommunikation mit der MS Access Datenbank mit Hilfe einer Texttabelle

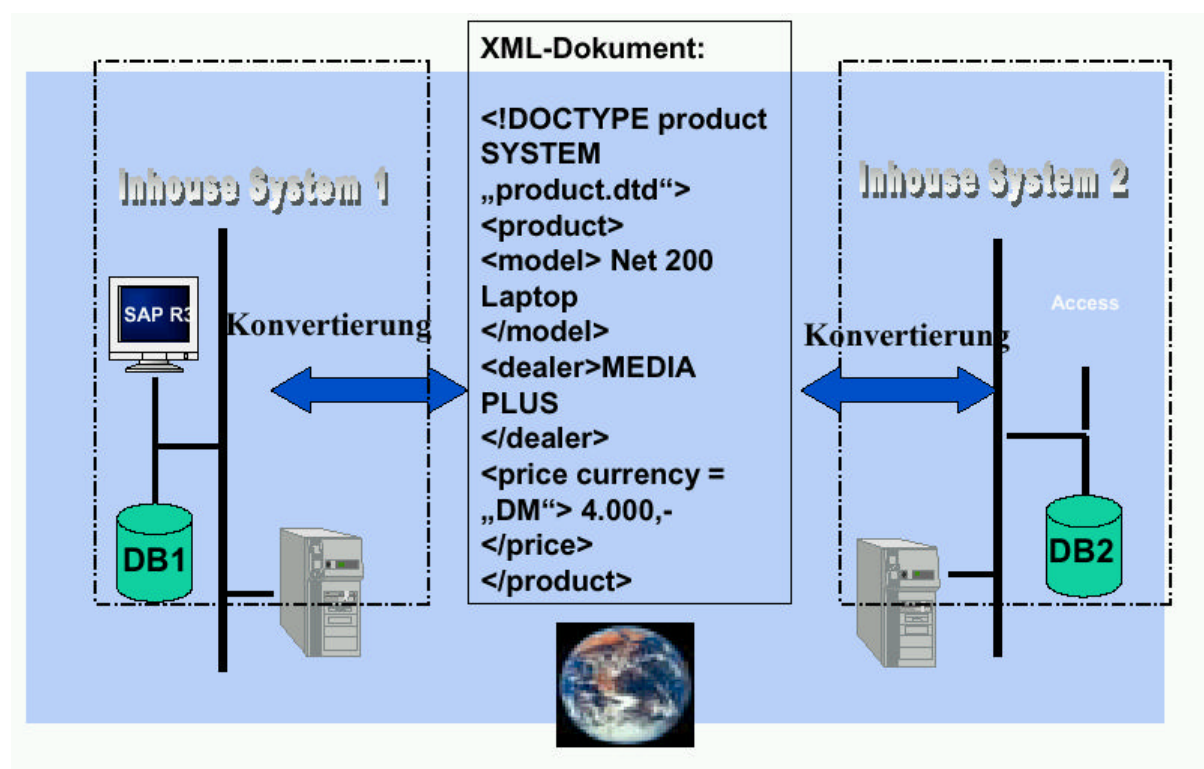


Abbildung 15 Datentransfer zwischen verschiedenen Datenbanken mit Hilfe der XML-Dokumente.  
Quelle: Prof. B. Breutmann, IT Kompaktkurs Datenbanken

Für den Datenaustausch zwischen MS Access-Datenbanken und anderen Datenbanksystemen könnte auch die XML-Datenschnittstelle benutzt werden (s. Abbildung 15). Dieser Vorgang

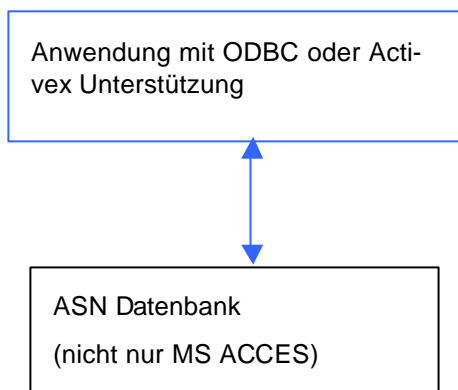
setzt jedoch XML-Konvertierungsprogramme voraus. Solche Programme müssten für MS Access 2000 zuerst geschrieben werden.

In einer MS Access Datenbank können VBA-Programme integriert werden. Wenn eine solche Datenbank mit MS Access geöffnet wurde, können diese Programme automatisch (z. B. als Reaktion an die Formularereignisse) oder manuell gestartet werden. Die Programme können Datenbankstrukturen ändern, Daten in den Tabellen lesen, in die Tabellen schreiben, Daten importieren und exportieren.

Zusätzlich dazu können die Programme mit anderen Anwendungen auf der Basis des TCP/IP Protokolls kommunizieren. Somit wäre es prinzipiell möglich, auch mit UNIX-Anwendungen zu kommunizieren und Daten auszutauschen.

Die MS Access Datenbank kann auch ohne MS Access Software, durch externe Programme gesteuert genutzt werden. Die gängigen Programmiersprachen (Visual Basic, C++, Java) erlauben es, direkt auf die Objekte der ASN-Datenbank zuzugreifen. Somit kann man direkt, aus der Anwendung heraus in der Datenbank Abfragen durchführen, Daten zu modifizieren, die Zugriffsberechtigungen für Objekte abzufragen und zu ändern.

Die Voraussetzung für den Zugriff sind entweder ODBC-Treiber, oder die ActiveX-Bibliotheken. So ist es möglich, in einer CAD-Anwendung (z. B. AutoCAD) die geometrische Merkmale der Objekte abzufragen und als Eigenschaften in die ASN-Datenbank einzutragen.



**Abbildung 16 Direkte Ansteuerung einer Datenbank durch externe Programme**

Die Ansteuerungsmöglichkeiten der ASN-Datenbank durch externe Programme erlaubt es, einen Teil der Funktionalität (Methoden), die in den Objekten oder Formularen der Datenbank integriert sind, in externe Programme zu verlagern.

Die externen Anwendungsprogramme können auch so aufgebaut werden, dass sie die ASN-Datenbank mit Hilfe der SQL- Befehle nutzen. So konzipierte Programme können leicht geändert und für andere Datenbanksysteme benutzt werden.

Die Funktionalität eines Anwendungsprogramms kann in mehrere kooperierende Programme aufgeteilt werden. Ein in Hintergrund laufendes Programm (Agent) kommuniziert direkt mit der Datenbank, die Clients der Benutzer kommunizieren nur mit dem Agent-Programm.

Das Agent-Programm kann auch direkt in die Datenbank integriert werden. Ein solches Programm kann automatisch, z. B. bei Öffnen der Datenbank gestartet werden. Dann ist es aber notwendig, die ASN-Datenbank von einer Workstation aus mit MS Access zu öffnen.

Im Agent-Programm können auch die Zugriffsberechtigungen der Benutzer gehandhabt werden. Die einzelnen Client-Programme werden benutzerspezifisch gestaltet.

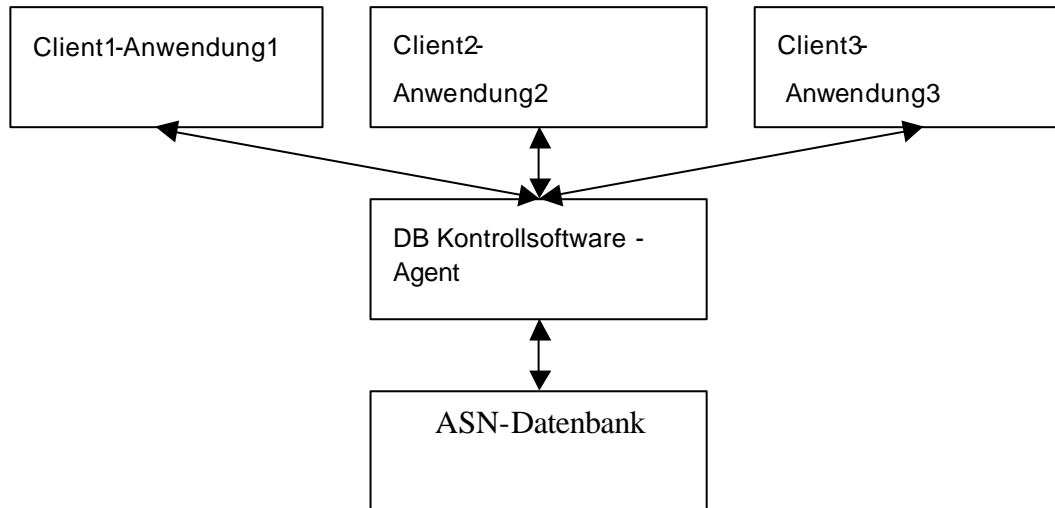


Abbildung 17 Indirekte Ansteuerung einer Datenbank mit Hilfe der Agent-Kontrollsoftware

## 8.2 Heterogene Netze und das ASN

Die Benutzung der Agentensoftware erlaubt es, auch in heterogenen Netzwerken und im Internet zu arbeiten. Wichtig ist, dass die Client-Software auf TCP/IP-Basis mit der Agent-Software kommunizieren kann.

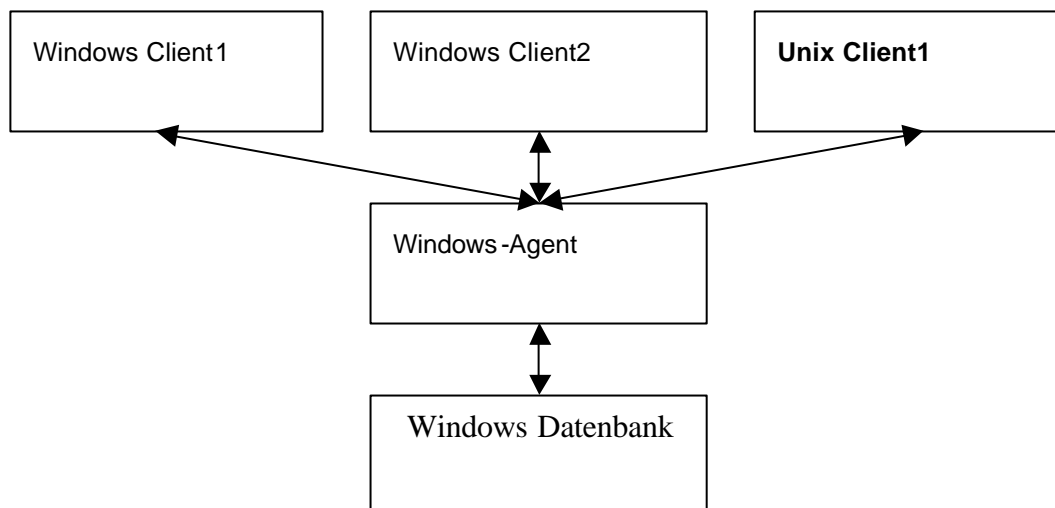


Abbildung 18 Datenbanksteuerung im heterogenen Netzwerkumfeld

Es ist auch denkbar, eine UNIX-Datenbank zu verwenden. Dann muss auch der Kontrollagent als UNIX-Software bereitgestellt werden.

Bei der Kommunikation zwischen den Anwendungen auf UNIX-Clients und Windows-Datenbanken können mehrere Agenten zwischengeschaltet werden. Ein Beispiel für eine solche Kommunikation ist Easysoft ODBC-ODBC Bridge (s. Abbildung 19)

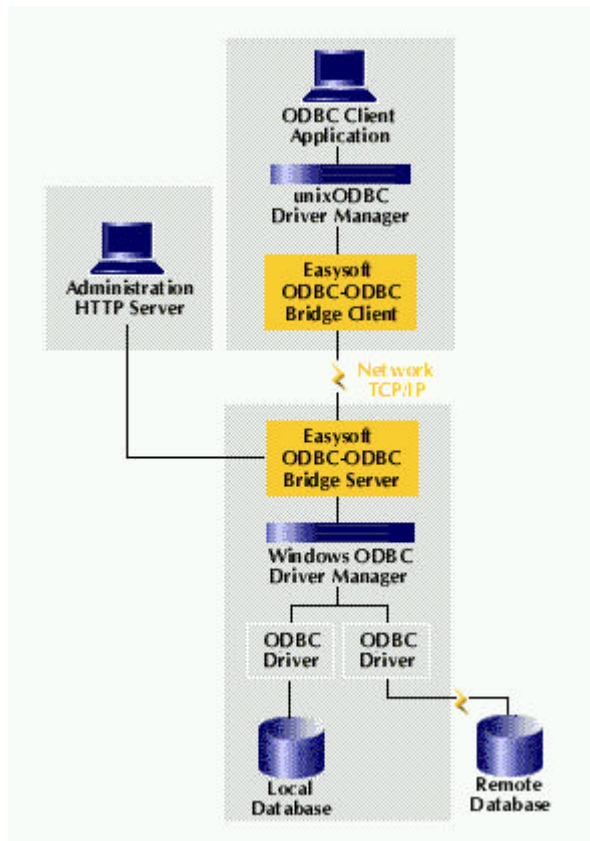


Abbildung 19 ODBC-ODBC Bridge

### 8.3 Zusammenfassung

Die Benutzung einer MS Access ASN-Datenbank wäre am einfachsten und optimal, wenn alle ASN-Benutzer mit Windows-Anwendungen und mit MS Access als Client-Software arbeiten würden. Mit einem durchschaubaren Zusatzaufwand kann erreicht werden, dass auch UNIX-Anwender mit einer Windows-Datenbank arbeiten können.

Mittelfristig wird eine ASN-Lösung mit eigenem Internet Server und Applikationsserver angestrebt. Bei dieser Lösung kann auch den UNIX-Anwendern temporär, auf Internet-Basis, Windows-Software wie z. B. MS Access bereitgestellt werden.

## 9 Zeitplan für die 1. Testphase (Vorschlag)

1. In einer Informationsveranstaltungen wird das vorgelegte Datenmodell ausführlich erklärt und diskutiert (bis Ende Juli 2002). Ca. 2-3 Termine werden durch Absprache mit den Teilnehmern festgelegt.
2. Festlegung des Datenmodells in Absprache mit den SFB 374 Arbeitsgruppen  
(ca. 7-8/2002)
3. Implementierung des Datenmodells in eine MS Access Datenbank  
(ca. 7-8/2002)
4. Bereitstellung der notwendigen Formulare für die Datenbearbeitung  
(ca. 8-9/2002)
5. Übernahme der Testdaten von Arbeitsgruppen durch die Datenschnittstelle und Dialogformulare  
(ca. 9/2002)
6. Bereitstellung der Demoprogramme für die externe Ansteuerung der ASN-Datenbank  
(ca. 10/2002)
7. Aufbau der ersten aktiven Komponenten, Beispiele für die ECA-Regeln, die Handhabung von Constraints  
(ca. 10/2002)
8. Anbindung der Unix-Anwendern durch die Datenschnittstellen und Kommunikationsprogramme  
(ca. 11-12/2002)
9. Gemeinsame Benutzung der ASN-Datenbank als Vorbereitung für die Begehung im Sommer 2003  
(ca. ab. 02.2003)