

# Eine integrierte CASE-Umgebung zur Entwicklung von Informationssystemen: Ein Erfahrungsbericht

Peter Jaeschke

Institut für Angewandte Informatik und Formale Beschreibungsverfahren

Universität Karlsruhe(TH)

D-76128 Karlsruhe

April 1993

E-mail: jaeschke@aifb.uni-karlsruhe.de

## 1. Einführung

Am Institut für Angewandte Informatik und Formale Beschreibungsverfahren der Universität Karlsruhe wurde ein Prüfungsverwaltungssystem unter Einsatz von Oracle CASE<sup>1</sup> und INCOME<sup>2</sup> entwickelt. Die gewählte Vorgehensweise orientiert sich an der Oracle CASE Methode [BaLo92, Bark90a/b] bzw. dem Information Engineering Ansatz von James Martin [Mart89, Mart90a/b].

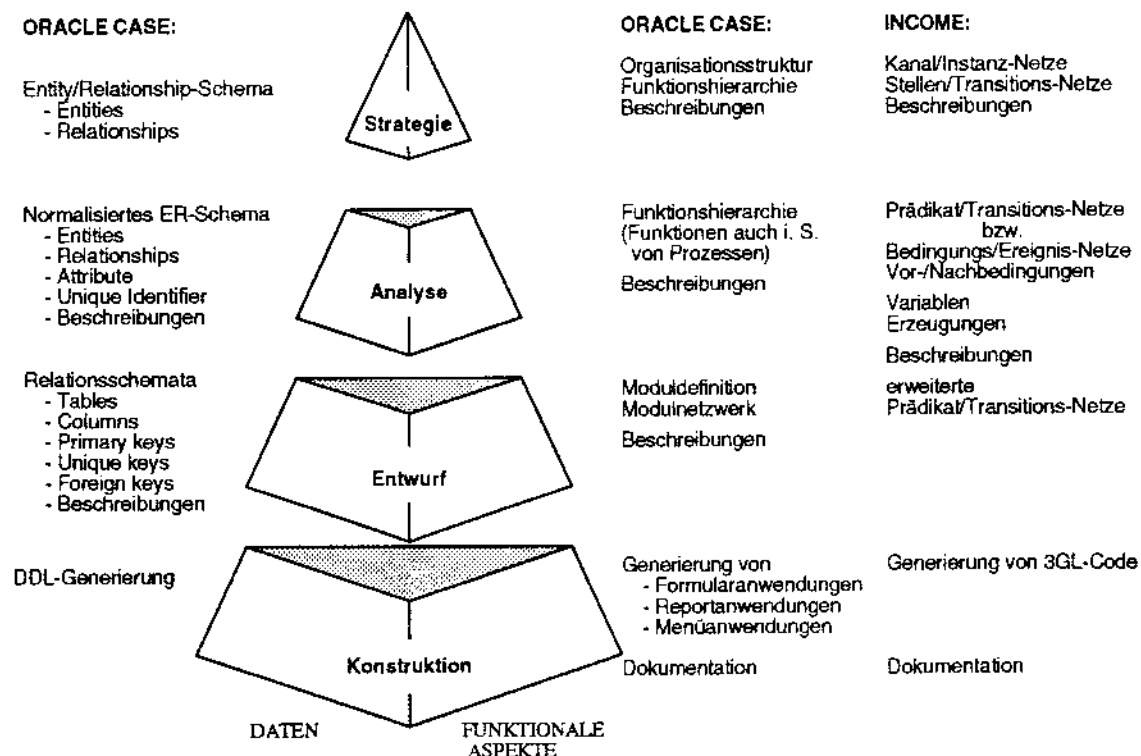


Abb. 1: Oracle CASE Methode mit Oracle CASE und INCOME

<sup>1</sup> Oracle CASE ist eine Produktfamilie der ORACLE Corporation, Belmont, CA, USA  
<sup>2</sup> INCOME ist ein Produkt der PROMATIS Informatik, Karlsbad, Deutschland

Die ausgewählten CASE-Tools unterstützen die in Abb. 1 schematisch dargestellte Vorgehensweise durchgängig, d.h. in jeder Phase stehen Utilities zur Verfügung, die die Ergebnisse der vorangegangenen Phase für die Weiterbearbeitung aufbereiten und den Übergang im Sinne von Standardumsetzungen durchführen.

## 2. Vorgehensweise

ORACLE\*CASE unterstützt den konzeptuellen Entwurf von Datenbankschemata auf der Basis eines binären ER-Modells, das um Subtypen und sogenannte Arc's [Bark90a] erweitert ist. Für die Modellierung der funktionalen Systemaspekte stellt Oracle CASE einerseits Funktions- bzw. Prozeßhierarchien und andererseits Datenflußdiagramme zur Verfügung. Da Datenflußdiagramme den Anforderungen der Prozeßmodellierung nur unzureichend genügen, wurde stattdessen INCOME [NOSS92, INC93] - ein CASE-Tool, das Petri Netze verwendet - eingesetzt. Dadurch kann insbesondere zur präzisen und kompakten Beschreibung von Prozessen und Geschäftsvorfällen auf höhere Petri-Netze (Prädikat/Transitions-Netze) zurückgegriffen werden. INCOME ist sowohl methodisch als auch funktional voll in die repository-basierte Oracle CASE-Umgebung integriert (Abb. 2).

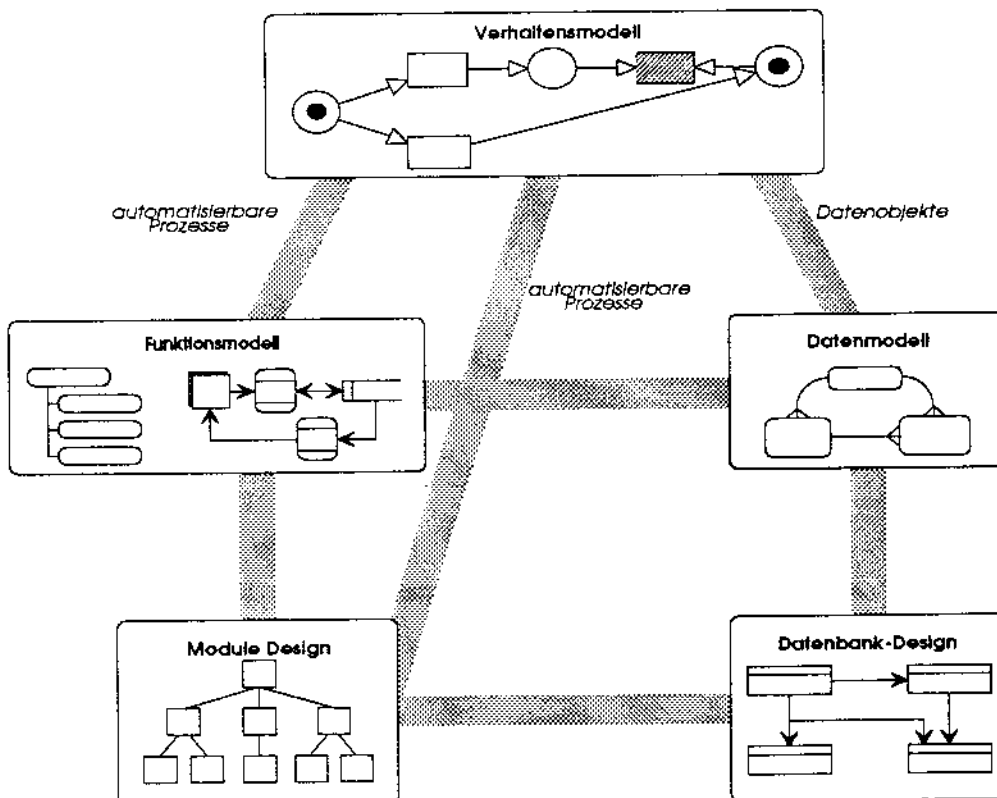


Abb. 2: Integration von INCOME und Oracle CASE

## 2.1. Strategie und Analyse

Zunächst wurde eine Informationsbedarfsanalyse in Form von Datenbestands-, Interview- und Formularanalysen durchgeführt. Basierend auf diesen Analysen wurde ein normalisiertes ER-Schema entworfen, für das Attribute und Unique Identifier festgelegt wurden. Das Ergebnis des Entwurfs wurde mit Quality-Checks, die von Oracle CASE angeboten werden, validiert. Hierbei ist zu beachten, daß bei der Definition der Unique Identifier keine Surrogate verwendet werden sollten, sondern ausschließlich berücksichtigt werden sollte, wie der Endbenutzer Entities eindeutig identifiziert. Ist ein Entity nur über Surrogate identifizierbar, so ist dies normalerweise ein Indikator dafür, daß entweder Attribute oder Relationships fehlen. Identifier können in Oracle CASE sowohl aus Attributen als auch aus Relationships bestehen.

Im Anschluß daran wurde die Funktionshierarchie aufgestellt und die Verwendung der Entities und ihrer Attribute durch die Funktionen spezifiziert. Diese Zuordnung war Grundlage weiterer Qualitätskontrollen. So muß beispielsweise für jedes Entity mindestens jeweils eine Funktion existieren, die es erzeugt, verändert, löscht bzw. liest. Es kann sich dabei auch um eine einzige Funktion handeln, die alle vier Operationen durchführt. Ausnahmen sind für einzelne Entities möglich, so darf ein Buchungssatz nach seiner Erstellung nicht mehr modifiziert werden. Diese Überlegungen können analog auf Attribute übertragen werden.

Die Abläufe wurden zunächst durch Stellen/Transitions-Netze modelliert, die dann zu Prädikat/Transitions-Netzen verfeinert wurden. Den Prädikatsstellen und Variablen wurden Entities und Attribute, den Transitionen entsprechende Funktionen aus der Hierarchie zugeordnet.

## 2.2. Entwurf

Der Übergang vom ER-Schema zum relationalen Datenbankschema wird von Oracle CASE durch entsprechende Utilities unterstützt. Beim Übergang vom ER-Schema zu den Relationsschemata kann die Umsetzung von Subtypen individuell für jede Subtyphierarchie festgelegt werden. Bei einem sorgfältigen konzeptuellen Entwurf werden Tables, Columns, Primary Keys, Unique Keys und Foreign Keys bis auf wenige Ausnahmen generiert. In den Protokollen wird an den entsprechenden Stellen auf eine eventuell notwendige manuelle Nachbearbeitung hingewiesen. Die Key Constraints mußten hinsichtlich des zulässigen Verhaltens bei Änderungen von Schlüsselwerten und dem Löschen von Tupeln überarbeitet werden:

- Können Schlüsselwerte nach der Erfassung verändert werden ?
- Cascading Updates/Deletes u.a.

Ein Utility, das über Eingabeparameter kontrolliert wird, leitet aus den Funktionen Module ab. Bei der betrachteten Anwendung waren keine größeren Änderungen der Moduldefinitionen notwendig. Die Module wurden anschließend im Detail spezifiziert. Es wurde einerseits das Modulnetzwerk -die gegenseitige Aufrufstruktur zwischen den Modulen- aufgebaut und anderer-

seits die Verwendungsmatrizen der einzelnen Module -es wird für jede einzelne Column einer Table festgelegt, welche Operationen auf ihr zulässig sind - erstellt bzw. vervollständigt.

### **2.3. Konstruktion**

Die Oracle CASE-Generatoren generieren auf der Basis des Modulnetzwerkes die gesamte Menüstruktur des Anwendungssystems. Die Verwendungsmatrizen werden als Grundlage für die Generierung von Formular- und Report-Anwendungen ausgewertet, der Code zur Überwachung der Integritätsbedingungen erfolgt gemäß den Angaben bei der Definition der Primary Keys, Unique Keys und Foreign Keys. Dementsprechend wird der Code zur Überwachung der referentiellen Integritätsbedingungen generiert. Die Zielumgebung der Oracle CASE-Generatoren sind die entsprechenden Oracle-Entwicklungstools: Oracle Forms, Oracle Reportwriter, Oracle Menu, PL/SQL und SQL\*Plus.

### **3. Beurteilung**

Oracle CASE und INCOME unterstützen den gesamten Entwicklungsvorgang und die Phasenübergänge durchgängig. Wünschenswert wäre auf Seiten von Oracle CASE noch, den Übergang zwischen Entwurf und Analyse in beide Richtungen zu unterstützen. Hingegen bietet das sogenannte 'Cross Referencing' in Bezug auf den Datenbankentwurf jederzeit die Möglichkeit den Inhalt des Repositories mit dem Inhalt des Online-Dictionaries abzugleichen.

Oracle CASE und INCOME bieten verschiedene Auswertungen bzw. Reports standardmäßig an, um für die einzelnen Entwicklungsschritte aktuelle technische Dokumentationen bzw. Rohfassungen für Benutzerhandbücher zu erstellen. Soweit diese Reports nicht ausreichen bzw. den individuellen Anforderungen nicht genügen, lassen sich über vordefinierte Views individuelle Reports auf einfache Weise programmieren.

Die Oracle CASE-Generatoren erzielen unter der Voraussetzung einer entsprechend detaillierten Moduldefinition einschl. exakter Verwendungsangaben für jede einzelne Column und ebenso exakter Angaben für Key Constraints sehr gute bis gute Ergebnisse. Ausnahmen hiervon treten nur auf, wenn für einzelne Module ein Umgehen der Basisfunktionalität der Zielumgebung - Oracle Reportwriter oder Oracle Forms - erforderlich ist. Gelegentlich entsteht erhöhter Aufwand für manuelle Nachbearbeitung dadurch, daß die Verwendung von Tabellen - entweder als Basis-Tabelle für einen Oracle Forms-Block oder als "Nachschlage"-Tabelle - mit dem Fremdschlüsselkonzept gekoppelt ist.

Die Oracle CASE-Generatoren unterstützen Prototyping nur unzureichend, bzw. nur im Sinne des evolutionären Prototyping jedoch nicht im Sinne des explorativen oder experimentellen Prototypings. D.h., Prototyping basierend auf den Oracle \*CASE-Generatoren kann erst in der Entwurfsphase sinnvoll eingesetzt werden. Hingegen bietet der INCOME/Simulator die

Möglichkeit des interpretativen Prototyping zur Überprüfung der modellierten Abläufe und unterstützt dadurch bereits in Strategie- und Analyse-Phase die Einbindung des Endanwenders in die Entwicklung.

Nachteilig wirkt sich aus, daß Relationships in Oracle CASE - wie in allen binären ER-Modellen - keinen eigenständigen Objektcharakter besitzen. Aufgrund dieser Einschränkung ist es auf konzeptueller Ebene nicht möglich, zu spezifizieren, welche Relationships durch welche Funktionen bzw. Transitionen manipuliert werden können. D.h., es können weder Funktionen/Relationships- noch Transitionen/Relationships-Zuordnungsmatrizen aufgestellt werden. Diese Zuordnungen können implizit erst auf logischer Ebene erfolgen, da hier für die einzelnen Module angegeben werden kann, wie sie Fremdschlüsselwerte manipulieren. Es ist daher beispielsweise nicht möglich, beim konzeptuellen Entwurf zu prüfen, ob jede Beziehung von mindestens einer Funktion aufgebaut oder manipuliert werden kann.

Die Kombination von Oracle CASE und INCOME stellt für die Anwendungsentwicklung ein effizientes und umfassendes Tool zur Verfügung. Es ist jedoch zu beachten, daß für ein einzelnes Projekt nicht jede Komponente von INCOME bzw. Oracle CASE benötigt wird, sondern die verwendeten Komponenten projektspezifisch zusammengestellt werden können.

### Literaturverzeichnis

- [BaLo92] Barker, R. und Longman, C.: CASE\*METHOD: Function and process modelling, Addison Wesley Publishing Company, Wokingham, England, 1992
- [Bark90a] Barker, R.: CASE\*METHOD: Entity relationship modelling, Addison Wesley Publishing Company, Wokingham, England, 1990
- [Bark90b] Barker, R.: CASE\*METHOD: Tasks and deliverables, Addison Wesley Publishing Company, Wokingham, England, 1990
- [Chen76] Chen, P. P.: The entity-relationship model: Toward a unified view of data, ACM TODS 1 (1976), No.1, 166 - 192
- [INC93] INCOME Benutzerdokumentation: INCOME/Designer, INCOME/Dictionary, INCOME/Simulator, INCOME/Generator, PROMATIS Informatik, Karlsbad, 1992
- [NOSS92] Németh, T., Oberweis, A., Schönthaler, F., Stucky, W.: INCOME: Arbeitsplatz für den Programmentwurf interaktiver betrieblicher Informationssysteme Forschungsbericht 251, Institut für Angewandte Informatik und Formale Beschreibungsverfahren, Univ. Karlsruhe, 1992
- [Mart89] Martin, J.: Information Engineering, Book I: Introduction, Prentice Hall, Englewood Cliffs, New Jersey, 1989
- [Mart90a] Martin, J.: Information Engineering, Book II: Planning and Analysis, Prentice Hall, Englewood Cliffs, New Jersey, 1990
- [Mart90b] Martin, J.: Information Engineering, Book III: Design and Construction, Prentice Hall, Englewood Cliffs, New Jersey, 1990
- [ScSt83] Schlageter, G.; Stucky, W.: Datenbanksysteme: Konzepte und Modelle, 2. Auflage, B. G. Teubner, Stuttgart, 1983