

# Zweitveröffentlichung



Sinz, Elmar J.

## Objektorientierte Analyse (ooA)

Datum der Zweitveröffentlichung: 19.09.2024

Akzeptiertes Manuskript (Postprint), Zeitschriftenartikel

Persistenter Identifikator: urn:nbn:de:bvb:473-irb-981780

## Erstveröffentlichung

Sinz, Elmar J. (1991): „Objektorientierte Analyse (ooA)“. In: Wirtschaftsinformatik : WI, Jg. 33, Nr. 5, S. 455-457, Wiesbaden: Springer Gabler.

## Verlagshinweis

This version of the article has been accepted for publication, after peer review (when applicable) and is subject to Springer Nature's AM terms of use, but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections.

## Rechtehinweis

Dieses Werk ist durch das Urheberrecht und/oder die Angabe einer Lizenz geschützt. Es steht Ihnen frei, dieses Werk auf jede Art und Weise zu nutzen, die durch die für Sie geltende Gesetzgebung zum Urheberrecht und/oder durch die Lizenz erlaubt ist. Für andere Verwendungszwecke müssen Sie die Erlaubnis der Rechteinhaberinnen und Rechteinhaber einholen.

Für dieses Dokument gilt das deutsche Urheberrecht.

# Das aktuelle Schlagwort

## Objektorientierte Analyse (ooA)

Elmar J. Sinz\*

### 1 Methodische Einordnung von ooA

Objektorientierte Ansätze werden im Zusammenhang mit nahezu allen Komponenten von Anwendungssystemen und Phasen der Anwendungsentwicklung diskutiert. Um diese unterschiedlichen Ansätze zu systematisieren, werden im folgenden zwei einfache Modelle mit zueinander orthogonalen Gliederungsprinzipien verwendet; ein einfaches Programmstrukturmodell für Anwendungssysteme (Bild 1a) und das als Wasserfallmodell bekannte Phasenmodell der Anwendungsentwicklung (Bild 1b).

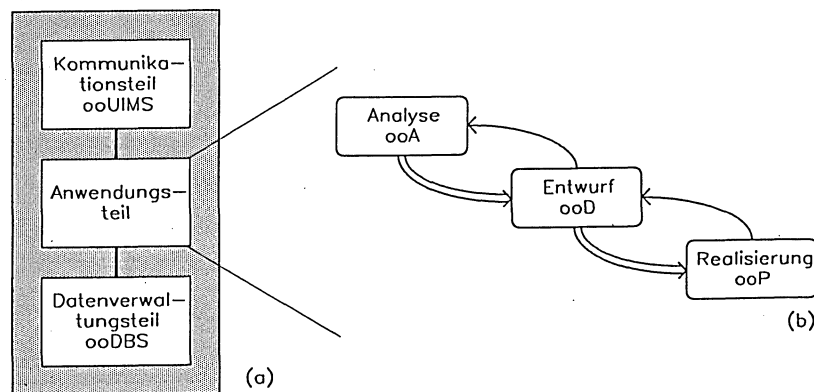
Den einzelnen Komponenten des Programmstrukturmodells sind folgende objektorientierte Ansätze zuordenbar:

- Für den *Kommunikationsteil*, dessen Aufgabe die Realisierung der Nutzerkommunikation ist, stehen objektorientierte Paradigmen für Nutzeroberflächen und objektorientierte UIMS (User Interface Management System) zur Verfügung (z.B. [8]).
- Für den *Datenverwaltungsteil* werden zunehmend objektorientierte Datenbanksysteme (ooDBS) (z.B. [1]) und zugehörige objektorientierte Daten(bank)modelle (z.B. [5]) diskutiert.

- Aus Sicht der Wirtschaftsinformatik werden ooUIMS und ooDBS als verfügbare Komponenten betrachtet. Der *Anwendungsteil*, der den funktionalen Kern der Anwendung enthält, soll dagegen zusätzlich unter Entwicklungsgesichtspunkten in die Hauptphasen Analyse, Entwurf und Realisierung zerlegt werden. Diesen Phasen sind die Ansätze der objektorientierten Analyse (ooA) (z.B. [3], [11]), des objektorientierten Entwurfs (Design, ooD) (z.B. [2], [9]) und der objektorientierten Programmierung (ooP) (z.B. [4]) zuordenbar.

Während zu ooP inzwischen ausgereifte Programmierumgebungen verfügbar sind, wird ooD erst seit wenigen Jahren intensiver untersucht. Die aktuellsten Ansätze widmen sich nun der objektorientierten Analyse, deren Grundprinzipien im folgenden dargestellt werden sollen.

Weitere objektorientierte Ansätze treten im Zusammenhang mit der Entwicklung wissensbasierter Systeme auf. Beispiele hierfür sind die Wissensrepräsentation mit Frames, objektorientierte Wissensrepräsentationssprachen und objektorientierte Shells.



**Bild 1** Programmstrukturmodell für Anwendungssysteme (a) und Phasenmodell der Anwendungsentwicklung (b)

\* Prof. Dr. Elmar J. Sinz, Universität Bamberg, Lehrstuhl für Wirtschaftsinformatik, insbes. Systementwicklung und Datenbankanwendung, Feldkirchenstraße 21, D-8600 Bamberg, e-mail X.400: sinz@sowi.uni-bamberg.dbp.de

## 2 Aufgabenstellung und Ziele von ooA

Aufgabe der objektorientierten Analyse (ooA) ist die Definition der fachlichen Anforderungen eines Anwendungssystems in objektorientierter Form. Bevor hierauf näher eingegangen wird, wird kurz die Struktur und Funktionsweise eines Anwendungssystems in objektorientierter Sichtweise beschrieben.

Danach besteht ein Anwendungssystem aus einer Menge von *Objekten*, die mit Hilfe von *Nachrichten* interagieren. Eine Nachricht an ein Objekt löst dort die Durchführung eines dem Objekt exklusiv zugeordneten Operators (*Methode*) aus. Diese Methode kann selbst wiederum Nachrichten an weitere Objekte erzeugen. Gleichartige Objekte gehören einer Klasse an. Objekte können als Spezialisierungen allgemeinerer Objekte beschrieben werden (*Generalisierung/Spezialisierung*), deren Eigenschaften sie erben (*Vererbung*). Objekte können außerdem strukturiert sein, d.h. komplexe Objekte können aus Teilobjekten zusammengesetzt sein (*Aggregation/Zerlegung*).

Eine gemäß ooA erstellte Anforderungsdefinition spezifiziert Struktur- und Verhaltensmerkmale eines Anwendungssystems aus fachlicher Sicht in objektorientierter Form. Den Kern dieser Anforderungsdefinition bilden Spezifikationen von

- *Klassen* mit exklusiv zugeordneten Attributen und Methoden,
- *Generalisierungs-/Spezialisierungsstrukturen*, bestehend aus allgemeinen und spezialisierten Klassen sowie
- *Aggregations-/Zerlegungsstrukturen*, bestehend aus zusammengesetzten Klassen und ihren Komponentenklassen.

ooA vereinigt damit die zentralen Abstraktionsprinzipien klassischer Analysemethoden. Darüber hinaus werden mit ooA mehrere Ziele verfolgt, die mit klassischen Analysemethoden nur beschränkt erreichbar sind. Die wichtigsten sind:

- Eine „natürliche“, an den Objekten und Beziehungen der Diskurswelt orientierte Analyse und, als Folge davon, ein höheres Maß an Konsistenz und Problemverständnis.
- Die Gewinnung wiederverwendbarer und erweiterbarer Analyseergebnisse.
- Eine ganzheitliche, Daten- und Funktionsicht integrierende Analyse von Anwendungssystemen.
- Ein „weicher“ Phasenübergang zwischen Analyse, Entwurf und Realisierung.

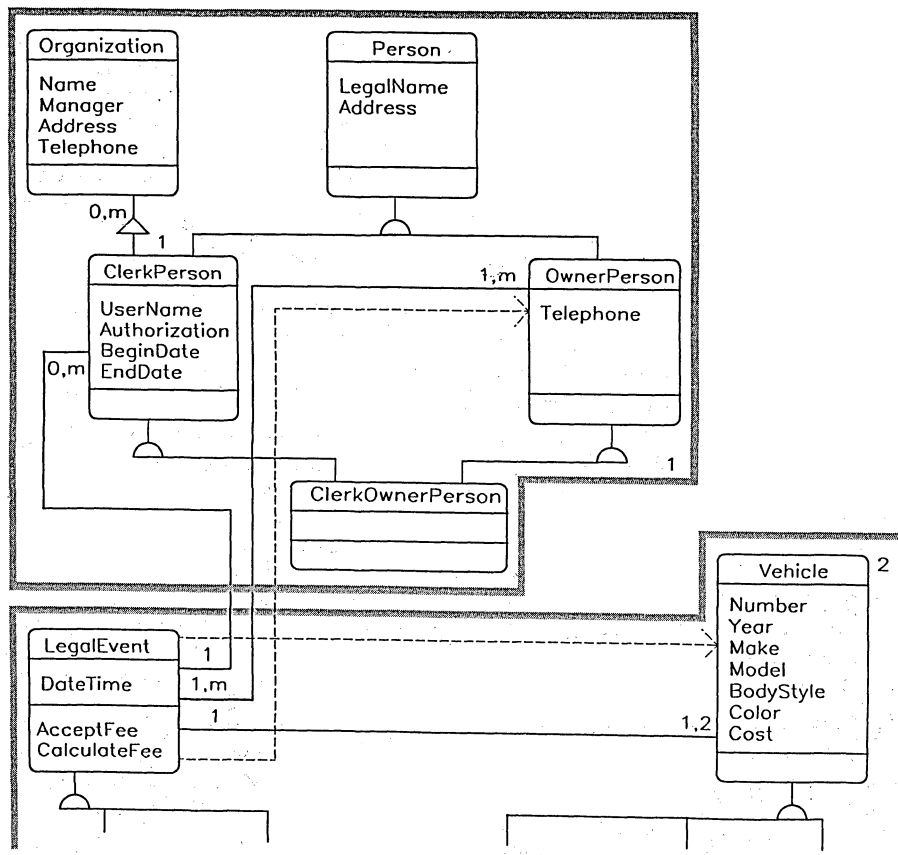


Bild 2 Beispiel zur ooA (vereinfachter Ausschnitt aus [3])

### 3 Ein ausgewählter Ansatz zur ooA

Derzeit wird eine Vielzahl von Vorgehensweisen und Darstellungsformen zur ooA vorgeschlagen. Einen aktuellen Überblick gibt die ausführliche Bibliographie in [2]. Stellvertretend wird im folgenden der Ansatz von Coad und Yourdon [3] vorgestellt und anhand des Beispiels einer KFZ-Zulassungsstelle (Bild 2) illustriert.

Coad und Yourdon unterscheiden fünf Hauptaktivitäten der ooA, deren Ergebnisse in je einer Schicht (layer) dargestellt werden. In Bild 2 sind diese 5 Schichten übereinanderprojiziert. Die einzelnen Schichten und ihre zugehörigen Aktivitäten sind:

- *Class-&-Object layer*: Auffinden von Klassen und Objekten in der Diskurswelt (z.B. *Organization*, *Person*).
- *Structure layer*: Identifikation von Generalisierungs-/Spezialisierungsstrukturen (z.B. *Person* wird spezialisiert zu *ClerkPerson* und *OwnerPerson*) und Aggregations-/Zerlegungsstrukturen (z.B. *ClerkPerson* wird aggregiert zu *Organization*).
- *Subject layer*: Identifikation von Teilmodellen (Umrandete Bereiche 1 und 2).
- *Attribute layer*: Definition von Attributen (z.B. *Name*, *Manager*, *Address* und *Telephone* als Attribute von *Organization*) und Referenzen zwischen Objekten (z.B. durchgezogene Linie zwischen *LegalEvent* und *OwnerPerson*).
- *Service layer*: Definition von Methoden (*AcceptFee* und *CalculateFee* als Methoden von *LegalEvent*) und Nachrichtenverbindungen (z.B. gestrichelter Pfeil zwischen *LegalEvent* und *OwnerPerson*).

### 4 Nutzen und Defizite von ooA

Die klassische Anforderungsanalyse, wie sie heute von nahezu allen marktgängigen CASE-Tools unterstützt wird, unterscheidet zwischen der Funktionsstruktur und der Datenstruktur eines Anwendungssystems. Als Methoden dominieren die Strukturierte Analyse (SA) zur Analyse der Funktionsstruktur und das Entity-Relationship-Modell (ERM) zur Analyse der Datenstruktur (siehe z.B. [10]). Diese klassische Analyse besitzt im wesentlichen zwei Schwachstellen:

- a) Funktions- und Datenstruktur werden getrennt analysiert und sind nur schwer abstimmbare. Viele der bekannten Softwareprobleme, die sich oft erst in den späten Phasen des Lebenszyklus eines Anwendungssystems zeigen, sind auf diese Schwachstelle zurückführbar.
- b) Die Analyse wird auf der Ebene des Anwendungssystems durchgeführt. Die dem Anwendungssystem zugrundeliegende Diskurswelt wird nicht analysiert und modelliert.

Die Ansätze der ooA verfolgen eine ganzheitliche, Daten- und Funktionssicht integrierende Analyse eines Anwendungssystems. Dies stellt einen entscheidenden methodischen Fortschritt im Bereich der Anforderungsanalyse dar, welcher geeignet ist, die Schwachstelle (a) zu beseitigen. Ein weiterer Vorteil von ooA tritt im Kontext einer durchgängig objektorientierten Anwendungsentwicklung zutage. Zwischen ooA, ooD und ooP ist im Vergleich zu klassischen Ansätzen ein „weicher“ Phasenübergang möglich, da die analysierten Objektklassen auch als Grundlage für den Entwurf und die Realisierung verwendbar sind.

Schwachstelle (b) wird dagegen im obigen und in vergleichbaren Ansätzen zur ooA nur gemildert und nicht beseitigt. Zwar ist eine „größere Nähe“ zur Diskurswelt und damit ein besseres Problemverständnis erreichbar, das Auffinden der Klassen und Strukturen beruht aber immer noch weitgehend auf der Analyse textueller Beschreibungen und auf Interviews. Dies birgt die Gefahr gravierender Abweichungen zwischen den fachlichen Anforderungen des Anwendungssystems und der dem Anwendungssystem zugrundeliegenden Diskurswelt in sich. Dieses Problem wird in [6] und [7] aufgegriffen. Hier wird ein Ansatz zur ooA entwickelt, der die Anforderungsanalyse des Anwendungssystems auf der Grundlage einer expliziten Analyse und Definition der Diskurswelt durchführt.

### Literatur

- [1] *Atkinson M.*, et. al.: The Object-Oriented Database System Manifesto. In: Proceedings DOOD '89, Kyoto, December 1989. Reprinted in: Datenbank-Rundbrief, Mitteilungsblatt der GI-Fachgruppe 2.5.1 (Datenbanken), Mai 1990, 28 - 36
- [2] *Booch G.*: Object-Oriented Design with Applications. Benjamin/Cummings Publishing Company, Redwood City, California 1991
- [3] *Coad P.*, *Yourdon E.*: Object-Oriented Analysis. Second Edition, Prentice Hall, New Jersey 1991
- [4] *Cox B.*: Object-Oriented Programming. An Evolutionary Approach. Addison-Wesley, Reading, Massachusetts 1986
- [5] *Dittrich K.R.*: Objektorientierte Datenmodelle als Basis komplexer Anwendungssysteme. Stand der Entwicklung und Einsatzperspektiven. In: Wirtschaftsinformatik 32 (1990) 3, 228 - 237
- [6] *Ferstl O.K.*, *Sinz E.J.*: Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell (SOM). In: Wirtschaftsinformatik 32 (1990) 6, 566 - 581
- [7] *Ferstl O.K.*, *Sinz E.J.*: Ein Vorgehensmodell zur Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell (SOM). Erscheint in: Wirtschaftsinformatik 33 (1991) 6
- [8] *Hartson R.*: User-Interface Management Control and Communication. In: IEEE Software, January 1989, 62 - 70
- [9] *Meyer B.*: Object-oriented Software Construction. Prentice Hall, Cambridge 1988
- [10] *McMenamin S.M.*, *Palmer J.F.*: Strukturierte Systemanalyse, München 1988
- [11] *Shlaer S.*, *Mellor S.J.*: An Object-Oriented Approach to Domain Analysis. In: ACM Software Engineering Notes, July 1989