

Der Entwicklungsprozeß

Wie zu Anfang festgestellt, stellt der eigentliche Entwicklungsprozeß das Hauptproblem bei der Verwendung objektorientierter Techniken dar, u.a. auch durch die vielfältigen Beschreibungsmöglichkeiten

Anknüpfend an die vorige Einheit sollen deswegen noch zwei alternative Vorgehensweisen vorgestellt werden:

Slide 1

- Grady Boochs „Objektorientierte Analyse und Design“
- das Vorgehensmodell der OMT

Entwicklung und Bewertung

Unabhängig von der benutzten Methode lassen sich (wie auch schon bei der BON gesehen) zwei unterschiedliche Tätigkeiten erkennen:

- Erstellung eines Gesamtsystems in mehreren Phasen
- in jeder Phase auftretende Standardaktivitäten

Slide 2

Diese Tätigkeitsfelder lassen sich in unterschiedlichen Ausprägungen in allen drei Methoden wiederfinden. Zuerst sollen jeweils die übergeordneten Aktivitäten beschrieben werden.

Grady Boochs „Objektorientierte Analyse und Design“

Booch unterteilt den Entwicklungsprozeß in folgende Tätigkeitsbereiche:

Makro-Prozeß: der übergeordnete Prozeß der Projektabwicklung, welcher nachfolgend behandelt werden soll.

Slide 3

Mikro-Prozeß: verstärkt in der eigentlichen Entwicklungsphase auftretende Tätigkeiten, welche später zusammen mit den Standardaktivitäten der BON betrachtet werden sollen

Die Trennung soll zum einen kontrollierte Managementtechniken ermöglichen, ohne auf der anderen Seite Kreativität und Innovation zu behindern

Der Makro-Prozeß

Der Makro-Prozeß ähnelt stark dem klassischen Wasserfallmodell und bietet damit folgende Vorteile (vgl. auch Modellierung 1):

Slide 4

- Erzeugung von (extern) bewertbaren Produkten
- Fortschrittskontrolle durch Meilensteine und klar abgegrenzte Phasen
- Erleichterung der Planbarkeit und Risikobewertung

Phasen im Makro-Prozeß

Der Makro-Prozeß ist in folgende Phasen unterteilt:

Konzeptualisierung: Festlegen der Kernanforderung (vgl. Requirements Engineering, Modellierung 1)

Slide 5

Analyse: Entwicklung eines Modells, Beschreibung des Systemverhaltens

Design: Erzeugung einer Architektur für die Implementierung

Evolution: schrittweise Umsetzung des Designs

Wartung: Verwaltung der Systems nach der Auslieferung

Der Prozeß wiederholt sich dabei nach jeder wichtigen Produktversion

Konzeptualisierung

Booch deckt mit der Konzeptualisierung einen Bereich ab, der in klassischen Entwicklungsmethoden nur sehr wenig berücksichtigt wird und teilweise sogar vor dem eigentlichen Requirements Engineering liegt.

Die Aufgaben der Konzeptualisierung sind:

Slide 6

- Entwicklung und Beschreibung einer Vision des neuen Systems
- Erstellung eines (oder auch mehrerer) Prototypen
- Festhalten der Kernanforderungen

Analyse

In der Analysephase soll geklärt, *was* das System macht, im Gegensatz zum Design, wo das *wie* beschrieben wird. Aufgaben sind:

Slide 7

- Identifikation von Hauptfunktionalitäten
- Beschreibung von Szenarien mittels *Use-Cases*, Objektdiagrammen, ...
- Beschreibung von Objektlebenszyklen
- Finden von Mustern zwischen Szenarien und Beschreibung der Muster in Klassendiagrammen
- Aufbau eines Datenverzeichnisses

Design

Die Designphase unterteilt Booch in drei Aktivitäten:

Architekturplanung: Entwicklung einer Schichtenarchitektur, bei der Funktionalitäten auf Schichten verteilt werden

Slide 8

taktisches Design: Festlegung von Vorgehensweisen und Mustern

Versionsplanung: Planung der Reihenfolge der zu implementierenden Systembestandteile

Evolution

Für die eigentliche Systementwicklung schlägt Booch einen evolutionären Ansatz vor, um möglichst früh wichtige Funktionalitäten zu realisieren, und erst später Leistungsmaße u.ä. zu verbessern.

Slide 9

Das Vorgehen wird im Mikro-Prozeß beschrieben, allerdings ist die Einschränkung der Tätigkeiten im Mikro-Prozeß auf die Kernentwicklung nicht zwingend, wie die BON zeigt.

Die Aktivitäten bei der Evolution betreffen dabei hauptsächlich den Entwurf und die Reorganisation der Klassenstruktur.

Wartung

Die Wartung des Systems betrifft hauptsächlich die Verbesserung des bestehenden Systems und nicht die Veränderung der Architektur.

Falls eine Veränderung der Architektur notwendig ist, so beginnt der Makro-Prozeß von vorne.

Slide 10

Vorgehensmodell der OMT

Das Vorgehensmodell der OMT ist in drei Phasen unterteilt:

Analyse: Entwicklung eines Objekt-, Datenfluß- und Kontrollflußmodells

Systemdesign: Aufteilung des Systems in Subsysteme

Slide 11 **Objektdesign:** Identifikation von Implementierungsalgorithmen, Optimierung des Objektmodells

Analyse

Aufgabe der Analysephase ist das Verstehen des Problems und des Anwendungsgebiets und die Dokumentation in einem Objektmodell, einem dynamischen Modell und einem Funktionsmodell. Die einzelnen Schritte sind:

Slide 12 **Anforderungsdefinition:** Analyse von Problembeschreibungen und Aufstellung der Anforderungen

Objektmodellierung: Beschreibung der statischen Struktur des Systems

Dynamische Modellierung: Beschreibung des Zeitverhaltens

Funktionale Modellierung: Beschreibung der funktionalen Zusammenhänge und Datenflüsse

Systemdesign

Das Systemdesign umfaßt u.a. folgende Punkte:

- die Aufteilung des Systems in Subsysteme (Gruppen zusammengehöriger Klassen, Ereignisse, Operationen usw.), welche unterschiedliche *Dienste* anbieten
- Identifikation nebenläufiger Prozesse
- Definition der Strategie für die Datenhaltung
- Definition der Kontrollstrukturen (ereignisbasiert, prozedural)

Slide 13

Objektdesign

Im Objektdesign wird das eigentliche Systemdesign vollzogen. Die einzelnen Schritte sind:

- Zusammenführung von Objekt-, dynamischem und funktionalem Modell
- Entwurf von Algorithmen
- Optimierung der Leistungsmaße
- Implementierung des Kontrollflusses
- Entwurf der Assoziationen
- Entwurf von Softwaremodulen

Slide 14

Standardaktivitäten

Unter Standardaktivitäten sollen hier die Tätigkeiten zusammengefaßt werden, die in jeder Phase des Entwicklungszyklus stattfinden können. Darunter fällt auch der Mikro-Prozeß von Booch, obwohl dessen Verwendung primär bei der eigentlichen Implementierung angesiedelt ist.

Slide 15

Verwandt mit den Standardaktivitäten sind die Bewertungskriterien für Klassen- und Objektstrukturen, weswegen sie hier mit vorgestellt werden.

Standardaktivitäten der BON

Bei der BON werden u.a. folgende Standardaktivitäten identifiziert:

- Identifikation von Klassen
- Klassifikation
- Gruppierung in Clustern
- Definition von Operationen

Slide 16

Es existieren noch fünf weitere Aktivitäten, jedoch treten die hier aufgeführten Aktivitäten am häufigsten auf.

Identifikation von Klassen

Die Entscheidung, welche Klassen gebildet werden, stellt eins der größten Probleme beim Entwurf dar. Die BON gibt dabei eine Reihe von Empfehlungen:

Slide 17

- Klassen sollen abstrakte Datentypen sein und nicht eine Ansammlung von Daten mit zugehörigen Funktionen
- Klassen aus der Analysephase sind häufig durch Konzepte des Anwendungsgebiets definiert, während Designklassen näher an der Implementierung liegen
- Physikalische Objekte stellen nicht immer einen guten Ausgangspunkt für Klassen dar, da es bei der Implementierung eher um abstrakte Eigenschaften geht

Identifikation von Klassen

Slide 18

- Klassennamen sollen so gewählt werden, daß *andere Personen* das Programm leichter verstehen können
- Nachbedingungen von Operationen sollen so gewählt werden, daß abgeleitete Operationen nicht zu sehr eingeschränkt werden, wichtige Beschränkungen aber ersichtlich sind
- Da Daten auch durch Operationen erzeugt werden können und nicht auf einer internen Datenspeicherung beruhen müssen, ist es legitim, Klassen bestehend nur aus Operationen zu bilden

Identifikation von Klassen

Slide 19

- Klassen sollten so gebildet werden, daß sie vollständig sind, z.B. sollte zu einer Einfügeoperation auch eine Entfernenoperation existieren
- die Operationen einer Klasse lassen sich oft in grundlegende Operationen und darauf aufbauend spezialisierte oder vereinfachende Operationen einteilen
- Elemente (Operationen, Daten, Verhalten), welche wiederverwendet werden können, sollten eine Klasse bilden
- Es sollte ein Satz von abstrakten Klassen für allgemeine Anwendungsfälle wie z.B. Liste, Mengen u.ä. existieren bzw. erstellt werden

Der Prozeß der Identifikation von Klassen

Die eigentliche Systementwicklung wird als ein Prozeß beschrieben, in dem der Designer zwei Rollen einnimmt:

Anbieter: als Anbieter sind bestehende Klassen sinnvoll zu kombinieren, um ein bestimmtes Ziel zu erreichen

Slide 20

Kunde: als Kunde werden die zur Erfüllung einer Aufgabe fehlenden Teile abstrakt beschrieben, welche dann wiederum von Anbietern konkretisiert werden müssen

Demnach ist der Entwicklungsprozeß ein ständiger Wechsel zwischen Kunden- und Anbieterrolle, wobei die geschaffenen Abstraktionen möglichst schon auf Wiederverwendbarkeit ausgerichtet sein sollen.

Klassifikation

Klassifikation bezieht sich bei der BON auf die Definition von Vererbungs- und Kunde-Anbieter-Beziehungen. Dabei gibt es mehrere Möglichkeiten:

- Es ist abzuwägen, ob Wiederverwendung durch Vererbung oder durch eine Kunde-Anbieter-Beziehung implementiert wird
- Es können *Mix-in*-Klassen gebildet werden, welche zusätzliche Funktionalitäten implementieren falls von ihnen geerbt wird (Dieser Mechanismus sollte nicht mißbraucht werden, um eine Klasse aus mehreren kleinen Klassen durch Vererbung zusammenzustellen)
- Von Klassen, die nur Operationen anbieten und deren Objekte keinen internen Zustand haben, kann legitim geerbt werden, d.h. sie müssen nicht über eine Client-Beziehung genutzt werden

Slide 21

Gruppierung in Clustern

Bei der Gruppierung von Klassen zu Clustern gibt es eine Reihe von Empfehlungen:

- die Kohäsion der Klassen in einem Cluster sollte hoch sein, während die Kopplung der Cluster untereinander gering sein sollte
- die Schnittstelle zu einem Cluster sollte in einer oder mehrerer Klassen zusammengefaßt werden
- die Klassen eines Clusters sollten auf ähnlichem Abstraktionsniveau liegen (hauptsächlich in der Analysephase, später kann dieses Prinzip abgeschwächt werden)

Slide 22

Definition von Operationen

Bei der Definition von Operationen wird in der BON ein Schwerpunkt auf die Namensgebung gelegt:

Slide 23

- Namen sollten sich möglichst an die natürliche Sprache anlehen
- sie sollen aussagekräftig und nicht übermäßig abgekürzt sein
- Anfragen sollten als Hauptwörter und Aktionen als Verben benannt werden
- Namen sollten konsistent vergeben werden

Zuätzlich sollte die Anzahl der Parameter einer Operation minimiert werden.

Boochs Mikro-Prozeß

Der Mikro-Prozeß von Booch wird zwar der Implementierungsphase untergeordnet, jedoch spricht nichts dagegen, Teile davon z.B. auch während früherer Phasen zu benutzen. Die Bestandteile des Mikro-Prozesses sind:

Slide 24

- Identifikation von Klassen und Objekten
- Identifikation von Klassen- und Objektsemantik
- Identifikation von Beziehungen zwischen Klassen und Objekten
- Spezifikation der Schnittstellen und Implementierung von Klassen und Objekten

Diese Einzelschritte werden bei der Entwicklung zyklisch durchlaufen.

Identifikation von Klassen und Objekten

Die Identifikation von Klassen und Objekten kann auf drei unterschiedliche Arten erfolgen:

- Identifikation von Klassen, die aus der Realwelt entstammen
- Identifikation von Klassen, die bestimmte Funktionalitäten erfüllen sollen
- *Use-Case-Analyse*

Slide 25

Identifikation von Klassen- und Objektsemantik

Die Definition der Semantik von Klassen und Objekten erfolgt durch:

Szenarien: Ausarbeitung von Szenarien, bei denen Systemfunktionen erbracht werden

Slide 26

isoliertes Klassendesign: Beschreibung und Verfeinerung einer Abstraktion durch bestehende und neu zu erstellende untergeordnete Operationen

Identifikation von Mustern: Suche nach Verhaltensmustern, die in unterschiedlichen Situationen wiederverwendet werden können

Identifikation von Beziehungen zwischen Klassen und Objekten

Beziehungen zwischen Klassen und Objekten werden durch drei Aktivitäten bestimmt:

Slide 27 **Definition von Assoziationen:** Zusammenfassen von Klassen gemäß Szenarien oder semantischer Zusammengehörigkeit

Identifikation von Zusammenarbeit: Bilden von Vererbungshierarchien und Zusammenfassen von Klassen in Subsystemen und Modulen

Verfeinerung von Assoziationen: Überarbeitung der Vererbungs- und Zusammenarbeitsstrukturen

Spezifikation der Schnittstellen und Implementierung von Klassen und Objekten

Aufgaben bei der Schnittstellenspezifikation und bei der Implementierung sind:

- Slide 28**
- Beschreibung von Operationssignaturen
 - Festlegung des Sichtbarkeitsbereichs von Operationen
 - Realisierung von Operationen entweder durch Delegation an andere Objekte oder durch direkte Implementierung
 - Auswahl von Implementierungsalgorithmen

Standardaktivitäten bei der OMT

Ähnlich wie bei Booch existieren zwar keine expliziten Standardaktivitäten, es ist jedoch eine Reihe von allgemeiner einsetzbaren Prozessen aufgeführt:

Slide 29

- Identifikation von Klassen und Objekten
- Identifikation von Beziehungen zwischen Klassen
- Identifikation von Attributen
- Erstellung einer Vererbungshierarchie
- Überarbeitung des Modells
- Gruppierung von Klassen in Module

Identifikation von Klassen und Objekten

Als Ausgangspunkt wird empfohlen, Klassen relativ unformal aus dem Anwendungsgebiet abzuleiten und folgendermaßen weiter zu bearbeiten:

Slide 30

- Entfernen von Klassen, die nicht zum Problembereich gehören
- Vermeidung ungenau definierter Klassen
- Klassen, die nur einzelne Objekte beschreiben, sollten nur als Attribute umgesetzt werden
- Eine Klasse sollte keine Rolle sondern ein unterscheidbares Konzept repräsentieren
- Typische Designklassen sollen in der Analyse nicht betrachtet werden

Identifikation von Beziehungen zwischen Klassen

Jede Abhängigkeit zwischen unterschiedlichen Klassen soll als Assoziation modelliert werden, mit folgenden Ausnahmen:

- Implementierungsspezifische Assoziationen
- Aktionen
- Beziehungen zwischen mehr als zwei Klassen sollen aufgeteilt werden
- Assoziationen, die aus anderen Assoziationen abgeleitet werden können

Slide 31

Identifikation von Attributen

Attribute stellen Eigenschaften von Objekten dar, sie sind keine eigenständigen Objekte. Die OMT führt folgende Auswahlkriterien an:

- Attribute sollen keine internen Zustände darstellen
- Attribute, die einer Assoziation zwischen zwei Klassen zugeordnet sind, sollten vermieden werden
- Beschränkung auf wichtige Attribute

Slide 32

Erstellung einer Vererbungshierarchie

Vererbungsbeziehungen werden durch

- Zusammenfassung genereller Konzepte in Oberklassen und
- Verfeinerung spezieller Konzepte in abgeleiteten Klassen

Slide 33 gebildet.

Überarbeitung des Modells

Die Überarbeitung des Modells fügt neue Klassen hinzu oder entfernt überflüssige Klassen. Dieser Prozeß führt oft dazu, daß frühere Phasen erneut durchlaufen werden, weshalb der Ablauf der o.g. Schritte nicht rigoros eingehalten werden muß

Slide 34

Gruppierung von Klassen in Module

Auch bei der OMT wird empfohlen, Kohäsion in den Modulen zu maximieren und Kopplung zwischen den Modulen zu minimieren. Außerdem sollen Module so gebildet werden, daß sie Klassen mit gleichem Abstraktionsgrad enthalten, und abstrakte Module auf konkreten Modulen aufbauen.

Slide 35