

Softwareentwicklung mit UML

Die Unified Modeling Language im Projekteinsatz

Softwareentwicklung mit UML

Übersicht

- 1 Einleitung
- 2 Die Unified Modeling Language (UML)
- 3 Vorgehensmodelle und UML
- 4 Ausblick
 - 4.1 UML 2.0
 - 4.2 Model Driven Architecture (MDA)

Softwareentwicklung mit UML

1 Einleitung

Softwareentwicklung mit UML

Einleitung

Was ist die Unified Modeling Language (UML)?

- UML ist eine grafische Sprache für die objektorientierte Modellierung
- UML geht auf Arbeiten von Rumbaugh, Booch, Jacobsen und anderen zurück.
- UML wurde von der Object Management Group (OMG) standardisiert.
- Für die Modellierung mit UML existieren diverse Produkte für
 - Modellierung (Anforderungen, Architektur)
 - Entwicklung (Codegenerierung, Reverse Engineering)
 - Unterstützte Programmiersprachen: Java, C++, .Net, Ada

Softwareentwicklung mit UML

Die Unified Modeling Language (UML): Diagrammtypen

- Use Case Diagramm
- Aktivitätendiagramm
- Zustandsdiagramm
- Klassendiagramm
- Interaktionsdiagramm
 - Sequenzdiagramm
 - Kollaborationsdiagramm
- Komponentendiagramm
- Deploymentdiagramm

Softwareentwicklung mit UML

Einsatzgebiete für UML

Modellierung von:

- Anforderungen
- Geschäftsprozessen
- Statischer Softwarestruktur
- Dynamischem Verhalten der Software
- Struktur von Softwarekomponenten / Bibliotheken
- Deployment von Software auf Rechnersysteme

Softwareentwicklung mit UML

Einsatzgebiete von UML-Diagrammen

- Anforderungen
Use Case Diagramm, Aktivitätendiagramm
- Geschäftsprozessen
Business Use Case Diagramm, Klassendiagramm
- Statischer Softwarestruktur
Klassendiagramm
- Dynamischem Verhalten der Software
Sequenzdiagramm, Zustandsdiagramm
- Struktur von Softwarekomponenten / Bibliotheken
Komponentendiagramm
- Deployment von Software auf Rechnersysteme
Deploymentdiagramm

Softwareentwicklung mit UML

Kommunikation mit UML

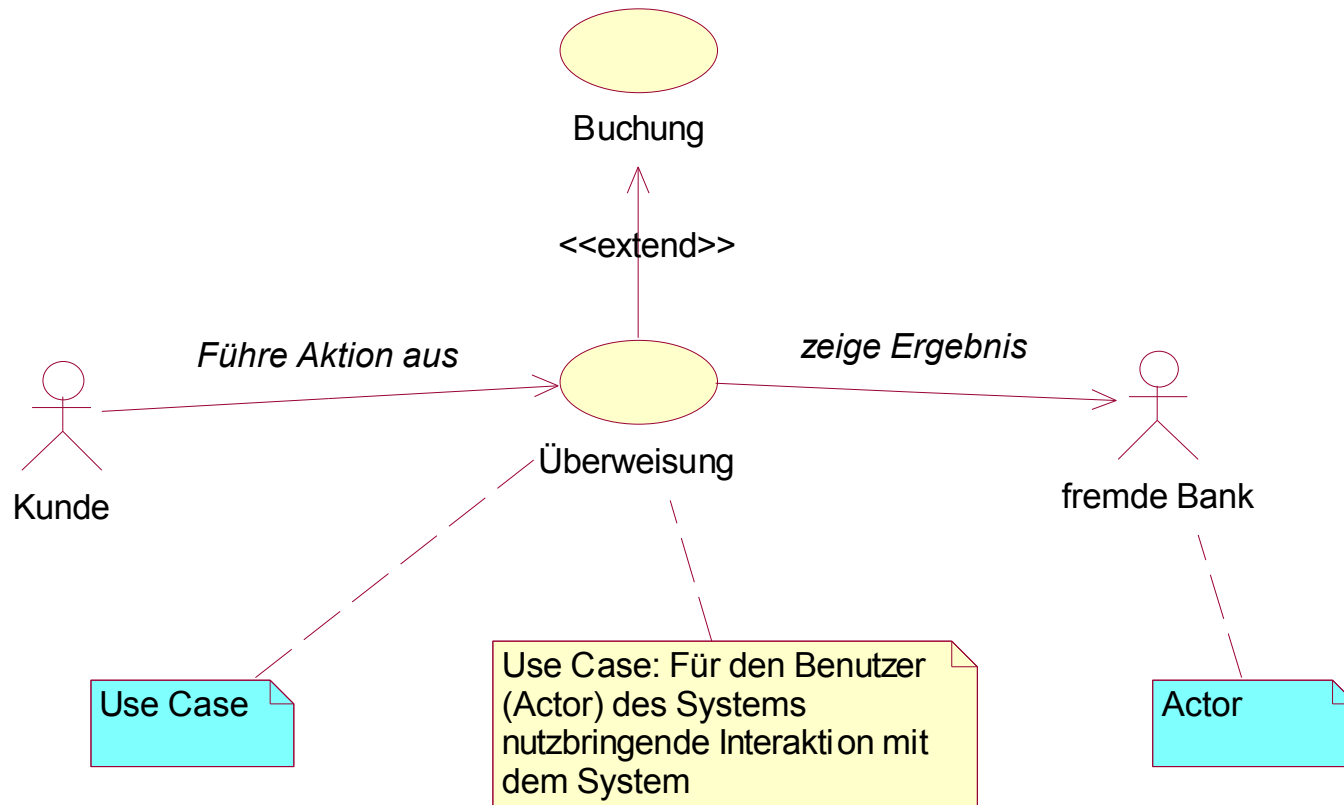
UML als einheitliche Sprache der an der Softwareentwicklung beteiligten Parteien.

- Kunde
- Business Engineer
- Lieferanten von Softwarekomponenten
- Softwarearchitekten und -entwickler
- Tester

2 Die Unified Modeling Language (UML)

Softwareentwicklung mit UML

Die Unified Modeling Language (UML): Use Case Diagramm



Softwareentwicklung mit UML

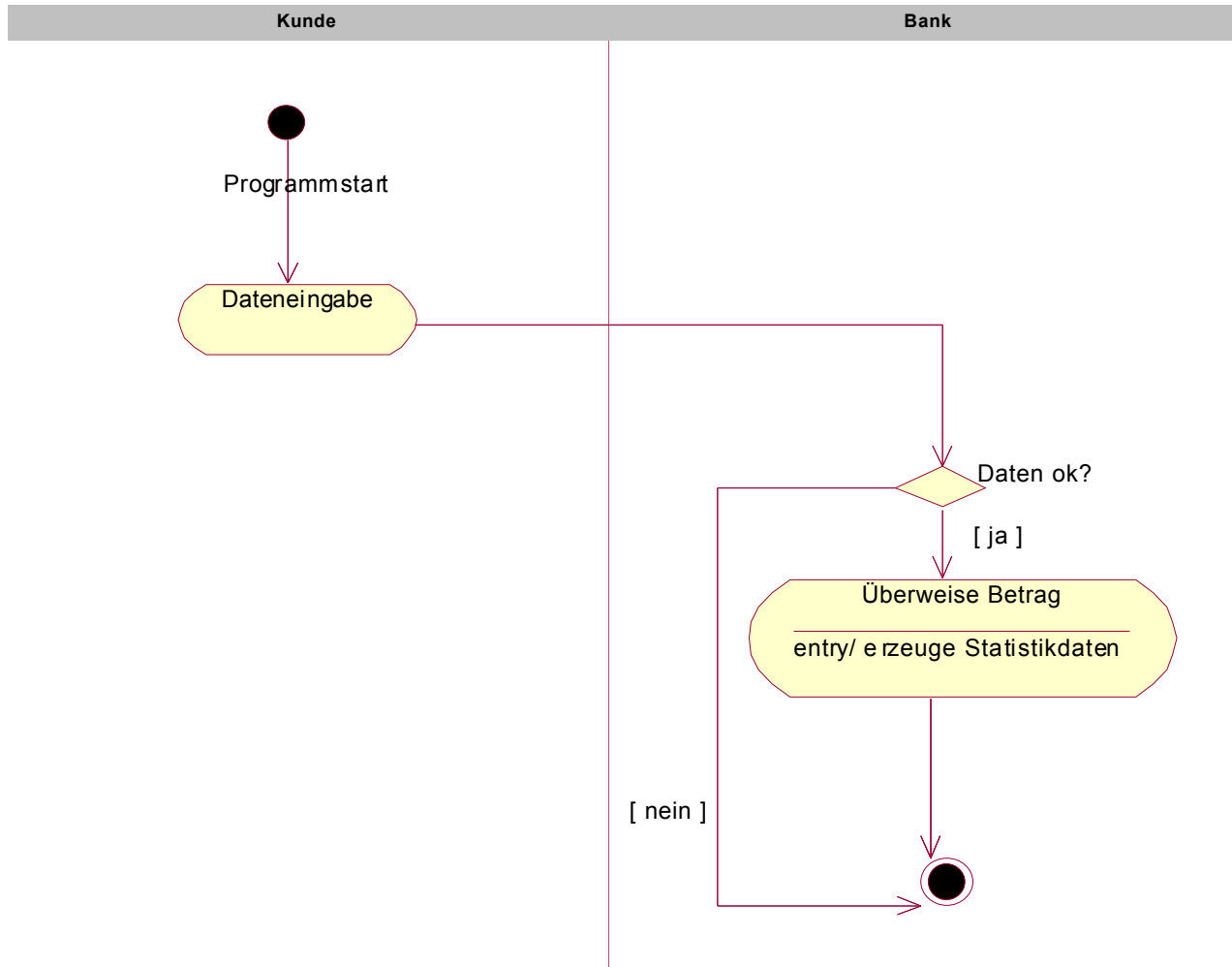
Die Unified Modeling Language (UML): Use Cases

Use Case Diagramme sind nicht sehr aussagekräftig

- Use Cases werden in Prosa nach vorgegebener Struktur beschrieben. Für jeden Use Case werden
 - Name
 - Vorbedingung, Nachbedingung
 - Hauptszenario
 - (wichtige) weitere Szenarien
 - Zugeordnete Aktivitätsdiagramme und Szenarien

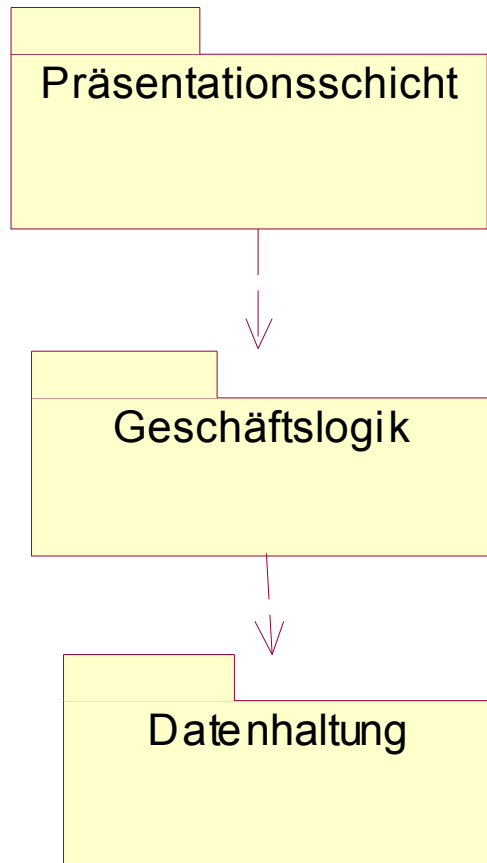
Softwareentwicklung mit UML

Die Unified Modeling Language (UML): Aktivitätendiagramm



Softwareentwicklung mit UML

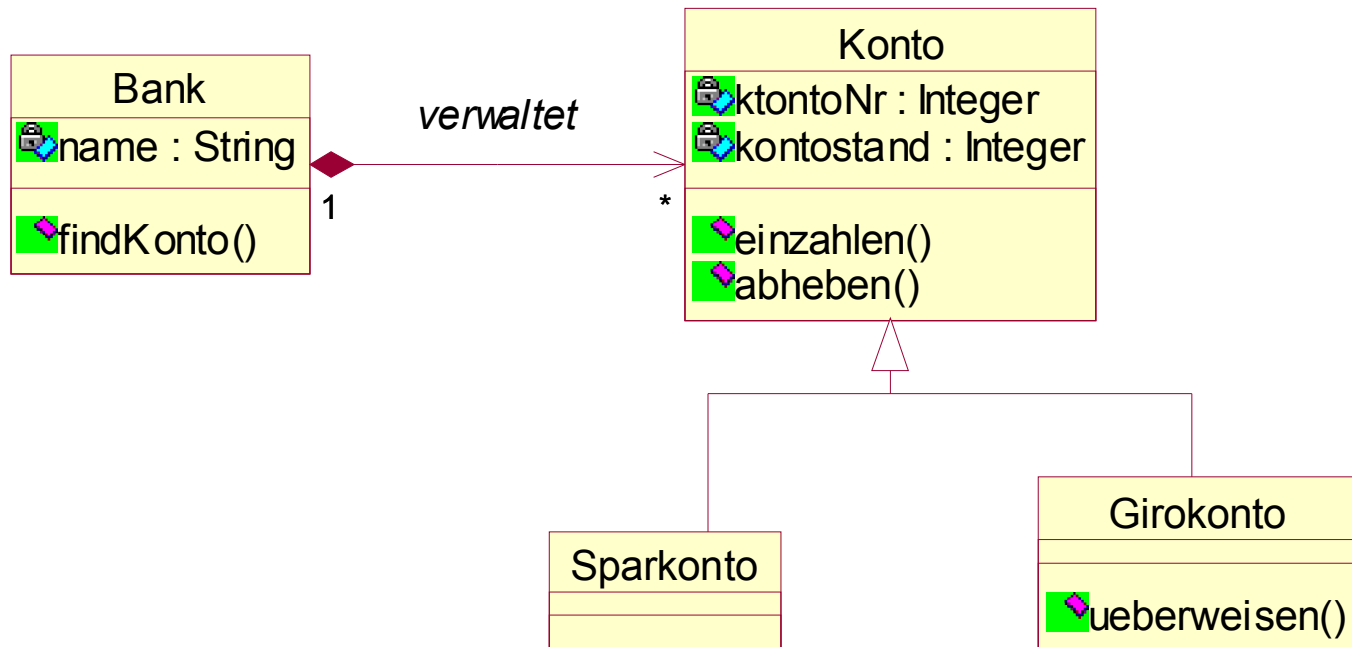
Die Unified Modeling Language (UML): Klassendiagramm: Packages



Softwareentwicklung mit UML

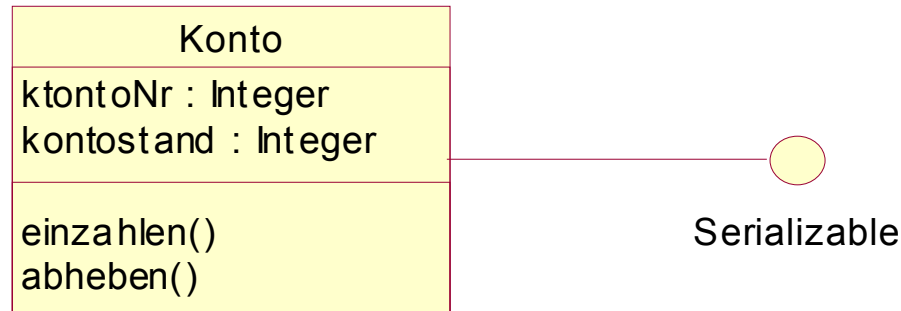
Die Unified Modeling Language (UML):

Klassendiagramm: Klassen



Softwareentwicklung mit UML

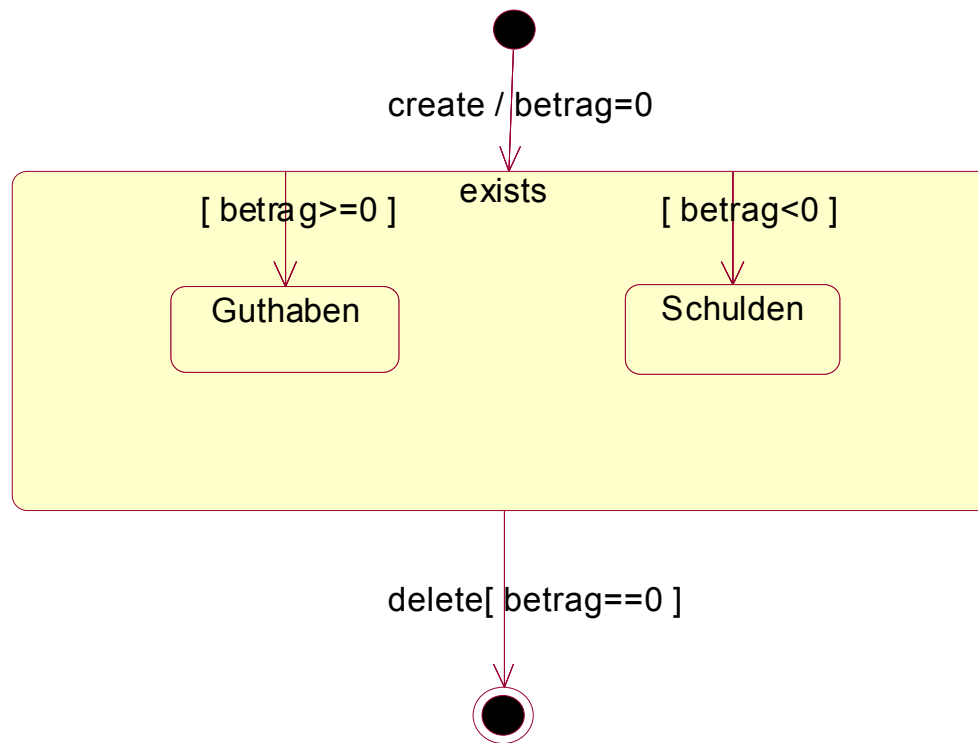
Die Unified Modeling Language (UML): Klassendiagramm: Klassen



Softwareentwicklung mit UML

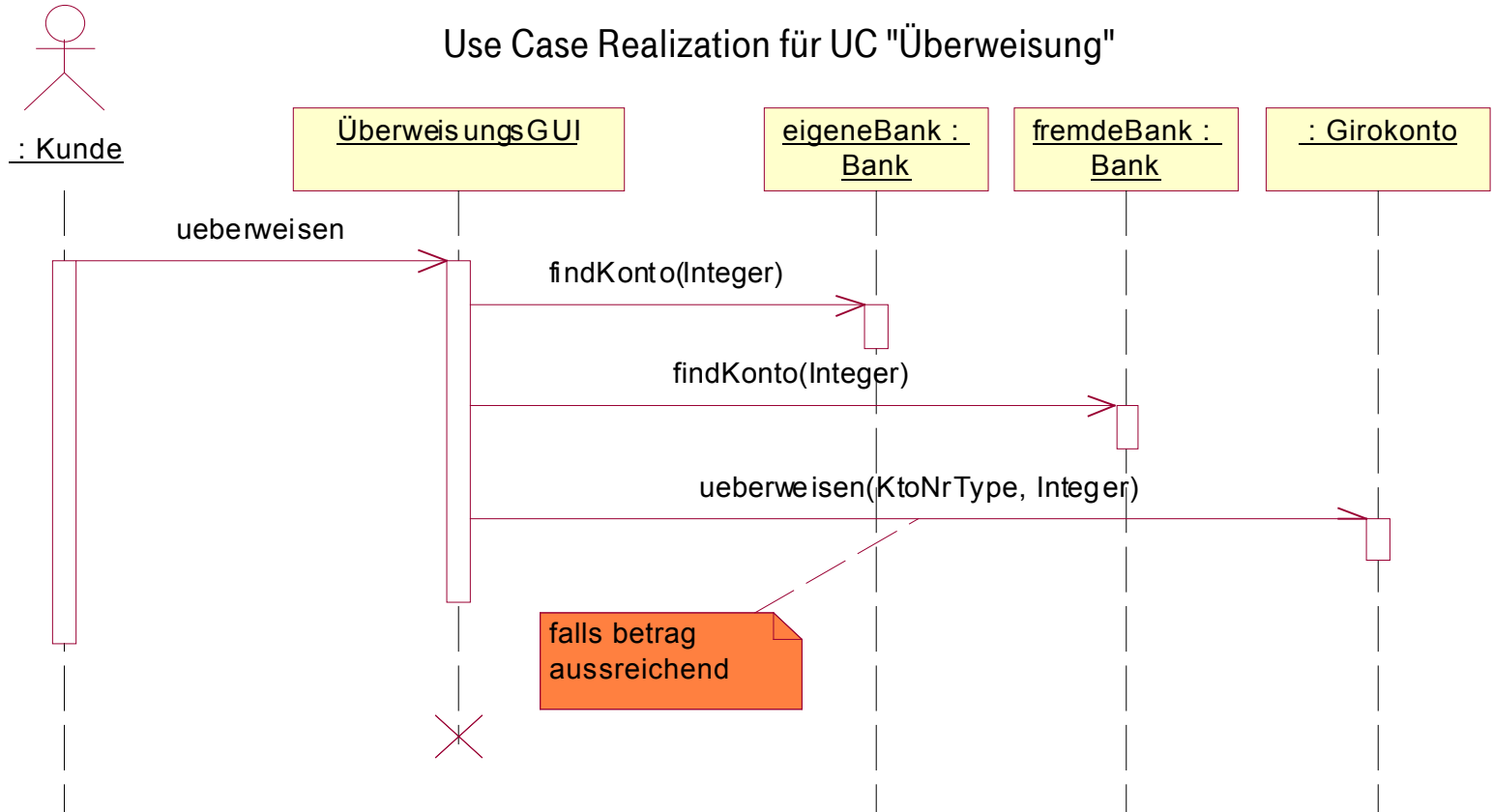
Die Unified Modeling Language (UML): Zustandsdiagramm

Zustandsdiagramm für Klasse Konto



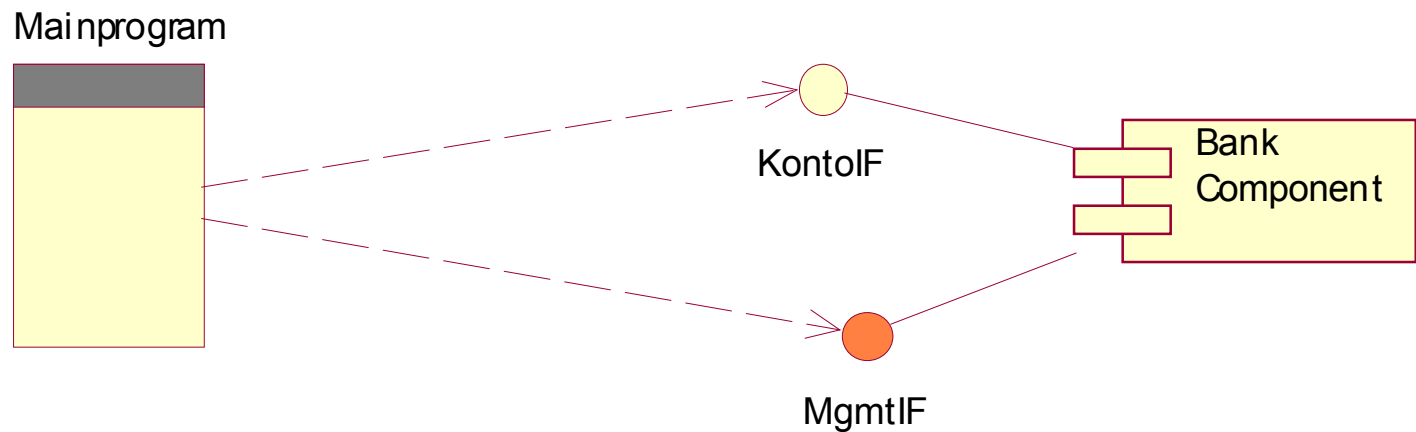
Softwareentwicklung mit UML

Die Unified Modeling Language (UML): Sequenzdiagramm



Softwareentwicklung mit UML

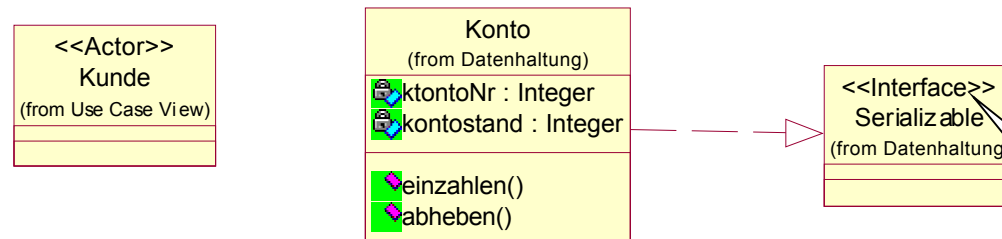
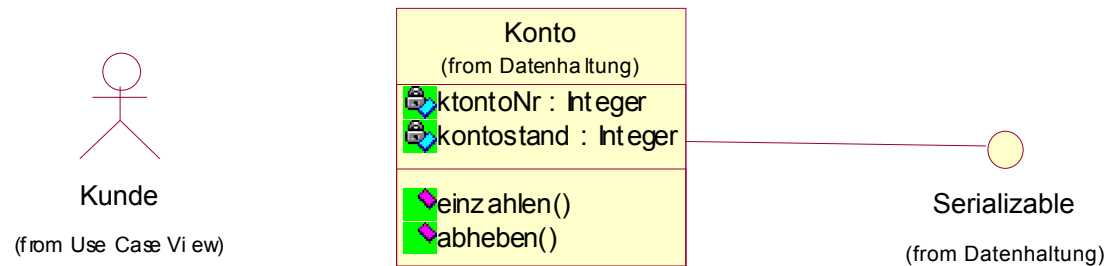
Die Unified Modeling Language (UML): Komponentendiagramm



Softwareentwicklung mit UML

Die Unified Modeling Language (UML): Stereotypes

- UML ist erweiterbar durch Stereotypes



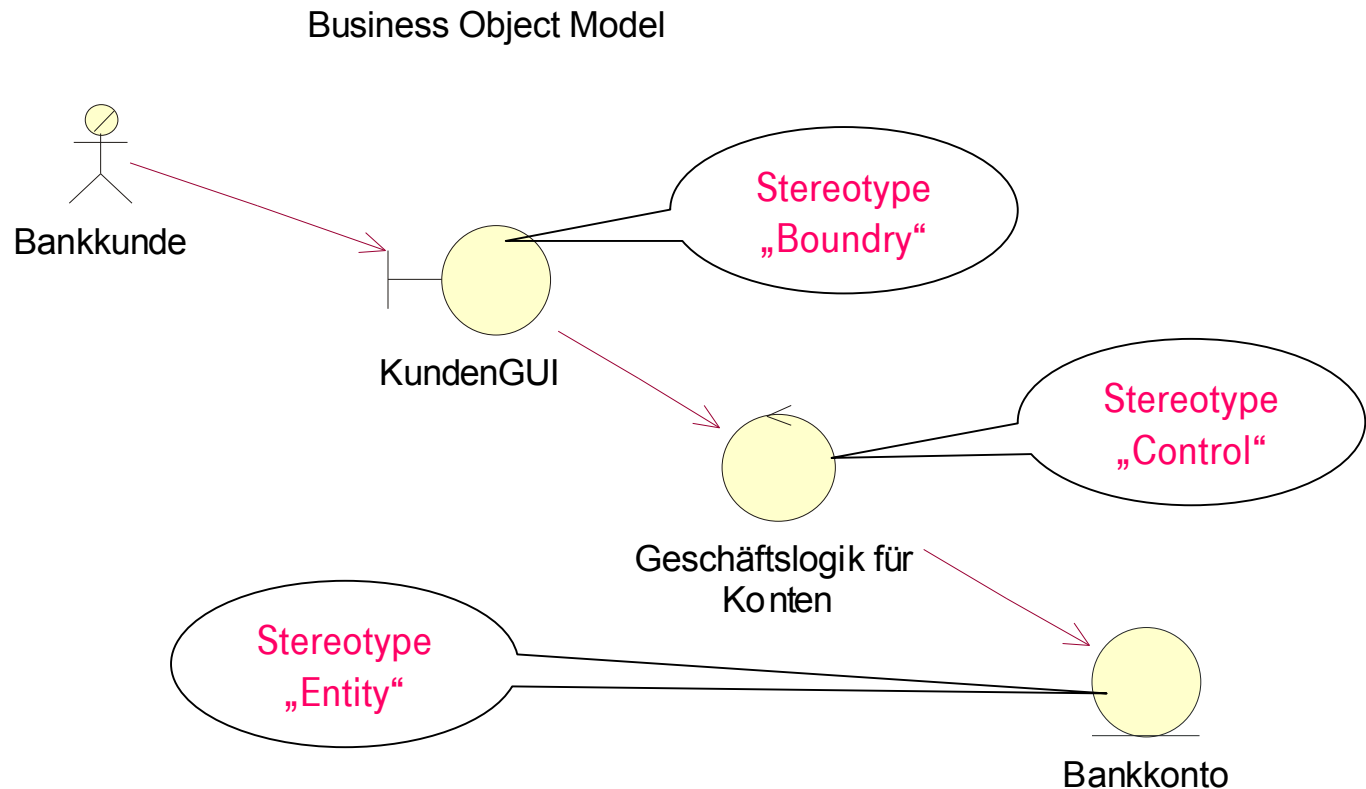
Stereotype



Softwareentwicklung mit UML

Die Unified Modeling Language (UML): Stereotypes: Geschäftsprozessmodellierung

- UML hier nicht direkt für Softwareentwicklung

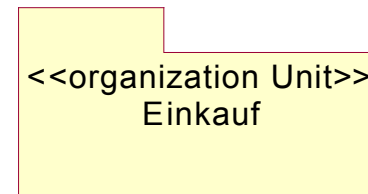
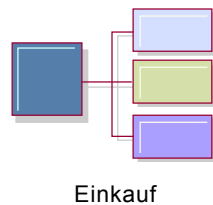


Softwareentwicklung mit UML

Die Unified Modeling Language (UML): Stereotypes

- Stereotypes erweitern UML
- Die dadurch definierte neue Semantik sollte klar definiert sein
- Leser des Modells verstehen nicht automatisch die Bedeutung eines neuen Stereotypes

Beispiel:



- Codegenerierung für einen neuen Stereotype muss selbst geschrieben werden, dadurch sind allerdings mächtige Erweiterungen möglich

Softwareentwicklung mit UML

Die Unified Modeling Language (UML): Weitere Eigenschaften von UML Tools

- Codegenerierung (Klassenrumpfe) für Java, C++ aus Klassenmodellen
- Reverse Engineering (Klassendiagramme aus Java-Code)
- Erweiterung der Funktionalität des UML-Tools durch Scriptsprachen (Bsp: Generierung von Zustandsautomaten im Quellcode aus Zustandsdiagrammen)
- Dokumentationserzeugung (z.B. UML in Word);
(Problem: Modell und Dokument konsistent halten)
- Requirements Engineering (Zusatztools)

3 Vorgehensmodelle und UML

Softwareentwicklung mit UML

Vorgehensmodelle und UML

- UML ist eine Modellierungssprache.
- In UML werden keine Vorgaben über den Entwicklungsprozess gemacht, mit dem ein Modell erstellt wird.
- werden i.a. im Rahmen eines Vorgehensmodells erstellt.

Der Prozess definiert

- Rollen
- Artefakte
- Aktivitäten
- und deren Zusammenarbeit

Softwareentwicklung mit UML

Vorgehensmodelle und UML

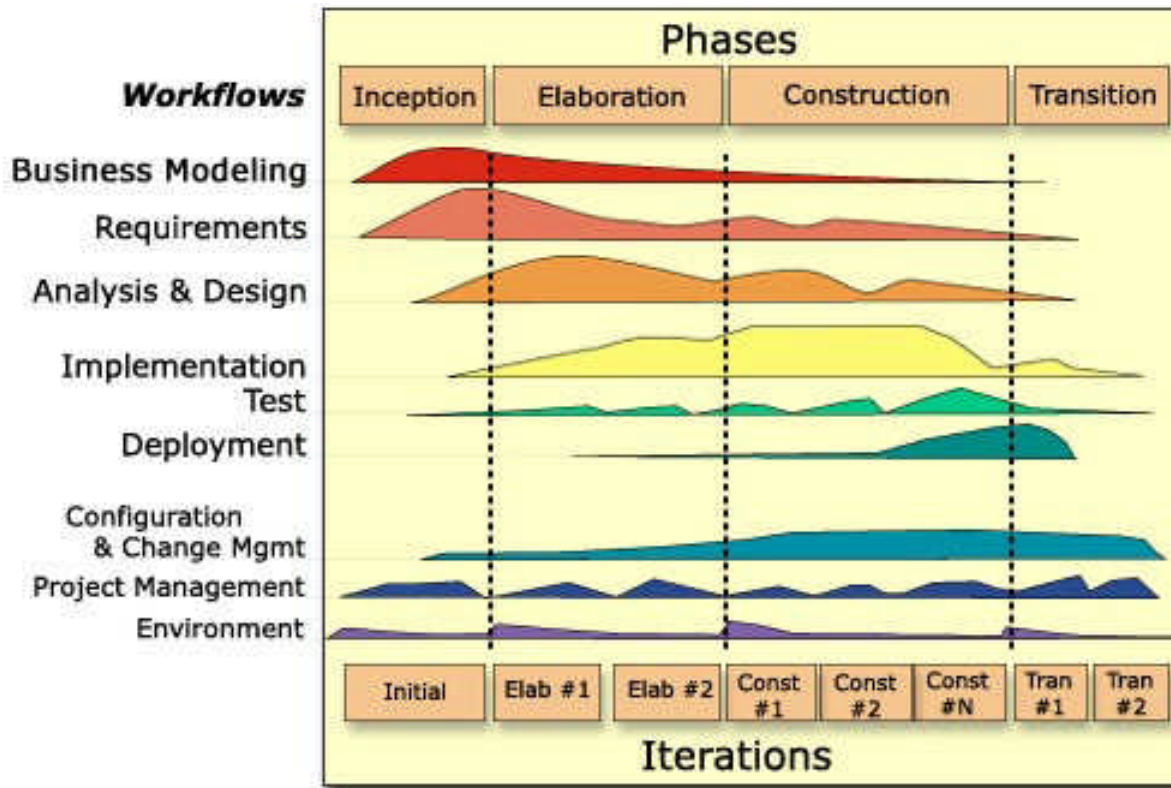
Agile Methoden versus Unified Process

- Agile Methoden sind auch für die Modellierung möglich
 - Schlankes Vorgehensmodell
 - Wenig Formalismus
 - Modellierung wenn möglich, nur an der Tafel, nicht mit einem Tool
- Unified Process
 - Mächtiges Prozessframework, dass für eigene Projekte „getailort“ (konfiguriert) werden muss
 - Auch (oder gerade) für große Projekte einsetzbar
 - Gefahr, den Prozess zu komplex zu machen

Softwareentwicklung mit UML

Vorgehensmodelle und UML

■ Der Rational Unified Process (RUP)



Softwareentwicklung mit UML

Vorgehensmodelle und UML

UML und der Unified Process

UML in den Phasen des RUP:

- Requirements Engineering
 - Use Cases, Aktivitäten

- Geschäftsprozessmodellierung
 - Klassen, Objekte, Interaktionen

- Analyse, Design
 - Use Cases, Aktivitäten, Klassen, Interaktionen, ...

- Implementierung
 - Klassen, Interaktionen, Zustände, Komponenten

- Test
 - Prüfen der Abbildung von Use Cases auf Klassen

Softwareentwicklung mit UML

Vorgehensmodelle und UML

UML: Wunsch und Wirklichkeit

- UML ermöglicht die problemlose Kommunikation der "Stakeholder"
.Nicht alle Stakeholder beherrschen UML.
- UML spezifiziert ein Softwaresystem eindeutig. Häufig sind nur Diagramme, aus denen Code generiert wird, oder sie dem Kunden kommuniziert werden, aktuell.
- UML macht die Softwareentwicklung produktiver. Investitionen in die Konfiguration der Tools und Styleguides sind erforderlich. Es muss genau festgelegt sein, welche Artefakte mit einem Tool bearbeitet und gepflegt werden.

Softwareentwicklung mit UML

4 Ausblick

Softwareentwicklung mit UML

UML 2.0

Kritikpunkte an UML 1.4

- Semantik einiger Diagramme nicht präzise und eindeutig genug (z.B. für die Codegenerierung)
- Unterschiedliche Diagramme für ähnliche Aufgaben
- Austauschbarkeit von Modellen zwischen Tools verschiedener Hersteller verbesserungsbedürftig

Softwareentwicklung mit UML

4.1 UML 2.0

Softwareentwicklung mit UML

UML 2.0

- Klare Konzepte
- Neues Metamodell

- Wenig Neuigkeiten bei Strukturdiagrammen, außer
 - Kompositionsstrukturdiagramm: zeigt
 - die Zusammenarbeit von Klassen (über „Ports“)
 - Kollaboration
 - Deploymentdiagramm deutlich erweitert

- Verhaltensdiagramme stark erweitert:
 - Klare Semantik
 - Aktivitätsdiagramm faktisch neu
 - Interaktionsdiagramme:
 - Kontrollflusselemente (if... then...; loop...)
 - Timingdiagramm: neu!

Softwareentwicklung mit UML

UML 2.0

- Datenaustausch zwischen Modellierungstool funktioniert durch XMI (XML Metadata Interchange)
- OCL (Object Constraint Language) erlaubt die Formale Definition von Bedingungen, Invarianten

(Bereits in UML 1.4 vorhanden, aber jetzt überarbeitet)

4.2 Model Driven Architecture (MDA)

Softwareentwicklung mit UML

Model Driven Architecture (MDA)

- Ziel: Die Modellierung Geschäftsmodellen unabhängig von
 - Middleware (CORBA, EJB, .NET)
 - Programmiersprache
- Ziel: Das Modell enthält eine formale Definition der Geschäftslogik
- Das Geschäftsmodell wird in UML als **Platform-Independent Model (PIM)** repräsentiert
- Aus dem PIM wird über Mappings für konkrete Technologien (wie EJB, .NET) ein (oder mehrere) **Platform-Specific Model (PSM)** generiert. Ein PSM wird als UML Modell repräsentiert.

Softwareentwicklung mit UML

Model Driven Architecture (MDA)

- Das PIM muss manuell annotiert werden, um alle für das Mapping in ein PSM notwendigen Informationen zur Verfügung zu haben

Softwareentwicklung mit UML

Model Driven Architecture (MDA)

- MDA ist ein vielversprechender Ansatz.
- Beträchtlicher Einarbeitungsaufwand erforderlich
- Frage: Ist der Wechsel der Technologien so schnell, dass MDA sinnvoll ist

Softwareentwicklung mit UML

Vielen Dank für Ihre Aufmerksamkeit!

Kontakt:

T-Systems Service Line Systems Integration
Business Unit Entwicklungszentrum Nord
Netz- und Servicemanagement I
Ulrich Witte
Willy-Brandt-Platz 3
28215 Bremen

Tel: (421) 3799-446

ulrich.witte@t-systems.com

Softwareentwicklung mit UML

Quellen zu UML

- www.uml.org der OMG (www.omg.org)
- www.jeckle.de mehrere Vorträge zu UML 2 (und vieles andere)
- www.cetus-links.com Linksammlung zu OO
- www.rational.com/uml/index.jsp UML Ressource Center
- www.togethersoft.com die Konkurrenz von Borland
- www.telelogic.com UML 2.0 basiertes Tool
- argouml.tigris.org Open Source Tool
- www.gentleware.com Kommerzielle Version von ArgoUML
- www.oose.de/oep Vorgehensmodell von Bernd Oestereich
- www.ambysoft.com Scott Ambler über OO