

Methodik V
Dozent: Prof. Dr. Knorz
SS 99

ER-Modellierung

Eine Bibliographie und Sammlung
von Werkzeugen

Nermin Altas
nermin.altas@planet-interkom.de
FB IuD, 6. Semester

Steven Wolf
boeserwolf@bigfoot.com
FB IuD, 6. Semester

Ralph Bühr
ralph@buehr.com
FB IuD, 6. Semester

Inhaltsverzeichnis

1	EINLEITUNG.....	4
1.1	CASE-TOOLS.....	4
2	DATENBANK-ENTWURF	6
2.1	ENTWURFSPHASEN.....	6
2.2	ANFORDERUNGEN AN EIN DATENBANKENTWURFS-TOOL	7
3	SOFTWAREPAKETE.....	10
3.1	ALLGEMEINES.....	10
3.2	ÜBERSICHT.....	11
3.3	DAS BEISPIEL BIBLIOTHEK.....	12
3.4	DBMAIN.....	13
3.4.1	<i>Allgemeines.....</i>	<i>13</i>
3.4.2	<i>Historie.....</i>	<i>13</i>
3.4.3	<i>Merkmale von DBMAIN Version 4.1.....</i>	<i>13</i>
3.4.4	<i>Eigenschaften des Programms.....</i>	<i>14</i>
3.4.5	<i>Systemvoraussetzungen.....</i>	<i>14</i>
3.4.6	<i>Installation.....</i>	<i>14</i>
3.4.7	<i>Einschränkungen der Demo-Version.....</i>	<i>15</i>
3.4.8	<i>Gesamtansicht.....</i>	<i>15</i>
3.4.9	<i>Konzeptioneller Entwurf.....</i>	<i>17</i>
3.4.10	<i>Fazit.....</i>	<i>18</i>
3.5	DEZIGN.....	19
3.5.1	<i>Allgemeines.....</i>	<i>19</i>
3.5.2	<i>Merkmale von DeZign 1.11.....</i>	<i>19</i>
3.5.3	<i>Funktionalität.....</i>	<i>19</i>
3.5.4	<i>Berwertung.....</i>	<i>19</i>
3.5.5	<i>Gesamtansicht.....</i>	<i>20</i>
3.5.6	<i>Systemanforderungen.....</i>	<i>21</i>
3.5.7	<i>Einschränkungen der Demo-Version.....</i>	<i>21</i>
3.6	EASYSER/EASYOBJECT	22
3.6.1	<i>Einleitung.....</i>	<i>22</i>
3.6.2	<i>Beschreibung.....</i>	<i>22</i>
3.6.3	<i>Installation.....</i>	<i>24</i>
3.6.4	<i>Das Erscheinungsbild.....</i>	<i>24</i>
3.6.5	<i>Die Benutzerverwaltung.....</i>	<i>25</i>
3.6.6	<i>Die Bedienung.....</i>	<i>25</i>
3.6.7	<i>Notationen.....</i>	<i>25</i>
3.6.8	<i>Kritikpunkte.....</i>	<i>28</i>
3.6.9	<i>Fazit.....</i>	<i>28</i>
4	AUSBLICK.....	29
4.1	OBJEKTORIENTIERTE ENTWICKLUNG.....	29
4.2	UML.....	29
5	BIBLIOGRAPHIE	31

6 LINKS	35
6.1 CASE	35
6.2 OO	35
6.3 UML.....	35
7 QUELLEN	35
8 INDEX	36
9 ANHANG	37

1 Einleitung

Diese Ausarbeitung gliedert sich in fünf Teile:

Zunächst soll ein Überblick über CASE-Tools im allgemeinen gegeben werden. (Was sind CASE-Tools, warum wurden CASE Tools eingeführt, etc.)

Im zweiten Abschnitt werden die Datenbank-Entwurfsphasen erläutert, die die Grundlage für alle CASE-Tools bilden. Außerdem soll ein Überblick über die Anforderungen an ein CASE-Tool gegeben werden.

Im dritten Abschnitt stehen drei exemplarisch ausgewählte CASE-Tools im Mittelpunkt. Diese drei Tools sollen einen Überblick über den Funktionsumfang gängiger CASE-Werkzeuge zur ER-Modellierung geben.

Der vierte Abschnitt zeigt die mögliche Weiterentwicklung von CASE-Tools zur Datenbankentwicklung in Hinblick auf die Objektorientierung und Unterstützung der "Universal Modeling Language (UML)" auf.

Daran schließt sich der fünfte Abschnitt, die Bibliographie der CASE-Werkzeuge, an. In der Bibliographie werden die CASE-Tools und deren Hauptmerkmale in einer Art Marktübersicht mit ihrer Bezugsquelle dargestellt.

1.1 CASE-Tools

In der Informationsverarbeitung und Datenbankentwicklung gelingt es trotz neuer Softwaretechnologien nicht, hochwertige Softwaresysteme schnell und preiswert genug zu entwickeln, um die jährlich wachsende Nachfrage zu befriedigen.

Seit Mitte der 60er Jahre redet man in der Informationsverarbeitung daher von einer *Softwarekrise*.

Diese *Softwarekrise* ist durch folgende Merkmale gekennzeichnet:

- In der Regel keine termingerechte Fertigstellung
- Regelmäßige Überschreitung der geschätzten und bewilligten Entwicklungsmittel
- Höhere Wartungskosten als geschätzt
- Mangelhafte Gesamtqualität der Softwareprodukte

Verschärft wird diese Krise durch die dramatisch zunehmende Komplexität der Softwareprodukte (viele Anwendungen bestehen inzwischen aus mehreren Millionen Programmcode-Zeilen) und die Vergrößerung der Einsatzgebiete in der Informationstechnik.

Auch bei der Datenbankentwicklung findet sich heute eine derart hohe Komplexität der Aufgabenstellungen (E-Commerce verteilte Datenbanken, Online Transaction Processing (OLTP), etc.), daß diese mit Standardmitteln nicht mehr zu bewältigen sind.

Als Ausweg aus der Krise werden Lösungen angeboten, die sich in erster Linie auf die Verbesserung der Methoden und Werkzeuge der Software-Entwicklung beziehen und sich auf die Entwicklungsphasen des Software-Lebenszyklus

konzentrieren. Beim Software-Lebenszyklus handelt es sich um den Zeitraum von der Idee der Software, seine Entwicklung und Wartung bis zur Ablösung der Software durch ein neues leistungsfähigeres Produkt. Den Lebenszyklus unterteilt man üblicherweise in 5 Phasen:

- Analyse
- Design
- Implementierung
- Test
- Wartung

Ein Weg, die obengenannte Softwarekrise anzugehen, ist seit Ende der 80er Jahre die verstärkte Benutzung von CASE-Tools bei der Softwareentwicklung. Man unterscheidet zwischen einzelnen CASE-Werkzeugen und CASE-Werkzeugumgebungen². Bei letzteren handelt es sich um eine Gruppe integrierter CASE-Werkzeuge, die alle Phasen des Software-Lebenszyklus im Zusammenhang sehen und eine direkte Unterstützung für jede einzelne Phase anbieten.

Viele moderne Applikationen werden auf objektorientierten Prinzipien (Klassen, Methoden, Vererbung) basierend entwickelt. Objektorientierte CASE-Tools stellen objektorientierte Modellierungsnotationen und -methoden zur Verfügung und generieren sogar Teile objektorientierter Applikationen. Neue Versionen dieser Tools unterstützen Programmiersprachen wie z.B. JAVA und objektorientierte Datenbanken wie POET und ORACLE 8i direkt. Auf das Thema Objektorientierung wird im Kapitel 4 nochmals eingegangen.

CASE Tools bieten dem Entwickler vor allem dann Vorteile, wenn es sich um mittlere bis größere Projekte handelt. Immer weiter steigende Ansprüche der Anwender in Bezug auf Komfort und Funktionsumfang führen zu bisher ungeahnten Dimensionen der Komplexität der Anwendung. Je größer das Projekt, desto wichtiger ist die lückenlose Benutzung von CASE-Tools über alle Ebenen der Entwicklung hinweg.

Im Rahmen dieser Ausarbeitung werden wir uns auf einzelne CASE-Tools für den Datenbankentwurf konzentrieren. Die Evaluierung und Beurteilung komplexer CASE-Werkzeugumgebungen wie z.B. den "ORACLE Designer" für Oracle 8i, die außer der Datenbankentwicklung auch die Software-Entwicklung umfassen, würden den Rahmen dieser Ausarbeitung bei weitem sprengen.

¹ Computer Aided Software Engineering
² CASE-Environments

2 Datenbank-Entwurf

2.1 Entwurfsphasen

Der Entwurf einer Datenbank gliedert sich in folgende Phasen:

1. In der **Anforderungsanalyse** werden zunächst die Anforderungen der Nutzer der zu entwerfenden Datenbank ermittelt. Diese Analyse kann mittels Interviews mit den einzelnen Benutzern oder durch Fragebogenaktionen durchgeführt werden. Die Ergebnisse der Anforderungsanalyse werden i.d.R. in einem Anforderungsdokument (User Requirements Document) schriftlich festgelegt. Dieses Dokument muß von den Nutzern abgenommen werden und bildet dann die Grundlage für den folgenden Entwurf.
2. Im **konzeptionellen Entwurf** wird der Ausschnitt der realen Welt, der in der Datenbank gespeichert werden soll, mit Hilfe eines Datenmodells dargestellt, nämlich des Entity-Relationship-Modells (ERM). Da für das Verständnis dieses Datenmodells Vorkenntnisse notwendig sind, kann es trotz seiner Eignung als Entwurfsmittel nur schlecht als Basis für die Verständigung mit dem Datenbanknutzern dienen.
3. Im **logischen Entwurf** erfolgt eine Umsetzung des in der vorherigen Phase erzeugten Datenmodells in das zu verwendende Datenbankschema. Meist wird hierbei das relationale Modell benutzt. Aus dem Entity-Relationship-Modell werden also direkt die entsprechenden Relationentabellen generiert. Diese Tabellen müssen nun auf Einhaltung der Normalformen überprüft werden. Normalformen dienen der Vermeidung von Operationsanomalien, die zu Inkonsistenzen im Datenbankinhalt führen können. Relationentabellen, die nicht den Normalformen genügen, müssen entsprechend umgeformt werden.
4. Im **physischen Entwurf** wird entsprechend zu den generierten Relationentabellen der Datenbankcode für das zu verwendende Datenbanksystem erzeugt.
5. Die **Implementierungsphase** besteht aus der tatsächlichen Implementierung und Installation der Datenbank auf dem verwendeten Datenbanksystem.

2.2 Anforderungen an ein Datenbankentwurfs-Tool

Generell sind folgende Funktionen bei der Datenbankentwicklung notwendig bzw. nützlich:

Funktion	Vorteil
Generierung von SQL für mehrere Datenbanksysteme (Oracle, DB2, Informix, INGRES etc.)	Unabhängigkeit von einem RDBMS-Hersteller und dessen proprietären Modellierungs-Tools
Graphischer ER-Editor	Visualisierung der Relationen hilft bei der Entwicklung
Attribut-Tabellen-Editor	Definiert Attribute für die SQL-Generierung durch eine mit der Entity verbundenen Attribut-Tabelle
Automatische Wandlung des konzeptionellen Modells in ein logisches und physisches Modell	Eliminierung von Flüchtigkeitsfehlern
Automatische Normalformwandlung	Wandlung in die 1NF im logischen Modell
Automatischer Korrektheits- und Konsistenz-Check über alle Modelle und Ansichten hinweg	Vermeidung von Fehlern beim Datenbankentwurf
Speicherung aller Entities, Relationen, Attribute, Daten, Regeln, usw. in einem "Repository" (Data Dictionary)	
Automatische Erstellung von Fremdschlüsseln, Indizes und Zugriffsschlüsseln bei der SQL-Ausgabe	Erstellung einer kompletten Datenbank
Report-Generierung	Konsistente Dokumentation
Versionskontrolle	Protokollierung der Nutzeraktionen

Zusätzlich sind folgende Funktionen wünschenswert:

Funktion	Vorteil
Unterstützung der Anforderungsanalysephase (z.B. durch "Wizards") und Integration in die speziellen Anforderungen und Gegebenheiten des DB-Systems	Integrierte Entwicklungsumgebungen sind meist besser aufeinander abgestimmt als einzelne Tools
Objektorientierte Datenbankentwicklung	Zunehmend wichtiger, siehe Kapitel "objektorientierte Entwicklung"
Integration von Publishing-Tools (z.B. Adobe Framemaker oder Interleaf) zur automatischen Generierung von Dokumentation und Reports	Zeitersparnis, Sicherstellung der Korrektheit der Dokumentation
Freie Datentyp-Definition	Maximale Flexibilität
Automatische Optimierung des SQL-Codes	Performance-Optimierung
Aufteilen komplexer Modelle in mehrere Diagramme	Einzelne Diagramme zeigen jeweils einen Aspekt des Gesamtbildes -> Erhöht die Verständlichkeit des ER-Modells auch für Dritte
Freier Zoom in der graphischen Ansicht	Variable Zoomfaktoren erhöhen die Übersichtlichkeit
Freie Farbwahl, Fontwahl und Fontgrößenwahl in der Darstellung	Beliebiges Einfärben der Entities / Relationen / Attribute sowie die freie Wahl der Schrift und Schriftgröße kann die Übersichtlichkeit erhöhen
Import vorhandener Datenbanken in möglichst vielen verschiedenen Formaten und graphische Darstellung deren ER-Modells (Re-Engineering)	Nützlich zur Optimierung bzw. Korrektur / Aktualisierung vorhandener Datenbanken
Integrierte ODBC-Schnittstelle	In der Wintel ³ -Welt weit verbreitete, weitgehend standardisierte Datenbankschnittstelle
Datenaustausch mit anderen CASE-Tools	Import und Export von Fremdprogrammen
Unterstützung strukturierter Design-Methoden	Unterstützung z.B. der Yourdan/Constantine und Page-Jones Methoden

³ WIN (dows) & (in)TEL

Die hier vorgestellten Tools decken unterschiedliche Bereiche des Datenbankentwurfs ab, meist jedoch nur den

- konzeptionellen Entwurf
(Bereitstellung eines graphischen Editors, der die Symbole des Entity-Relationship-Modells zur Verfügung stellt, so daß Diagramme direkt am Rechner gezeichnet werden können)
- den logischen Entwurf
(Überprüfung der Relationentabellen auf Einhaltung der Normalformen, die aus einem Entity Relationship (ER) Diagramm entstehen, das zuvor mit dem graphischen Editor gezeichnet wurde)
- und den physischen Entwurf
(Automatische Generierung des Datenbankcodes (der jeweilige SQL-Dialekt) für das gewählte Datenbanksystem aus dem ER-Diagramm)

Weitergehende Bereiche wie z.B. die

- Entwicklung des GUI (Graphical User Interface)
- Generierung von Reports

Werden von günstigen Tools meist nicht abgedeckt, und müssen durch andere Tools bzw. mit dem Datenbanksystem selbst vorgenommen werden.

Wie bei jeder anderen komplexen Software auch ist bei CASE-Tools eine intensive Einarbeitung in das spezielle Tool nötig.

3 Softwarepakete

3.1 Allgemeines

Da eine umfassende Darstellung aller vorhandenen Softwarepakete zum Thema "ER-Modellierung" aus naheliegenden Gründen nicht möglich ist, haben wir uns entschlossen, drei Programme auszuwählen und diese näher zu evaluieren.

Bei der Auswahl der drei Softwarepakete beschränkten wir uns auf Windows-Software, die "erschwinglich" ist. Erschwinglich für eine Privatperson dürften 180 DM sein (DeZign) - für eine Firma, die regelmäßig komplexe Datenbanken entwickelt, dürften knapp 5000 DM (DBMAIN, EasyER) ebenfalls erschwinglich sein. Im Idealfall sparen diese Entwicklungswerkzeuge dem Entwickler ja Zeit bei der Konzeption und Realisierung einer Datenbank, was sich in gesunkenen Gesamtkosten bemerkbar macht. Eine Investition in eine solche Software kann sich also schnell bezahlt machen.

Eine komplette Auflistung aller von uns gefundenen Tools zum Thema "ER-Modellierung" findet sich im Abschnitt "Bibliographie" (Kap.5).

Schließlich sei noch erwähnt, daß die betrachteten Tools überwiegend in englischer Sprache verfaßt waren, einige bieten jedoch als Option auch eine deutsche Menüführung.

Auf der nächsten Seite findet sich eine grobe Übersicht über den Funktionsumfang der drei von uns näher betrachteten Programme.

3.2 Übersicht

Programm	Anbieter	Preis (ca.) ⁴	SQL-Generierung	Verschiedene Notationen	OO-DB-Entwicklung	Export	Automat. Generierung von Dokumentation / Reports	Zoom der Ansicht	"Reverse Engineering"
DBMAIN	University of Namur, Belgium	DM 2440-DM 4890	JA	JA	JA (full object-oriented schemas)	Standard SQL, VAX SQL, CODASYL, COBOL	JA	NEIN	SQL, COBOL, IDS, IMS
EasyER / EasyObject	Visible Systems Corporation	DM 4337	JA	JA (u.a. UML)	JA (UML)	MS SQL Server, Sybase, Oracle, Informix, DB2, Watcom, SQL Anywhere, MS Access (u.v.m.)	JA	JA	MS SQL Server, Sybase, Oracle, Informix, DB2, Watcom, SQL Anywhere, MS Access
DeZign	Heraut Automatisering, Holland	DM 180	Ja (Watcom SQL)	NEIN	NEIN	Oracle, IBM DB2, Informix, Ingres, DBASE	JA	NEIN	NEIN

(Fortsetzung)

Programm	Flexible Datentypdefinition	Freie Farb-, Font und Fontgrößenwahl	ODBC-Schnittstelle	Konsistenz-Check	Data-Repository	Versionskontrolle	Datenaustausch mit anderen CASE-Tools	Assistenten ("Wizards")	Studentenversion verfügbar
DBMAIN	JA	JA	NEIN	JA	JA	JA	JA (Mit Voyager 2)	JA	JA
EasyER / EasyObject	NEIN	JA	JA	JA	JA	NEIN	JA (Powerbuilder)	NEIN	NEIN
DeZign	NEIN	JA	NEIN	JA	NEIN	JA	NEIN	NEIN	JA

⁴ Preise umgerechnet in DM

3.3 Das Beispiel Bibliothek

Um den Funktionsumfang der einzelnen Tools überhaupt vergleichen zu können, war es notwendig, ein ER-Modell in allen drei Programmen zu modellieren, da die jeweils zu den Programmen mitgelieferten Beispiele auf die Programme zugeschnitten waren und nicht miteinander vergleichbar waren.

Das von uns verwendete Beispiel ist ein vereinfachtes ER-Modell, das in komplexerer Form als Beispiel dem Programm "DBMAIN" beigefügt ist. Es handelt sich um folgenden Fall:

Ein Ausschnitt der realen Welt: Das Beispiel Bücherei

Beschreibung der darzustellenden Zusammenhänge:

Wir betrachten ein Buch als jegliches Stück Literatur, wissenschaftliche oder technische Schriften. Jedes Buch wird durch eine Nummer eindeutig identifiziert, es besitzt einen Titel, einen Herausgeber, ein Erstveröffentlichungsdatum, Schlagwörter und ein Abstract. Zu jedem Buch gehören die Namen der Autoren.

Für jedes Buch hat die Bücherei eine bestimmte Anzahl von Exemplaren angeschafft. Die einzelnen Exemplare haben eindeutige Seriennummern und zu jedem Exemplar ist bekannt, an welchem Datum es angeschafft wurde, an welcher Stelle der Bücherei es zu finden ist (z.B. in welcher Abteilung, welchem Regal und welcher Reihe), wieviele Bände es umfaßt und welcher Ausleiher sie im Moment ausgeliehen hat (wenn überhaupt). Es können keine einzelnen Bände ausgeliehen werden, es müssen immer alle auf einmal ausgeliehen werden; hinzu kommt, daß jedes Exemplar eines Buches verschieden viele Bände haben kann.

Zu jedem Exemplar wird dessen Zustand dokumentiert, wobei die einzelnen Zustände durch einen einstelligen Code beschrieben werden (z.B. 0=neu, 1=gebraucht,..., 6=schwer beschädigt). Zusätzlich vermerkt man bei Bedarf einen Kommentar zum Zustand des Exemplars.

Der Autor eines Buches hat einen Vor- und einen Nachnamen, einen Geburtstag und eine Herkunft, mit der z.B. die Organisation gemeint ist, bei der / für die diese Publikation entstanden ist. Bei manchen Autoren kann es durchaus vorkommen, daß nur der Name bekannt ist. Es ist natürlich auch denkbar, daß zwei Autoren den gleichen Namen haben.

In der Bücherei sind nur Autoren verzeichnet, von denen auch Publikationen in der Bücherei vorhanden sind.

Eine Kopie kann an einem bestimmten Tag von einem Ausleiher ausgeliehen werden. Jeder Ausleiher wird über eine Personnummer identifiziert. Zu jedem Ausleiher werden Vor- und Nachnamen, Adresse (Straße, PLZ, Ort) sowie die Telefonnummer gespeichert.

Die zurückkommenden Kopien werden in einem Korb gesammelt und am Ende eines Tages dann an ihren Platz zurückgebracht. So sind die Bücher dann am nächsten Tag wieder verfügbar. Wird eine ausgeliehene Kopie zurückgebracht, so vermerkt der Angestellte folgende Informationen: Ausleihdatum, aktuelles Datum und den Ausleiher.

Jede Ausleihe erfolgt anhand eines Ausleihvorgangs, welcher durch eine interne Nummer (Leih-ID) identifiziert wird.

3.4 DBMAIN

3.4.1 Allgemeines

DBMAIN (DataBase MAINTenance) ist ein universelles CASE-Tool zur Datenbank-Anwendungsentwicklung, das unter der Federführung des Instituts für Informatik an der Universität Namur in Belgien entwickelt wurde. Das Projekt wurde im September 1993 ins Leben gerufen und wurde von 27 Organisationen unterstützt (u.a. von DIGITAL, IBM, ORIGIN, dem "Institut National de Criminalistique", der Stadt Namur, der Stadt Winterthur und der Europäischen Union⁵). Einige Ergebnisse des Projekts wurden der Öffentlichkeit und nicht-kommerziellen Unternehmen frei zugänglich gemacht, um Feedback von den Anwendern zu bekommen. Die Schulversion des DBMAIN CASE-Tools, einige technische Dokumentationen und Forschungsdokumente sind für Universitäten und Forschungseinrichtungen kostenlos erhältlich.

3.4.2 Historie

DBMAIN wurde in Hinblick auf Datenbankdesign, "Reverse-Engineering", "Re-Engineering", DB-Integration, -evolution und -wartung entwickelt. Das gesamte DBMAIN-Projekt besteht dabei aus drei Teilen: Forschung, Entwicklung und Technologietransfer. Die Forschung stand dabei am Anfang und führte zur Entwicklung eines ersten Prototyps der DBMAIN-Applikation im September 1995. Das CASE-Tool überführte dabei viele Erkenntnisse der Forschung in handfesten Code und half somit viele Probleme der Entwickler zu lösen.

Seit Version 1 sind nun vier Jahre vergangen, und DBMAIN ist nun bei Version 4.1 angelangt.

3.4.3 Merkmale von DBMAIN Version 4.1

- Eingabe und Management von Multischema-Projekten
- Eingabe und Management von konzeptionellen, logischen und physischen Datenbankschemas
- Eingabe und Management von "processing units"
- Schema-Ansicht in 6 Formaten: 2 graphische Formate, 4 Hypertext-Formate
- U.a. relationale, COBOL- oder CODASYL-Schemas
- volle Unterstützung objektorientierter Schemas
- Ableitung logischer Schemas aus konzeptionellen Schemas (forward engineering)
- Ableitung konzeptioneller Schemas aus logischen Schemas (reverse engineering)
- Transformation von Schemas durch ein Toolset von mehr als 25 Operatoren (zur Normalisierung, Optimierung, Restrukturierung, Implementierung, für das "reverse engineering", etc.)
- Raffinierter Transformations-Assistent zur Problemlösung
- Automatisierung der Anwendung von Transformationen auf ein Schema durch einen globalen Transformations-Assistenten
- Integration von Projekten, Schemas oder Entity-Typen

⁵ Projekt "DB-MAIN/Objectif"

- Analyse von Schemas und Suche nach komplexen Strukturmustern
- Heuristische Suchmethoden um Fremdschlüssel zu finden
- Entwickeln und Testen von Skripten und strukturellen Untermodellen
- Generierung von ausführbarem Programmcode für mehrere DBMS (SQL, IDS/II, CODASYL, COBOL)
- Generierung von Reports
 - "Repository"-basierte integrierte 4GL-Programmiersprache (Voyager 2) zur benutzerdefinierten Report-Generierung oder Code-Generierung)
 - Paraphrasor (Report-Generator in natürlicher Sprache)
- Extraktion physischer Schemas aus SQL, COBOL, IMS, RPG und IDS/II DDL Quell-Code
- Integrierter Quell-Code Editor mit einer ausgefeilten interaktiven Mustererkennungs-Engine
- Hochentwickelte Programmanalyse
- Update eines logischen Schemas durch den Import von Quell-Code
- Protokollierung der Benutzeraktionen und selektive Wiedergabe der Aktionen
- Ansichten mit automatischen Update-Hinweis
- Import und Export externer Spezifikationen
- Entwicklungsprozeß - Kontrolle

3.4.4 Eigenschaften des Programms

- 32-bit Architektur
- geschrieben in C++
- Unterstützung von Drag&Drop und frei schwebenden Paletten
- flexible Farbwahl in der Darstellung von Objekten

3.4.5 Systemvoraussetzungen

- Pentium PC
- 32 MB RAM
- Bildschirmauflösung 800x600 Pixel
- OS: Windows 9x, Windows NT

3.4.6 Installation

Von der Homepage des Institutes für Informatik der Universität Namur (<http://www.info.fundp.ac.be/~dbm>) läßt sich das (gepackt) etwa 4 MB große Paket herunterladen. Es enthält das Programm selbst, ein ausführliches Handbuch, ein Tutorial (207 Seiten) und eine Dokumentation der integrierten Programmiersprache "Voyager 2".

Nach der problemlosen Installation (Verzeichnis anlegen, Programm entpacken) belegt das Programm 7,6 MB auf der Festplatte, ist also im Vergleich mit anderen Paketen angenehm klein und kompakt.

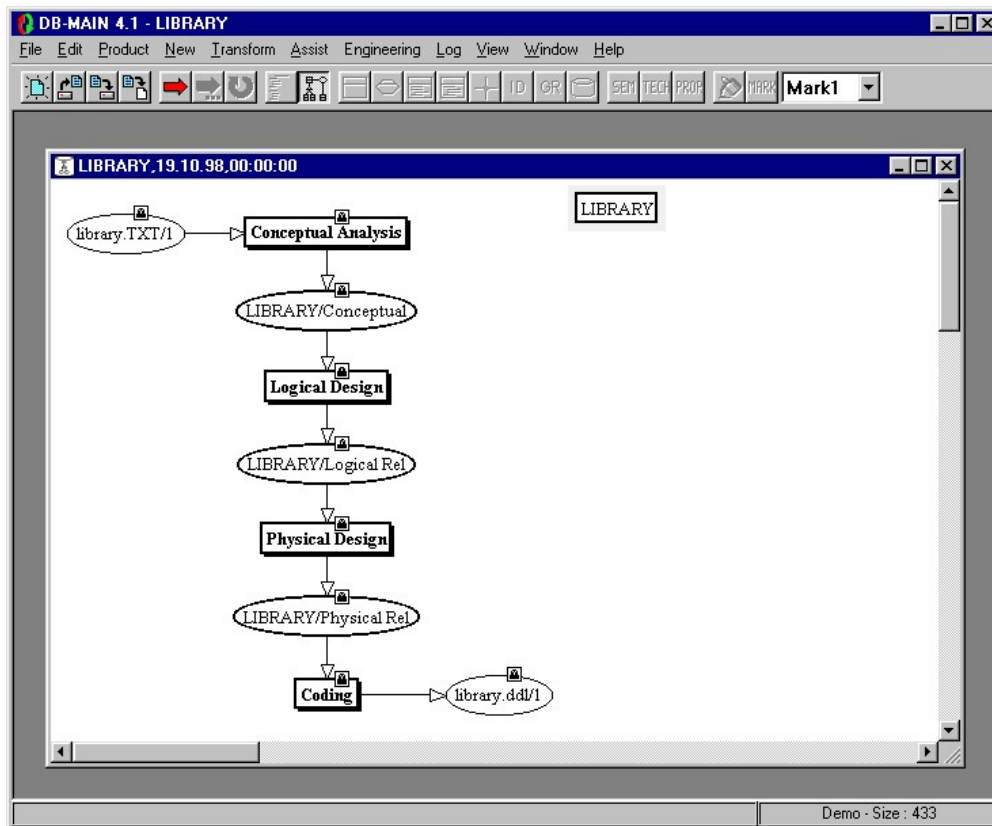
3.4.7 Einschränkungen der Demo-Version

Die Schul- bzw. Demoversion besitzt keine funktionellen oder zeitlichen Einschränkungen, ist aber auf Projekte mit 500 Objekten (Entity-Typen, Relationstypen, Attributen) limitiert.

Wenn das Limit überschritten wird, nimmt das System keine Eingaben mehr an, man sollte also die Anzeige der Projektgröße im Statusfeld am unteren Rand des Bildschirms im Auge behalten.

3.4.8 Gesamtansicht

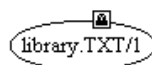
Nach dem Start des Programmes und dem Laden eines Projektes wird man mit folgender Ansicht begrüßt:



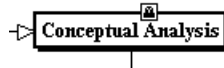
Das Projekt selbst wird hierbei durch einen rechteckigen Kasten dargestellt:



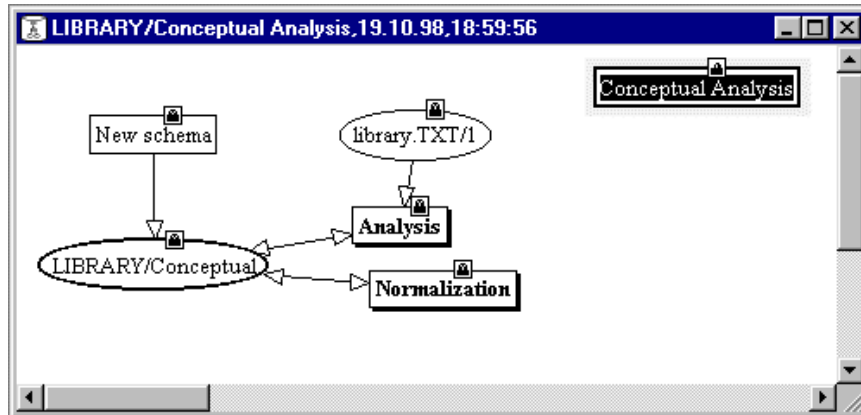
Ein Doppel-Klick auf den Kasten bringt eine Dialogbox mit den Eigenschaften des Projektes hervor.



Hinter diesem Icon verbirgt sich der konzeptionelle Text, eine Textdatei im ASCII-Format, die das Szenario beschreibt.



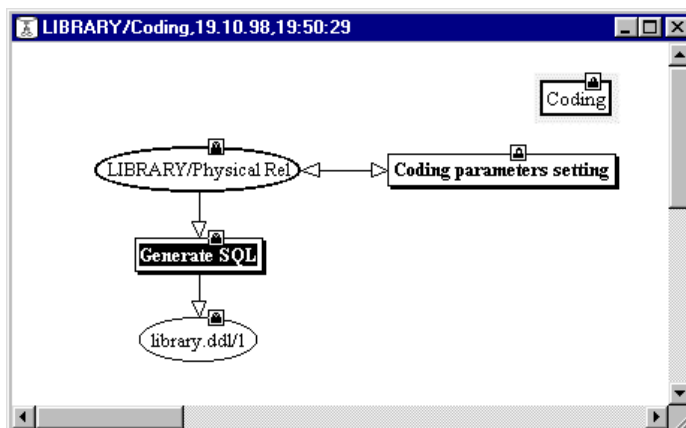
Hinter den rechteckigen Icons verbirgt sich immer eine Analyse eines Sachverhalts, hier die konzeptionelle Analyse.
Bei Click auf dieses Icon erscheint folgendes Schema:



Analog hierzu erfolgt eine weitere analytische Detailansicht, wenn der Benutzer "Logical Design" oder "Physical Design" anklickt.



Unter "Coding" findet sich schließlich das Schema für die Code-Erzeugung:

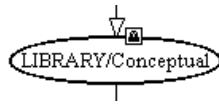


Die Datei "library.ddl" enthält dann den erzeugten SQL-Code.

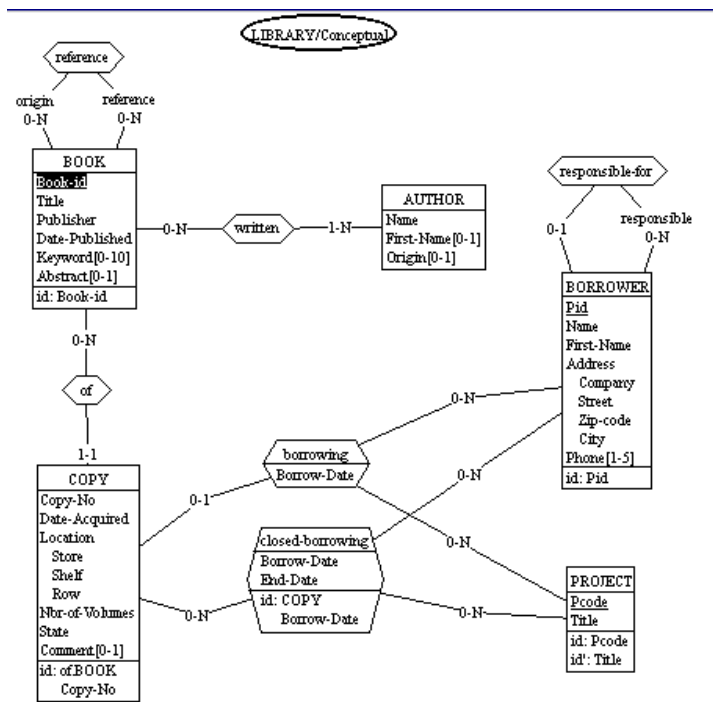
Hinter den Icons "LIBRARY/Conceptual", "LIBRARY/Logical Rel" und "LIBRARY/Conceptual" verbirgt sich das "Repository" und die eigentlichen Modellierungen.

3.4.9 Konzeptioneller Entwurf

Beispielhaft sei hier nun der konzeptionelle Entwurf näher betrachtet.
Wir erinnern uns:



Dieses Icon steht für die Repository, d.h. die Programminterne "Datenbank" für das eigentliche konzeptionelle Schema:



Hier können nun die Attribute, Entites und Relationen erzeugt und verändert werden.

3.4.10 Fazit

DBMAIN präsentiert sich optisch schlicht, ist dennoch aber ein sehr leistungsfähiges Paket zur Konzeption und Analyse, als auch zum Design, Generierung und zur Wartung von Datenbankanwendungen.

Dieser umfassende Ansatz macht das Paket sehr attraktiv und sprichwörtlich preiswert, da auf Zusatzprogramme von Fremdherstellern verzichtet werden kann.

Auch der Support ist sehr gut, auf Anfragen per Email wurde uns aus Belgien immer am gleichen Tag noch geantwortet.

Durch die Tatsache, daß hinter diesem Projekt eine große Anzahl an finanzstarken Partnern (u.a. IBM und DIGITAL) und nicht zuletzt die EU stehen, ist eine Weiterentwicklung des Programms sowie eine schnelle und effiziente Beseitigung von eventuell auftretenden Fehlern gewährleistet (obwohl wir bei unseren Tests keine finden konnten). Auf der Entwicklerseite steht die Universität Namur mit Prof. Jean-Luc Hainaut und seinem Team von 14 Programmierern hinter dem Projekt, was für eine stetige Weiterentwicklung der Software zumindest bis zum offiziellen Ende des Projektes nach 8 Jahren im Jahr 2001 sorgen dürfte.

Das Programm machte insgesamt einen sehr stabilen und ausgereiften Eindruck und kann durchaus empfohlen werden.

3.5 DeZign

3.5.1 Allgemeines

DeZign ist ein Datenmodellierungstool, das die Erstellung von Datenbanken für Desktop sowie für Client/Server Anwendungen erleichtern soll. Hierbei bedient sich das Programm der Darstellung anhand von ER-Modellen bzw. ER-Diagrammen.

3.5.2 Merkmale von DeZign 1.11

- Graphischer Entity-Relationship-Editor
- Automatische Generierung von SQL-Schemas für führende RDBMS (Oracle, IBM DB2, Informix, Ingres, etc.)
- Unterstützung von benutzerdefinierten Domains-Typen
- Integriertes System zur Versionskontrolle
- Generierung von Reports

3.5.3 Funktionalität

DeZign bietet den Nutzern die Möglichkeit, Daten einfach und schnell zu modellieren, durch die selbsterklärenden Funktionalitäten (add/delete- entity / add/delete relationship).

Der Nutzer hat außerdem die Wahl, das Datenmodell unterschiedlich ausgeben zu lassen (Entities and Attributes/ Entities and Primary Key/ Entities and Attributes, Primary Keys).

Es ist bestimmbar, in welchen vorgegebenen unterschiedlichen Farben man sein Datenmodell ausgegeben haben möchte.

Zudem gibt es die Möglichkeit, für das erstellte Datenmodell einen „Report“ ausgeben zu lassen. In diesem werden alle Entitäten mit Ihren Attributen und den dazugehörigen Feldtypen aufgelistet.

Direkt aus dem Programm heraus können SQL-Statements für folgende Datenbanken erstellt werden:

Oracle, Interbase, Informix, Sybase, IBM DB2, Ingres, Watcom SQL, ANSI Level 2, dBase und Paradox.

Eine besondere Funktion dieses Tools ist das eingebaute Versionskontrollsystem, welches es ermöglicht, jederzeit eine frühere Version der gerade bearbeiteten Datenbank wiederherzustellen.

Den Entwicklern des Tools kam es nach eigenen Angaben darauf an, die Bedienung der Software möglichst einfach zu halten und dabei trotzdem ein professionelles Tool bereitzustellen.

3.5.4 Bewertung

Zur Erstellung von Datenmodellen bietet DeZign, wie auch von den Herstellern beabsichtigt, einfache und überschaubare Bedienungsmöglichkeiten.

Die Auswahl an unterschiedlichen Ausgabefarben und –Formen bietet dem Nutzer die Möglichkeit, den Überblick zu behalten.

Abgesehen von Kleinigkeiten (Eine Zoomfunktion der Ansicht ist nicht vorhanden) ist dieses Tool im Hinblick auf Kosten und Nutzen geeignet, um einfache Zusammenhänge darzustellen.

3.5.5 Gesamtansicht

Abbildung 1 : DeZign Datenmodell

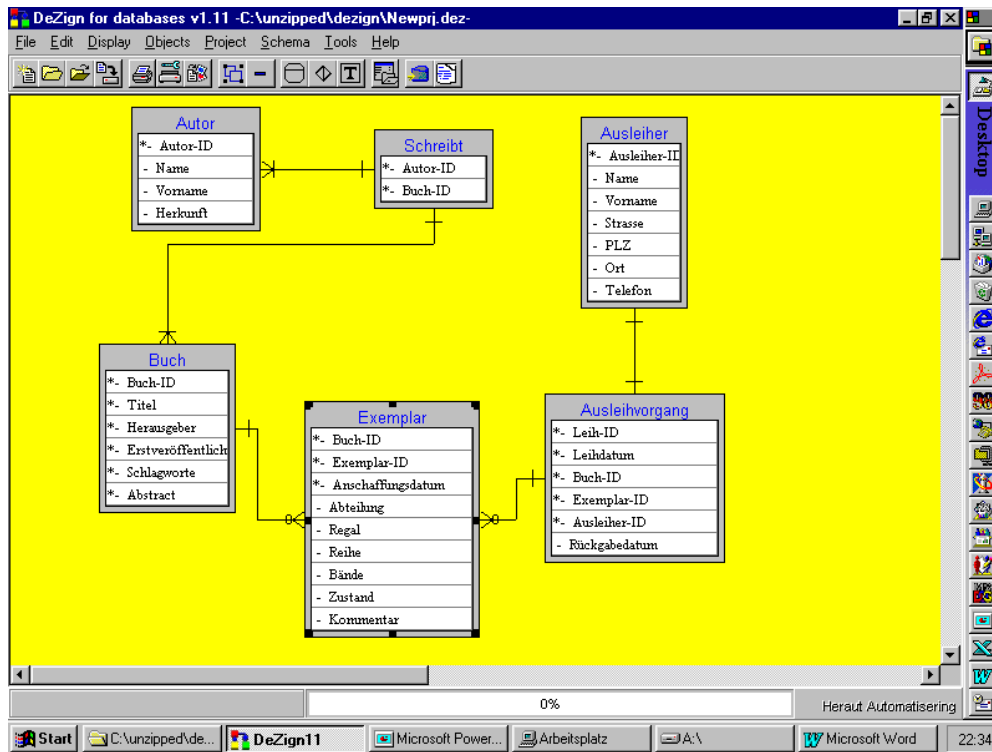


Abbildung 2 : DeZign Report

The screenshot shows the DeZign Report window, displaying a detailed table of attributes for the entities: Autor, Schreibt, and Buch. The table columns are: Attributname, Label, Domain, Fieldtype, Fieldsize, Decimalplaces, Unique, ID, Required, Description, Defaultvalue, and Checkconstrau.

Attributname	Label	Domain	Fieldtype	Fieldsize	Decimalplaces	Unique	ID	Required	Description	Defaultvalue	Checkconstrau
Autor											
Autor-ID			NUMERIC	0	0	no	yes	no			
Name			CHAR	0	0	no	no	no			
Vorname			CHAR	20	0	no	no	no			
Herkunft			CHAR	25	0	no	no	no			
Autor-ID			NUMERIC	0	0	no	yes	no			
Buch-ID			INTEGER	0	0	no	yes	no			
Schreibt											
Autor-ID			NUMERIC	0	0	no	yes	no			
Buch-ID			INTEGER	0	0	no	yes	no			
Buch											
Buch-ID			INTEGER	0	0	no	yes	no			
Titel			CHAR	0	0	no	yes	no			
Herausgeber			CHAR	0	0	no	yes	no			
Erstveröffentlichung			DATE	0	0	no	yes	no			
Schlagworte			CHAR	0	0	no	yes	no			
Abstract			TEXT	0	0	no	yes	no			

3.5.6 Systemanforderungen

"DeZign for databases 1.11" ist nur als 32-Bit Anwendung verfügbar und kann somit auf Windows 9x, Windows NT 3.51 oder Windows NT 4.0 eingesetzt werden. Das Programm ist bereits ab einem 486er Prozessor mit 4 MB RAM und 4MB verfügbarem Festplattenplatz lauffähig.

3.5.7 Einschränkungen der Demo-Version

Das Demo ist in seinem Funktionsumfang leicht beschränkt: Die Anzahl der Attribute einer Entity sind limitiert.

Dieses Tool wurde vor allem wegen seinem relativ günstigen Preis von \$97,50 in die engere Auswahl genommen. Neben dieser "Vollversion" gibt es noch eine etwas günstigere "Lite"-Version (\$67,50), welche dann aber nur die Datenbanken "dBase" und "Paradox" unterstützt. Falls man die Software gerne auf anderem Wege als per Email bekommen möchte, so muß man zusätzlich \$9 bezahlen.

3.6 EasyER/EasyObject

3.6.1 Einleitung

Die Firma Visible Systems Corporation ist laut eigenen Angaben einer der führenden Hersteller von Enterprise Software und Engineering Tools für komplexe Software. Das Flaggschiff dieses Unternehmens ist die Software "Visible Analyst", ein integriertes CASE-Tool für die synchrone Modellierung von Prozessen, Daten und Objekten.

Hier geht es jedoch um die Software "EasyER/EasyObject 2.0", die sich im Vergleich zu "Visible Analyst" auf den Bereich des Datenbankentwurfs konzentriert.

Von der Homepage der "Visible Systems Corporation" (<http://www.visible.com>) läßt sich das ca. 22 MB große Programm herunterladen. Diese Datei stellt sowohl eine 30-Tage-Evaluierungsversion zu Demonstrationszwecken dar, ist gleichzeitig aber auch bereits die Vollversion, in der alle Funktionen verfügbar sind. Diese können aber erst durch die Seriennummer freigeschaltet werden, die man erhält, wenn man sich für das Programm registrieren läßt und es bezahlt hat.

Im Demonstrationsmodus ist die Software bis auf folgende Einschränkungen voll funktionsfähig:

- Beim Ausdruck von Charts wird immer ein Demonstrationshinweis mit ausgedruckt, welcher auch bei der Arbeit mit dem Tool ständig zu sehen ist.
- Die Anzahl der Charts ist auf vier beschränkt.
- Die maximale Anzahl von Entities in einem Chart beträgt 20.
- Die Testzeit beträgt 30 Tage, danach kann das Programm nicht mehr benutzt werden, es sei denn man erwirbt die Vollversion.

3.6.2 Beschreibung

"EasyER/EasyObject" ist ein Data-Dictionary-basiertes Datenmodellierungs- und Datenbankdesigntool. Es stützt sich dabei auf Entity-Relationship-Diagramme und unterstützt viele verschiedene Notationen. Der Name des Produktes läßt bereits erkennen, daß neben den herkömmlichen Modellierungsnotationen auch objektorientierte Notationen eingesetzt werden. Gewundert hat uns allerdings, daß die uns aus den Vorlesungen des Fachbereichs bekannten Notationen wie die von Chen und Denert nicht im Auswahlmenü auftauchen.

Für jedes Projekt in EasyER/EasyObject wird ein Data Dictionary (DATADICT.DB) im jeweiligen Projektordner angelegt. Um mit dem Data Dictionary zu arbeiten, wird die "SQL Anywhere 5.5 Database" benötigt, die zusammen mit dem Programm installiert werden kann. Zusätzlich muß eine ODBC Quelle ausgewählt werden (dies ist Teil der Installation). Die "SQL Anywhere 5.5 Database" wird automatisch und ohne eine Änderungsmöglichkeit in C:\SQLANY50\WIN32 installiert und der Pfad in der AUTOEXEC.BAT entsprechend gesetzt.

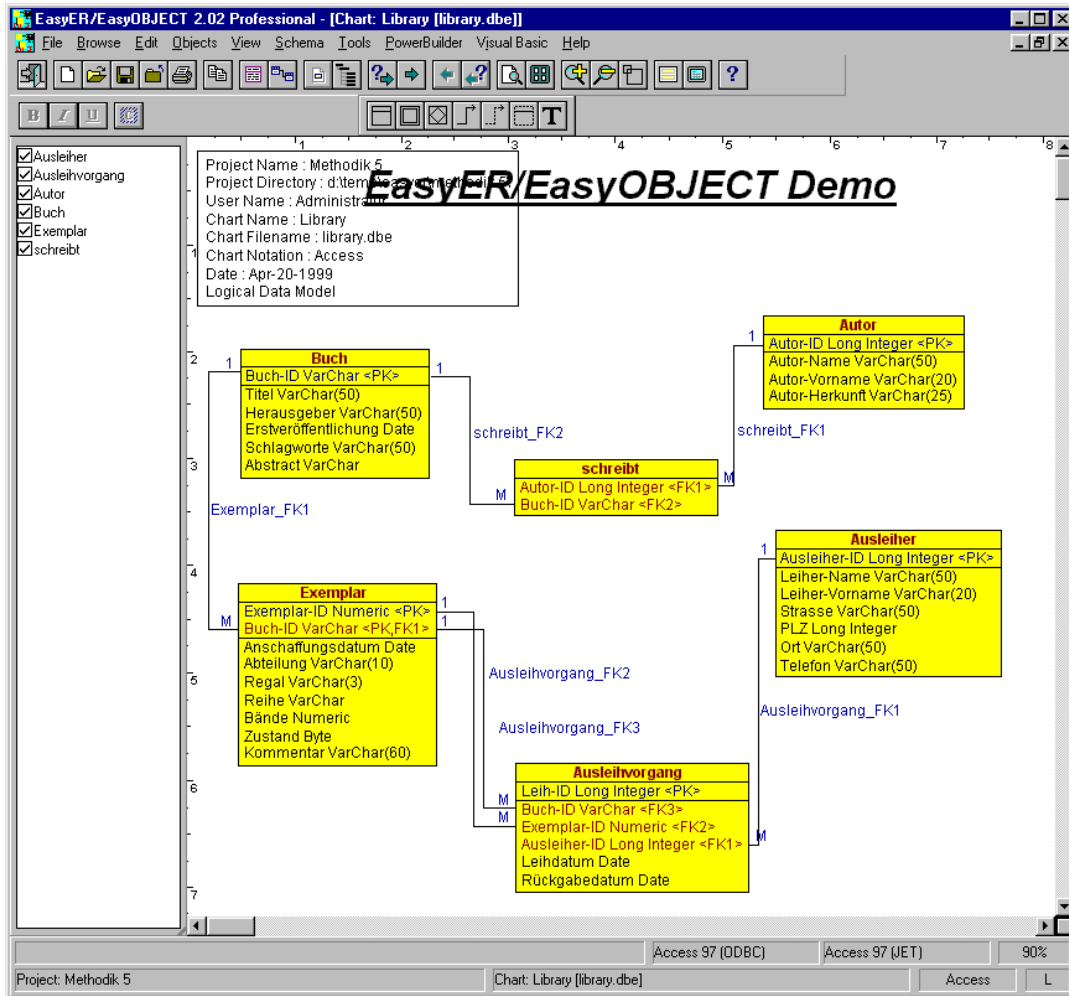


Abbildung 1: Arbeitsoberfläche von EasyER

Das Programm besitzt folgende Merkmale und Besonderheiten:

- Es bietet eine große und gut skalierbare Chart-Fläche, die es erlaubt hunderte von Entities darzustellen.
- Die gesamte Projektinformation wird in einem Data-Dictionary gespeichert.
- EasyER/EasyObject bietet Unterstützung für ein großes Spektrum von Desktop und Client/Server Datenbanken (ODBC).
- Es generiert Datenbanken mit JET (nur Microsoft Access), ODBC oder SQL DDL-Skripten.
- Für objektorientierte Notationen werden Klassen angeboten.
- Es verfügt über Modi zur logischen wie zur physischen Modellierung.
- Das Programm enthält bereits viele vorgefertigte Reports.
- Es besteht die Möglichkeit, Datenmodelle anhand von "Reverse Database Engineering" zu generieren.
- "EasyER/EasyObject" bietet erweiterte Möglichkeiten für Powersoft's "PowerBuilder".
- "EasyER/EasyObject" arbeitet direkt mit "Visual Basic 4/5" zusammen.

Folgende Datenbanksysteme werden unterstützt:

- Access 1.x, 2.0, 7.0, 8.0
 - Btrieve
 - dBASE 3, 4, 5, 5.5
 - FoxPro 2.x, 3.0
 - Visual FoxPro 3.0, 5.0
 - IBM DB2, DB2 v2.11 for Windows NT/95
 - Ingres
 - Informix 5, 6, 7.x
 - InterBase 4.x
 - ORACLE 6, 7, 8
 - Paradox 3, 4, 5, 7
 - Progress 6, 7, 8
 - Raima Velocis
 - R:BASE 5.x
 - SQLBase 5.x, 6.0
 - SQL Server 4.2, 6.x
 - Sybase System 10/11
 - Sybase SQL Anywhere 5.x
 - Watcom SQL 3.x, 4.0
 - XDB 3,4
- ...und mehr

3.6.3 Installation

Systemanforderungen:

- Windows NT 3.51, 4.0 oder Windows 9x
- Pentium 100MHz +
- 16 MB Arbeitsspeicher
- 20-40 MB Festplattenplatz, je nach den ausgewählten Optionen

Bei der Installation verhält sich das Programm wie Standard-Windows-anwendungen, so daß der "normale" PC-Benutzer eigentlich keine Probleme damit haben sollte. Dem Programm liegt eine einführende PDF-Datei bei, die die Installation sehr ausführlich beschreibt und einen Einstieg in das Programm bietet. Das gesamte Programm inklusive der Dokumentation ist in Englisch gehalten, andere Sprachen werden nicht unterstützt.

3.6.4 Das Erscheinungsbild

Hat man vor dem ersten Kontakt mit "EasyER/EasyObject" bereits andere Modellierungstools gesehen, so erscheint dieses Programm auf den ersten Blick recht sympathisch, da es sich mit seiner Oberfläche sehr "Windows"-konform zeigt. Auch die Verwendung von mehr als zwei Farben vermittelt einen moderneren Eindruck als manche anderen Tools. Als sehr praktisch erweist

sich die Möglichkeit die Arbeitsfläche (Chart) von 10% bis 200% (jeweils in 10% Schritten) zu zoomen; dies ermöglicht zum einen eine Anpassung an die jeweils vorhandene Hardware bzw. die Einstellungen (Monitor, Grafikkarte und Auflösung) als auch die bequeme Arbeit mit besonders großen und umfangreichen Charts.

3.6.5 Die Benutzerverwaltung

Gleich nach der Installation macht man auch schon mit der Nutzerverwaltung von "EasyER/EasyObject" Bekanntschaft: Um eines der Beispielsprojekte zu öffnen muß man sich erst anmelden – in der beiliegenden Hilfedatei und der Einführung findet man nach kurzer Suche dann die Möglichkeiten, mit denen man sich anmelden kann: "administrator/password" oder "user/password" sind die Zauberwörter die man benötigt.

Als Administrator hat man dann die Möglichkeit, neue Nutzerprofile anzulegen oder vorhandene zu ändern oder zu löschen. Jedes mal wenn man ein Projekt öffnet, muß man sich für dieses anmelden.

3.6.6 Die Bedienung

Gibt man sich der Illusion hin, man könnte mit ein wenig ausprobieren in kurzer Zeit die Grundfunktionen des Programms beherrschen, so wird man schnell eines Besseren belehrt, zumindest wenn beim Benutzer keine ausgeprägten Vorkenntnisse im Bereich Datenbankmodellierung vorhanden sind. Als am Praktischsten stellt sich die Einführung mit dem mitgelieferten Tutorial, den Beispielen und der Online-Hilfe heraus. Hier bekommt man in relativ kurzer Zeit die wesentlichen Begriffe und Arbeitsweisen erklärt.

3.6.7 Notationen

Prinzipiell gibt es bei EasyER/EasyObject zwei Arten von Notationen: Die herkömmlicherweise für Entity-Relationship-Diagramme benutzten Notationen, deren Elemente auch Entities und Relationen heißen und die objektorientierten Notationen, bei denen man dann von Klassen, Objekten und Operationen spricht. Je nachdem, welche Notation man aktuell für sein Projekt gewählt hat variieren auch die verfügbaren Menüeinträge. Folgende Notationen werden unterstützt:

- Martin
- Bachman
- Shlaer-Mellor
- IDEF1X
- Access
- Referencing Arrowhead
- Referenced Arrowhead

Objektorientierte Notationen:

- Coad-Yourdon
- Rumbaugh OMT
- Rational's "Unified" Modeling Language (UML)

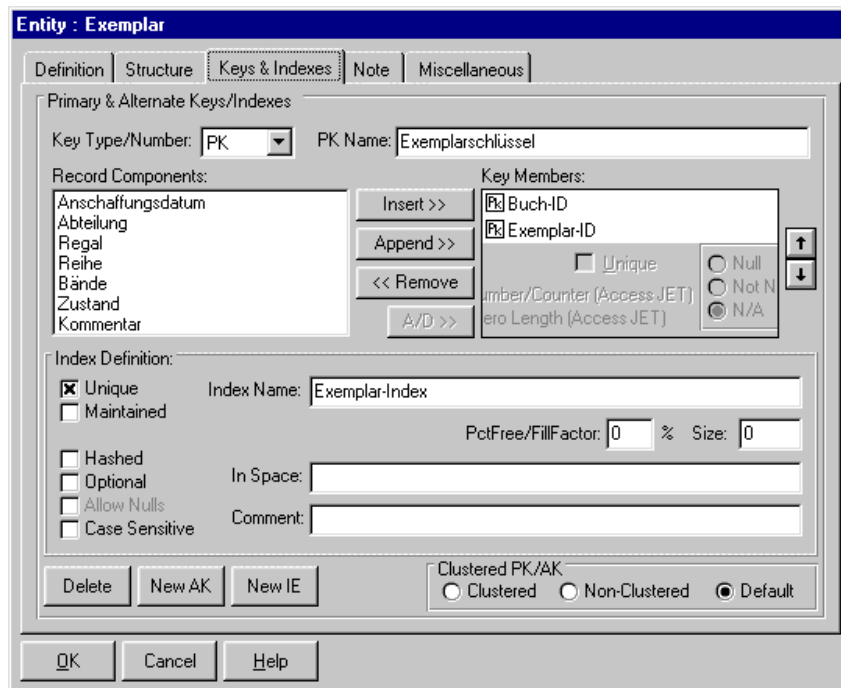


Abbildung 2: Schlüsselvergabe

Etwas verwirrend ist bei EasyER/EasyObject, daß man vergeblich nach einer Trennung des konzeptionellen vom logischen Datenmodell sucht, denn grundsätzlich unterscheiden sich diese:

Bei der ER-Modellierung wird der darzustellende Ausschnitt der Welt mit Entities und Relationen graphisch dargestellt. Dieses Modell geht normalerweise dem relationalen Datenbankentwurf voraus, welcher die im ER-Modell dargestellte Welt dann in relationale Tabellen umsetzt. Üblicherweise macht man erst an dieser Stelle von Fremdschlüsseln und ergänzenden Tabellen zur Darstellung der Beziehungen zwischen den Entities Gebrauch.

Bei EasyER/EasyObject gibt es keine Darstellungsform, die mit "konzeptionell" bezeichnet wäre oder ausschließlich einer solchen entspräche, man kann nur zwischen der logischen und physikalischen Darstellung wählen. Diese logische Ansicht vereint nun das ER-Modell mit dem relationalen Entwurf: Man definiert zwar Entities und die zugehörigen Relationen, die dann auch wie üblich durch Rechtecke und Verbindungslinien dargestellt werden, muß allerdings zusätzlich die Entities schon wie Tabellen behandeln und somit jede Verknüpfung auch mit Fremdschlüsseln modellieren. Dadurch, daß dem Benutzer dieser Schritt vom ER-Modell zum relationalen Datenbankentwurf nicht vom Programm abgenommen wird, ergeben sich weitreichende Konsequenzen. Während man in einem herkömmlichen ER-Modell eine viele-zu-viele Beziehung mit zwei Entities und einer Beziehung zeichnen kann, benötigt man bei EasyER / EasyObject zu dieser Modellierung drei Tabellen bzw. Entities (und zwei 1:n

Beziehungen) um die n:m Beziehung mitsamt den Fremdschlüsseln darzustellen. Die aus früheren Vorlesungen bekannten Sechsecke zur Visualisierung der Beziehungen mit ihren Attributen wird man in EasyER / EasyObject also nicht finden.

Je nachdem, welchen Datenbanktyp man für die Generierung des endgültigen Datenbankcodes ausgewählt hat, unterscheidet sich das physikalische Modell vom logischen. Während man bei der Konzeption des logischen Modells immer auf die gleichen einheitlichen Feldtypen zugreifen kann, werden diese in der physikalischen Ansicht an die Zieldatenbank angepaßt (so wird beispielsweise aus dem oft benutzten Feldtyp "varChar" bei einer Ausgabe für das Datenbanksystem Access der Feldtyp "Text" generiert). Die beiden Modelltypen können farblich und von der Schrifteinstellung her getrennt voneinander eingestellt werden.

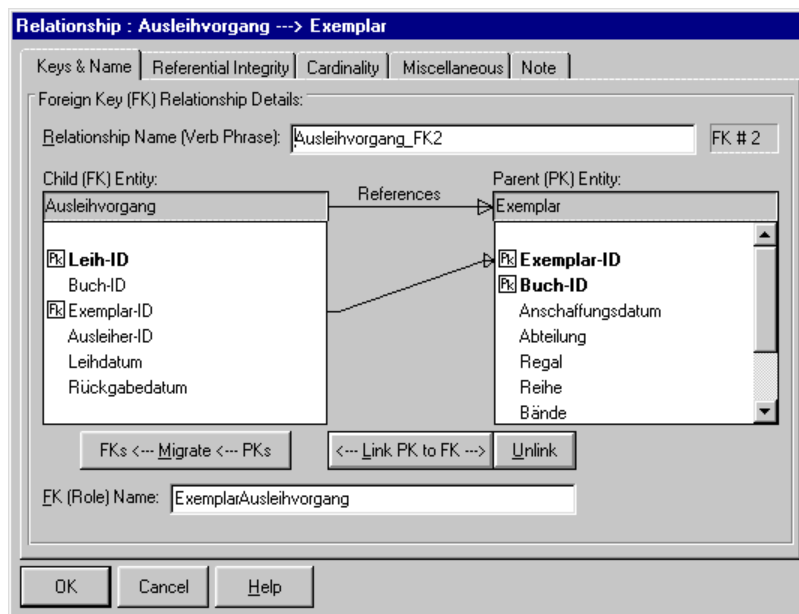


Abbildung 3: Schlüsselvergabe

Optisch erfüllt das Programm fast alle Wünsche: Allen dargestellten Elementen können Farben und Schriften nach Wahl zugeteilt werden, die gute Skalierbarkeit wurde bereits weiter oben erwähnt, sie setzt sich auch bei den möglichen Einstellungen zum Druck fort, denn auch hier kann man die Größe gut justieren, zwischen 20% und 200% kann man die optimale Druckgröße wählen. Allgemein hat man sehr viele individuelle Einstellmöglichkeiten, so kann man z.B. auch bestimmen, ob die Schlüsselfelder immer sichtbar sind oder nur auf Wunsch angezeigt werden und wie diese darzustellen sind etc.. Allgemein kann man sagen, daß EasyER/EasyObject in Bezug auf die Präsentation sehr viel bietet.

Es gibt eine Fülle von Möglichkeiten, die referentielle Integrität einzustellen und zu fast allen editierbaren Elementen lassen sich Kommentartexte eingeben, die die Orientierung im Modell erleichtern.

Neben der Möglichkeit ein Datenmodell komplett per Hand zu erstellen bietet EasyER/EasyObject das "reverse engineering" von einer bereits existierenden Datenbank an. Man erstellt also quasi rückwärts aus einer bereits bestehenden Datenbank (z.B. als SQL-, DDL- Datei) das gewünschte Modell, welches man daraufhin ganz normal mit dem Tool weiterverarbeiten kann. Diese Methode über die SQL DDL kann bei folgenden Datenbanken angewandt werden: Microsoft SQL Server 4.2, 5.0, 6.5, Sybase System 10/11, Oracle 7.2, 7.3, Informix 7.2, 7.3, DB2 2.1, Watcom SQL 4.0, SQL Anywhere 5.0, 5.5, SQLBase 6.0, Access 95/97 SQL und dBASE IV SQL.

Am Ende der Modellierung steht die Generierung eines Datenbankschemas, welches dann in die jeweilige Datenbank importiert werden kann. Hierfür sind die eingangs aufgeführten Datenbanken geeignet und auswählbar.

3.6.8 Kritikpunkte

Mag das Programm auch sehr viele Funktionen anbieten, so ist die Handhabung doch teilweise recht umständlich. Es ist uns beispielsweise nicht gelungen, eine einmal eingegebene Bezeichnung für eine Klasse nachträglich zu verändern - der Menüpunkt "rename" ist zwar vorgesehen, aber leider nicht auswählbar. Die einzige Möglichkeit, das Modell auf den korrekten Stand zu bringen ist demnach, die komplette Klasse sowie deren Beziehungen neu einzugeben; hatte man sich damit bereits vorher große Mühe gegeben, so ist der Ärger um so größer. Auch das Umbenennen von Feldnamen ist verwirrend: Der Button zum Umbenennen ist grau hinterlegt, aber über das Kontextmenü (rechte Maustaste) läßt sich das Feld dann doch mühelos bearbeiten. Weiterhin negativ aufgefallen ist uns, daß das Programm manchmal bereits vorgenommene Einstellungen wieder vergißt. Ganz extrem macht sich dies beim Layout bemerkbar: Nachdem man die Position eines Objektes oder einer Entity verändert hat, werden alle bereits manuell an der Anordnung der verbundenen Relationen vorgenommenen Änderungen wieder in ihren Ursprungszustand zurückversetzt. Aus diesem Grund sollte man mit der Ausrichtung der Elemente bis zum Schluß der Modellierung warten. Auch sehr sonderbar ist die Angewohnheit des Programms, nach manchen Änderungen plötzlich zur nächsten aktiven Anwendung zu wechseln, was bei längerer Arbeit mit dem Produkt doch negativ auffällt.

3.6.9 Fazit

"EasyER / EasyObject" hat seine Stärken und seine Schwächen. Einerseits überzeugt es durch den immensen Funktionsumfang und durch die optisch übersichtliche Gestaltung, andererseits macht die Arbeit mit dem Programm nicht immer Spaß.

Der Hersteller sollte in der nächsten Version unbedingt die größeren und kleineren "Bugs" aus dem Programm entfernen, dann könnte man es fast uneingeschränkt zur Datenbankmodellierung empfehlen.

4 Ausblick

Momentan ist eine Phase des Umbruchs im Bereich der Datenbankentwicklung festzustellen. Immer mehr Tools unterstützen OO-Ansätze, d.h. sowohl objektorientierte Datenbanken als auch die dazugehörige objektorientierte Datenbankentwicklung.

4.1 Objektorientierte Entwicklung

Die Vorteile der objektorientierten Entwicklung sind:

- Der Systementwurf ist einfacher, weil die Objekte direkt im Sinne des realen Problembereiches definiert werden können, und nicht im Sinne abstrakter Programmfunktionen
- Systeme sind einfacher zu verändern, weil Wechselbeziehungen zwischen Objekten durch die lokalisierte Verwendung von Daten und Prozeduren minimiert werden
- Die Konstruktion von Systemen wird vereinfacht, weil neue Systeme durch Wiederverwenden und Erweitern vorhandener Objekte erzeugt werden können

Objektorientierte Datenbank- Entwicklungstools müssen dabei eine große Anzahl von objektorientierten Methoden wie z.B. die "Object Modeling Technique" (OMT), Booch, "Jacobsen Use Cases", die "Unified Modeling Language" (UML, siehe nächster Abschnitt) und "Fusion" unterstützen. Die einzelnen CASE-Tools unterscheiden sich hauptsächlich in der Anzahl und Art der unterstützten Notationsformen, der Programmiersprachen, der Implementierung von Repository-Mechanismen und der Integration von Software zur Versionskontrolle. Weiterführende Informationen bietet hier der Artikel von Fred Heibel über objektorientierte CASE-Tools⁶.

4.2 UML

Im Bereich der Datenbankentwicklung hat wohl kein Akronym so schnell die Runde gemacht wie "UML". Auch in der Feature-Liste der von uns untersuchten Programme tauchten diese drei Buchstaben immer wieder auf.

Da die UML sich auf objektorientierte Datenbankentwicklung konzentriert und im Aufbau sehr komplex ist, würde hier eine nähere Betrachtung den Rahmen dieser Ausarbeitung sprengen.

Daher wollen wir hier einen Auszug aus der "Website Methodik"⁷ zitieren, in dem die Grundzüge der UML erläutert werden:

"Die Unified Modelling Language (UML) ist ein internationaler Standard und wurde enthusiastisch von der Softwareindustrie aufgenommen als einheitliche Darstellungsform für OO-Analyse und - Design. UML wird mittlerweile zunehmend auch außerhalb der Softwareentwicklung als allgemeine Modellierungssprache verwendet. UML wurde entwickelt aus drei der erfolgreichsten OO-Methoden (OMT, Booch und Objectory) und umfaßt alle für die OO-Entwicklung benötigten

⁶ Fred Heibel: "Using Object Modeling CASE Tools" <http://www.dbmsmag.com/9707d16.html>

⁷ WebSite Methodik, Informationsmethodik V: <http://www.iud.fh-darmstadt.de/methodik/iv/ss99/im5/monatb3.htm>

Diagramme, z.B. Klassendiagramme, Zustandsdiagramme, Use-Case-Diagramme und Diagramme für Interaktion und Implementierung. Die meisten der führenden objektorientierten Entwicklungsmethoden (wie z.B. Rational Unified Process, ROOM und Fusion2) benutzen nun die UML als ihre Analyse- und Designnotation (Collin Atkinson, Arbeitsgruppe "Systematische Softwareentwicklung", Fraunhofer Einrichtung für Experimentelles Software Engineering (IESE)).

Klassendiagramme zeigen die logisch-statische Struktur eines Systems, also seine Objektklassen und deren Beziehungen zueinander. Oft wird unterschieden zwischen Objektdiagrammen und Klassendiagrammen, wobei Objektdiagramme einzelne Objekte und deren Beziehungen zeigen. In der Praxis ist eine eindeutige Trennung aber nicht immer möglich oder nötig, so daß auch Klassendiagramme mit Objekten und umgekehrt existieren.

Message-Trace-Diagramme: Zur Verdeutlichung der Funktionsweise der verschiedenen Komponenten eines Systems werden im Rahmen der Unified Modeling Language auch sogenannte Message-Trace-Diagramme verwendet. Diese Diagramme stellen Folgen von Interaktionen zwischen Komponenten des Systems dar, wobei unterschiedliche Szenarien entwickelt werden können.

Das Object-Message Diagramm zeigt, wie auch das Message-Trace Diagramm, ein Szenario. Während Message-Trace Diagramme zeitliche Abläufe in den Vordergrund stellen, zeigt das Object-Message Diagramm diejenigen Objekte und deren Beziehungen, die unmittelbar vor Beginn einer betrachteten Operation existieren oder während der Operation erzeugt (und ggf. wieder gelöscht) werden.

Focus-of-control Diagramme sind verfeinerte Message-Trace Diagramme. Sie zeigen zusätzlich die Zeiträume, in denen ein Objekt über die Kontrolle des Ablaufs verfügt. Dargestellt werden diese Zeiträume durch doppelte, senkrechte Linien, wobei ein ausgefüllter Zwischenraum die volle Kontrolle verdeutlicht, während ein leerer Zwischenraum aussagt, daß das Objekt zu diesem Zeitpunkt "im Bereich" der Kontrolle liegt (in dem etwa von dort eine Funktion aufgerufen wurde, deren Fehlerkontrolle beim aufrufenden Objekt liegt).

Die Formulierungen sind z.T. [Vincent Winczewski: Entwurf und Implementierung ausgewählter, zentraler Komponenten eines Informationssystems für die Mathematik in Deutschland. (Diplomarbeit), Kapitel 4 Systemarchitektur und Datenmodell⁸] entnommen, wo sich auch konkrete Beispiele für diese Diagramme finden."

⁸ <http://elib.zib.de:88/~winczewski/diplomarbeit/kap4.htm>

5 Bibliographie

Nr.	Name des Tools	Kurzbeschreibung	Besonderheiten	Preis (ca.)	URL
1	Case/4/0	case/4/0 integriert Analyse-, Design- und Programmierwerkzeuge zu einem durchgängigen Tool für die Softwareentwicklung nach strukturierten Methoden.	Aus den graphischen Ergebnissen des Softwareentwurfs generiert Case/4/0 standardmäßig C, COBOL, Java, Visual Basic und Datenbankdefinitionen für relationale Datenbanken.	12.898 DM	http://www.microtool.de/ca/se.d/ca_430d_main.htm
2	ConceptBase	ConceptBase ist ein deduktiver Multi-user Objekt-Manager, dessen Einsatzgebiet hauptsächlich das konzeptionelle Modellieren und Koordinieren von Entwicklungsumgebungen ist. Das System enthält O-Telos, einen Dialekt der deduktiv-objektorientierten Sprache Telos.	Alle Klassen, Meta-Klassen, Instanzen, (etc.) werden als Objekte repräsentiert.	k. A.	http://www-i5.informatik.rwth-aachen.de/CBdoc/cbflyer.html
3	DBMain	Datenmodellierungstool Datenbankdesign Reverse-Engineering Integration u. Verwaltung Eigene Programmiersprache Flexibler Import/Export	Objekt-Orientierte Schemas Programmiersprache Voyager	2440 DM bis 4890 DM	http://www.info.fundp.ac.be/~dbm/informations.html
4	DeZign	Datenmodellierungstool Erstellen von Datenbanken für Desktop + Client/Server Anwendungen Darstellung anhand von ER-Modellen	" is a powerful, easy-to-use database design tool"	Voll-Version: 180 DM Günstigere Version (unterstützt nur Dben dBase und Paradox) 125 DM	http://www.heraut.demon.nl/dezign/dezign.html

5	EasyCase 4.2	EasyCASE 4.2 Professional - CASE-Tool für Windows 3.1/95/NT Werkzeug für die Analyse und Dokumentation von Softwareprojekten formale Festlegung der Leistungen für ein Softwareprojekt	Analyse und Aufteilung der Einzelleistungen Genauere Dokumentation des Leistungsumfangs in einem Softwareprojekt	3242 DM	http://www.microway.com.au/catalog/evergreen/easycase.html http://www.comfood.de/easycasi.phtml
6	EasyER/ EasyOBJECT	Planungswerkzeug für Datenbank- und Objekt design	UML-fähig	4337 DM	http://www.comfood.de/easyert.htm
7	ERwin /ERX	ER-Modellierungswerkzeug f. 30 relationale DB-Systeme. Datenmodellierungswerkzeug für den datenbanknahen Applikations-Entwickler. DBMS-unabhängige Datenmodellierung inkl. "Trigger" und "Stored Procedures" für alle wesentlichen SQL- und Desktop-Datenbanken.	Modellierungstools von Logic Works	k. A.	http://www.softguide.de/prog_e/pe_0063.htm http://www.activex.de/logic.htm
8	GRADE Modeler	Toolset für: -Datenmodellierung- und Simulation Business process engineering / reengineering	Ab 15. April 1999: GRADE Modeler 4.0 update 4.0.8 erhältlich Neue Features: UML package management	k.A.	http://www.latnet.lv/LU/MI/GRADE/
9	Innovater 5.0	Werkzeugfamilie für Objekt-, Funktions-, Daten-Modellierung und Business Engineering. I-CASE Lösung C/C++, Small Talk, Cobol, Object Cobol, Fortran	Unabhängig von Betriebssystemen und Plattformen und damit in allen gängigen Netzen einsetzbar.	k. A.	http://rzserv2.fh-lueneburg.de/user/slots/fopre/fopre/node8.html
10	Modelator	Datenmodellierungs- und Datenbankdesign-Tool Modelator (Version 4) is a powerful data modelling and database design tool	In England vertrieben von <u>Webstar Plc.</u> in Verbindung mit <u>Follet Management Systems</u>	Version für Studenten 1150 DM Für Unternehmen/ Professionelle Nutzung 1830 DM	http://www.modelator.webstar.co.uk/
11	ObjectIF 2.0	Multi-userfähiges objektorientiertes CASE Tool -Use-Case-Analyse -objektorientiertem Design		3911 DM	http://www.microtool.de/objecte.d/index.htm

		-Codegenerierung -Reverse Engineering			
12	Oracle Designer/ 2000	Oracle Designer unterstützt den gesamten Software-Entwicklungsprozeß, von der Strategie-, Analyse- und Designphase bis hin zur Modulspezifikation des Anwendungssystems. Geschäftsprozeßmodellierung, erweiterte Entity-Relationship-Modellierung, Objektmodellierung nach UML (inklusive des Object Database Designers)	Funktionsmodellierung Strukturierte Datenflußmodellierung Rapid Application Design (RAD)	von 11090 DM bis 14642 DM	http://www.oracle.de/orcl/dbsi/Main.htm http://oraclestore.oracle.com/
13	PCase	PCase ist ein einfaches CASE Tool, um die Struktur einer Progress Datenbank mit einem Entitäten-Relationen-Diagramm zu visualisieren.	Import und Aktualisierung von existierenden Datenbanken, Erzeugen von Standard-Triggern und Pflege von Datenbankversionen	Version für 1-2 Nutzer 2958 DM Version für unbegrenzte Anzahl von Nutzern 4524 DM	http://www.tools4progress.com/pcase1.html
14	Rational Rose	Visuelles Modellierungs-Tool, erlaubt es dem Entwickler, eine Software-Architektur zu definieren.	UML, OMT, Booch. Use Case, Class, Sequence, Collaboration, Statechart, und andere Diagrammtypen . Code-Generierung. Report-Generierung. Reverse Engineering.	3352 DM Rational Rose 98i Modeler Edition	http://www.rational.com/products/rose/index.itmpl
15	S-Designor	Das im S-Designor DataArchitect implementierte bi-level-Design gestattet absolut datenbankunabhängiges Modellieren von komplexen Datenbankprojekten. Die Überleitung des konzeptionellen ins physikalische Layout erfolgt automatisch, der S-Designor kennt über 30 Datenbanken.	Die direkte Datenbank-anbindung erlaubt automatische Updates der Datenbankstruktur bei Veränderung der Vorgaben.	1800 DM	http://www.cosy.sbg.ac.at/~pmeerw/S-Designor/sdesignor_CDM.html
16	Silverrun	SILVERRUN umfaßt Module zur Erstellung konzeptioneller Datenmodelle (Entity Relationship Expert (ERX)), zur	Von Hirschleber&Partner SILVERRUN ist	k.A.	http://www.activex.de/silverrun.htm http://www.silverrun.com/products.htm

		Geschäftsprozeßmodellierung (Business Process Modeler (BPM)) und zum Design logischer und physikalischer Daten- sowie Applikationsmodelle (Relational Data Modeler (RDM)).	unter 5 verschiedenen Plattformen (Windows, Windows NT, OS/2, Mac, SUN) verfügbar		
17	Software Through Pictures (StP)	Software through Pictures® for Structured Environments (StP/SE) ist eine strukturierte Analyse- und Designumgebung für die Modellierung neuer oder existierender Systeme. StP/SE unterstützt objekt-orientierte Analyse und Design, Objektmodellierung, Informationsmodellierung, strukturiertes Entwickeln und Code-Generierung.	Von: aonix	14731 DM	http://www.aonix.com/Products/SMS/StP_SE.html
18	System Architect 2001	System Architect 2001 unterstützt "Enterprise Modeling" aus verschiedensten Perspektiven. System Architect 2001 unterstützt außerdem die aktivitätsbasierte Kostensimulation. Methoden: OMT, Booch, Shlaer/Mellor, Coad/Yourdon, Jacobson Use Cases. System Architect unterstützt C++ und Java, und generiert Code für eine Vielzahl anderer Sprachen, wie z.B. CORBA, IDL, Visual Basic, Delphi, und PowerBuilder.	Von :Popkin Software System Architect 2001 unterstützt UML 1.1, den neuen Standard für objekt-orientierte Analyse und das Design von Systemen.	k.A.	http://www.popkin.com/products/sa2001/data/data.htm
19	Visible Analyst 6.2	Für Studenten (ältere Version)		229 DM	http://www.comfood.com/visana_i.phtml
20	Visible Analyst 7.0	CASE-TOOL für Windows 95/NT mit Codegenerator für Cobol und C. Mit Reverse Engineering aus SQL-DDL (Inkl. 1 Jahr Wartungsvertrag)		7540 DM	http://www.comfood/visana_i.phtml
21	Visio Professional 5.0	Netzwerkmanagement Datenbank-Analyse und Design Datenmodellierung Software-Entwicklung Internet- oder Intranet- Design		800 DM	http://www.visio.com
22	xCase Professional 4	Datenmodellierungs- und Analysewerkzeug		1480 DM	http://www.xcase.com/products.htm

k.A. = keine Angaben bezüglich des Preises gefunden

6 Links

6.1 CASE

CASE Tools: <http://osiris.sunderland.ac.uk/sst/casehome.html>

Vendor List: <http://osiris.sunderland.ac.uk/sst/case2/vendors.html>

Rational "Rose" (Marktführer OO Softwareentwicklung): <http://www.rational.com>

6.2 OO

Object-Oriented Information Sources: <http://iamwww.unibe.ch/~scg/OOinfo/index.html>

OO-Information Links: <http://ourworld.compuserve.com:80/homepages/rtdue/>

Using Object Modeling CASE Tools: <http://www.dbmsmag.com/9707d16.html>

6.3 UML

Website Methodik: <http://www.iud.fh-darmstadt.de/methodik/lv/ss99/im5/monatb3.htm>

UML Resource Center: <http://www.rational.com/uml/index.jttml>

Visual UML: <http://www.visualuml.com/products.htm>

UML Tools: <http://www.essaim.univ-mulhouse.fr/uml/outillage/outillage.html>

7 Quellen

Webseiten der Hersteller der von uns untersuchten Programme:

DBMAIN <http://www.info.fundp.ac.be/~dbm>

EasyER <http://www.comfood.com/easyerd2.phtml>

DeZign <http://www.heraut.demon.nl/dezign/dezign.html>

Weitere Webseiten:

<http://www.softguide.de/software/case.htm>

<http://osiris.sunderland.ac.uk/sst/case2/vendors.html>

Literatur:

Knorz, G.: Datenbank-Entwurfsmethoden

In: Buder, M.; W. Rehfeld; T. Seeger, D. Strauch [Hrsg]: Grundlagen der praktischen Information und Dokumentation. 4. Aufl.

München, New York: Saur-Bowker 1997. S.664-686.

Meier, A.: Relationale Datenbanken: eine Einführung in die Praxis, 3. Aufl.

Berlin, Heidelberg, New York: Springer 1998

8 INDEX

4GL-Programmiersprache	15	logischer Entwurf.....	6
Access.....	27	Martin.....	27
Analyse	5	Mustererkennungs-Engine.....	15
Anforderungen.....	7	Normalform.....	7
Anforderungsanalyse.....	6	Notationen	27
Attribute.....	7	Object Modeling Technique	31
Attribut-Tabellen-Editor	7	Objektorientierte CASE-Tools.....	5
Bachman	27	objektorientierte Datenbankentwicklung	31
Beispiel Bücherei.....	13	Objektorientierte Notationen	28
Booch.....	31	ODBC	25
CASE.....	5	ODBC-Schnittstelle.....	8
Coad-Yourdon.....	28	OMT <i>Siehe</i> Object Modeling Technique	
COBOL	15	OO-Ansätze.....	31
CODASYL.....	15	Optimierung des SQL-Codes.....	8
Coding.....	17	ORACLE 8i	5
Data Dictionary	24	Paraphrasor	15
Datenaustausch	8, 11	Physical Design	17
Datentyp-Definition	8	physischer Entwurf	6
DBMAIN	11	Relationen.....	7
DBMAIN-Projekt	14	Reports.....	8
Design.....	5	Repository.....	7
DeZign	11, 21	Rumbaugh OMT	28
Dokumentation	8	Shlaer-Mellor	27
EasyER.....	11	Softwarekrise	4
E-Commerce	4	Software-Lebenszyklus.....	5
Entities	7	SQL	7
Entwurfsphasen	6	SQL Anywhere.....	24
Fremdschlüssel	7	SQL-Statements	21
Graphischer ER-Editor	7	strukturierte Design-Methoden	8
GUI	9	Test.....	5
IDEF1X.....	27	Transformations-Assistent	14
IDS/II.....	15	UML... 28 <i>Siehe</i> Unified Modeling Lanuage	
Implementierung	5	Unified Modeling Lanuage	31
Implementierungsphase	6	Universität Namur.....	20
Import.....	8	Versionskontrollsystem	21
Indizes	7	Visible Analyst	24
Jacobsen Use Cases	31	Voyager 2	15
JAVA	5	Wartung	5
JET	25	Website Methodik	31
Konsistenz	7	Wizards.....	8
konzeptioneller Entwurf	6	Zoom.....	8
Logical Design.....	17		

9 Anhang

Vortragsfolien