

Generische Bearbeitung von Objektgraphen

Bedienkonzepte und technische Fallstricke

W3L AG
info@W3L.de

2013



Agenda

■ Motivation

- Modellgetriebene Softwareentwicklung
- Entwurfsziele für eine MDSD-Lösung

■ Beispielmodell

■ Fallstricke

- Präsentationsschicht
- Anwendungsschicht
- Persistenzschicht

■ Fazit



Motivation

■ Modellgetriebene Softwareentwicklung (MDSD)

- Aus formalen Modellen wird lauffähige Software erzeugt
- Abstraktion von technischen Details
 - Softwareschichten werden automatisch bereitgestellt bzw. generiert
 - Präsentationsschicht, Anwendungsschicht und Persistenzschicht
 - OR-Mapping, Sitzungsverwaltung, Objektvalidierung, Services [...]

■ Verschiedene MDSD-Lösungen sind vorhanden

- W3L.CoreFrame von der W3L AG
- JANUS von der Otris Software AG
- eXpress App Framework von DevExpress
- LightSwitch von Microsoft
- Eclipse Modeling Framework (EMF) von der Eclipse Foundation

Motivation

■ Modellgetriebene Softwareentwicklung (MDSD)

■ Vorteile

- Steigerung der Entwicklungsgeschwindigkeit (Rapid-Prototyping)
- Leichte Erlernbarkeit durch ein hohes Abstraktionsniveau
- Konzentration auf die Fachlogik / Wenig Infrastruktur-Code

■ Nachteile (je nach MDSD-Lösung unterschiedlich)

- Zu geringe Flexibilität und Anpassbarkeit
 - Die Benutzeroberfläche verhält sich stets nach einem festen Schema
 - Wenig Spielraum für Anpassungen am Standardverhalten
 - Zu viele Einschränkung bei der Entwicklung

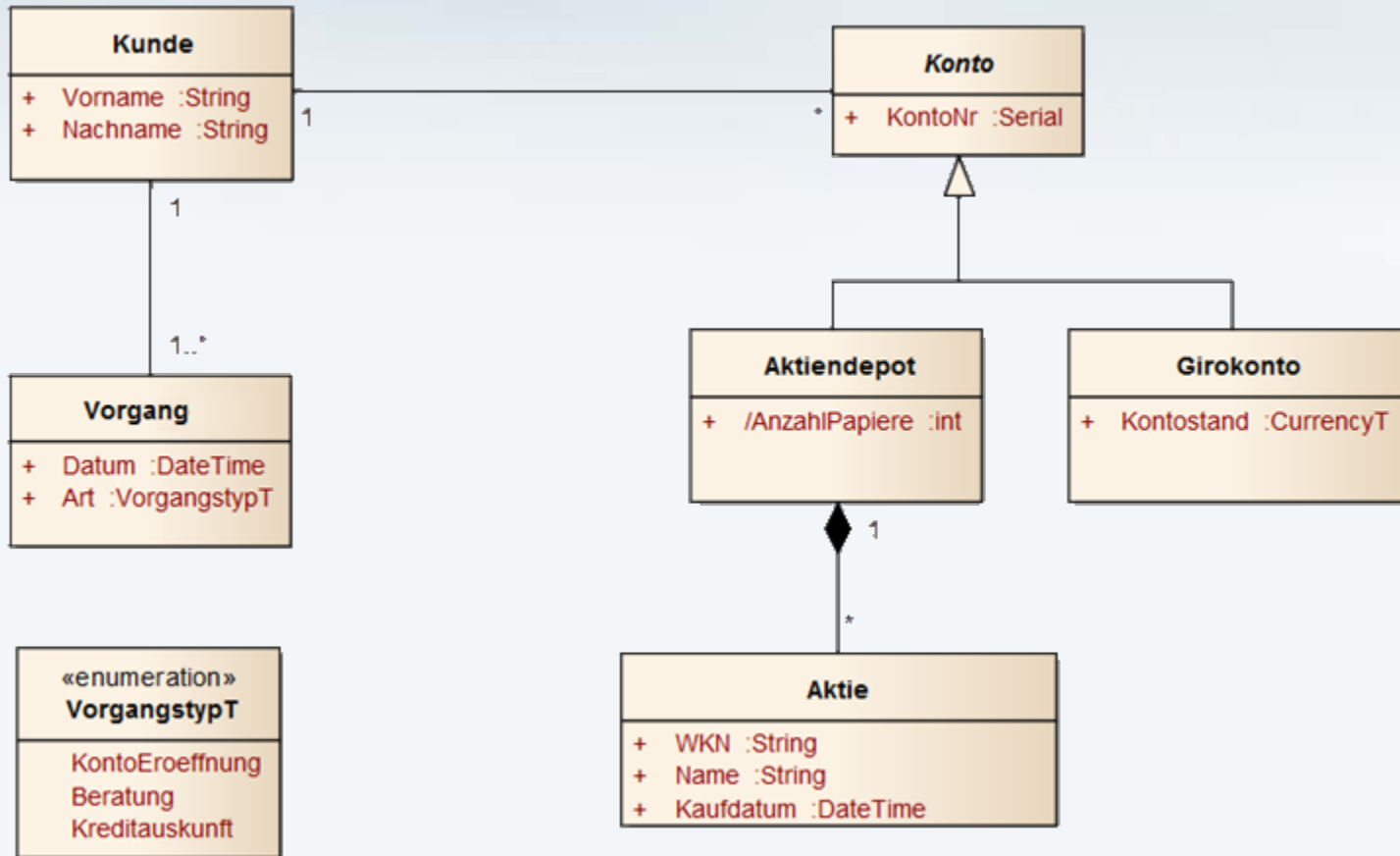
**Eine gute MDSD-Lösung muss eine hohe Abstraktion bieten
und zeitgleich flexibel und anpassbar sein**

Entwurfsziele für eine MDSD-Lösung

■ Hohe Abstraktion, Flexibilität und Anpassbarkeit

- Anspruchsvolles Ziel für ein Rahmenwerk zur Software-Entwicklung
- Viele technische Fallstricke sind zu lösen
 - Präsentationsschicht
 - Ein umfangreiches und flexibles Bedienkonzept zur Datenverwaltung wird benötigt
 - *Schnittstellen und Erweiterungspunkte zur Anpassung des Standardverhaltens*
 - Anwendungsschicht
 - Umgang mit verschiedenen Transaktionen und Benutzersitzungen
 - Validierung: Die Gültigkeit des Objektgraphs muss sichergestellt werden
 - *Schnittstellen und Erweiterungspunkte zur Anpassung des Standardverhaltens*
 - Persistenzschicht
 - Automatisiertes OR-Mapping
 - Referenzielle Integrität und NULL-Marken
 - *Schnittstellen und Erweiterungspunkte zur Anpassung des Standardverhaltens*

Beispielmodell



Fallstricke: Präsentationsschicht

■ Ziele

- Eine modellierte Anwendung soll nach der Generierung der Fachkonzeptklassen ausführbar sein
- Eine rudimentäre Datenerfassung muss möglich sein

■ Mögliche Vorgehensweisen

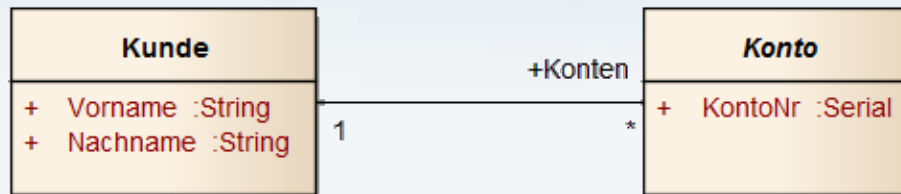
- „Klassischer Ansatz“: Das Generatorsystem generiert ebenfalls die GUI
 - Performant, aber sehr statisch
- „Moderner Ansatz“: Eine generische UI-Engine wird vom MDSD-Framework bereitgestellt
 - Erzeugung der Eingabedialoge erfolgt dynamisch auf Basis von Metainformationen
 - Anpassungen sind zur Laufzeit durch Veränderung der Metainformationen möglich

■ Unabhängig von der Vorgehensweise, wird ein „universelles“ Bedienkonzept zur Datenverwaltung benötigt

Fallstricke: Präsentationsschicht

■ „universelles“ Bedienkonzept zur Datenverwaltung

- Wie kann ein möglicher Dialog zur Erfassung eines Kunden aussehen?



■ Benötigte GUI-Komponenten

- Zwei Textfelder für die Erfassung von Vorname und Nachname
- Eine Datentabelle für die Darstellung der zu-N-Assoziation auf „Konto“

■ Welche Interaktionen sind innerhalb des Kundendialogs möglich?

- Datensatz Kunde: Änderungen speichern, Datenerfassung abbrechen
- Mögliche Aktion für die Assoziation „Konten“
 - Neues Konto anlegen, bestehendes Konto verbinden, verbundenes Konto bearbeiten, verbundenes Konto trennen, verbundenes Konto löschen

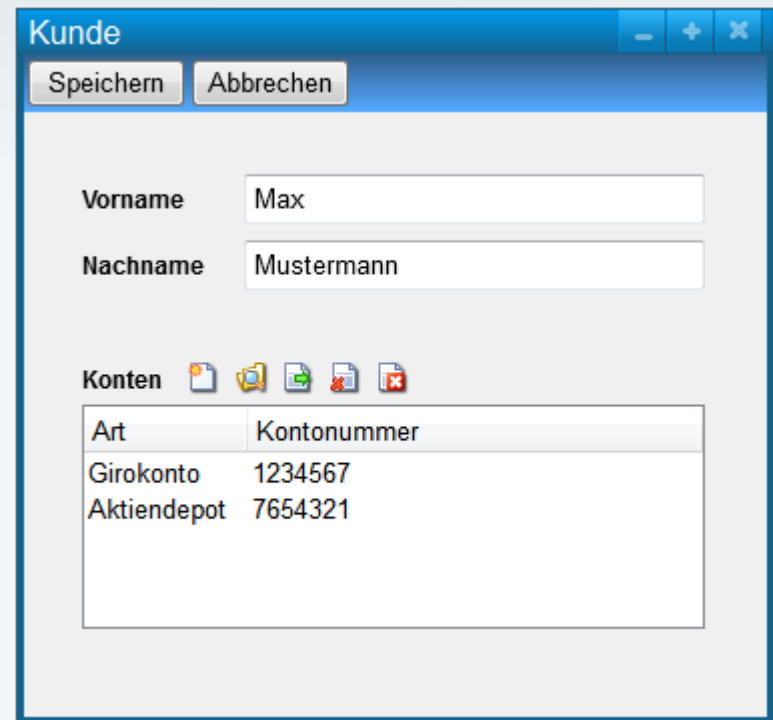
Fallstricke: Präsentationsschicht

■ „universelles“ Bedienkonzept

- Die Aktionen für den Datensatz „Kunde“ sind in einer Tool-Bar untergebracht
- Die Aktionen für die Assoziation „Konten“ werden über Piktogramme dargestellt

■ Offene Fragen

- Was passiert, wenn ein neues *Konto* angelegt wird?
 - Typabfrage
- Welche Anpassungsmöglichkeiten werden angeboten?



Art	Kontonummer
Girokonto	1234567
Aktiendepot	7654321

Fallstricke: Präsentationsschicht

■ Anpassungsmöglichkeiten der GUI

- Verschiedene Layoutlogiken: Zwei-Spalten-Layout
 - Vorname und Nachname in einer Zeile
- Gruppierung von Eingabeelementen in einzelne Bereiche
 - Gruppenrahmen (GroupBox)
 - Registerkarten (Tabs)
 - Reduzierbarer Bereich (Expander)
- Austausch von Eingabeelementen
 - Benutzerdefinierte Darstellung einer Assoziation
 - Datumseingabe über eine Datumsauswahl statt eines einfachen Textfelds

The screenshot shows a window titled "Kunde" with a blue header bar. Below the header are two buttons: "Speichern" and "Abbrechen". The main content area is divided into two sections: "Allgemein" and "Zuordnungen".

Allgemein

Vorname: Max Nachname: Mustermann

Zuordnungen

Konten

Kontonummer: 1234567 Kontostand: 2.000 €	GIRO	Kontonummer: 7654321 Anzahl der Wertpapiere: 400	DEPOT
Kontonummer: 9993331 Kontostand: 2.200 €	GIRO	Kontonummer: 3339991 Anzahl der Wertpapiere: 500	DEPOT

Fallstricke: Anwendungsschicht

■ Umgang mit verschiedenen Transaktionen und Benutzersitzungen

- Jeder Datensatz soll innerhalb einer Transaktion bearbeitet werden
 - Die Aktion „Speichern“ soll alle Änderungen persistieren (Commit)
 - Die Aktion „Abbrechen“ soll alle Änderungen verwerfen (Rollback)
- Einsatz eines Unit-of-Work-Containers: Protokollierung der Änderungen
 - Ein Unit-of-Work-Container beinhaltet alle Änderungen (= Hibernate.Session)
 - Rollback: Alle Änderungen im Container werden verworfen
 - Commit: Die Änderungen im Container werden im Rahmen einer Datenbank-Transaktion zurück in die DB geschrieben
 - Varianten
 1. Änderungen werden erst bei einem Commit geschrieben
 - Nachteil: Neue DB-Zugriffe liefern alte Daten
 2. Änderungen werden automatisch geschrieben, wenn innerhalb der gleichen UoW auf geänderte Datensätze zugegriffen wird
 - Vorteil: Kein Lesen von veralteten Daten
 - Nachteil: Langläufige DB-Transaktion

Fallstricke: Anwendungsschicht

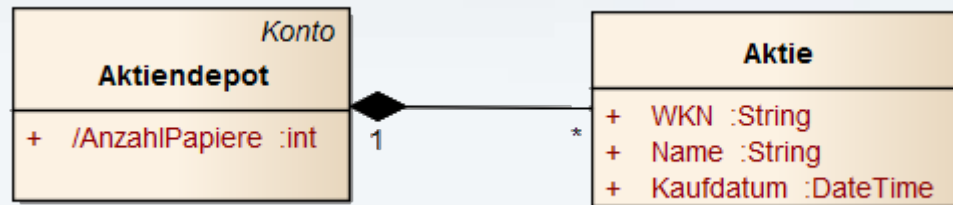
■ Umgang mit verschiedenen Transaktionen und Benutzersitzungen

- Szenario: Die generierte Anwendung ist eine Mehrbenutzeranwendung
 - Was passiert, wenn zwei Benutzer zeitgleich einen Datensatz bearbeiten möchten?
- Vorgehensweisen
 - Einsatz eines pessimistischen Sperrverfahrens
 - Ein zweiter Benutzer kann nur lesend auf den Datensatz zugreifen
 - Einsatz eines optimistischen Sperrverfahrens
 - Ein zweiter Benutzer darf den Datensatz ebenfalls bearbeiten
 - Konfliktlösungen
 - Das gleiche Eingabefeld wird bearbeitet → Wer zuletzt speichert, gewinnt!
 - Unterschiedliche Eingabefelder werden bearbeitet → Beide Änderungen werden übernommen
- Art des Sperrverfahrens sollte in Abhängigkeit zur Geschäftsdomäne konfigurierbar sein

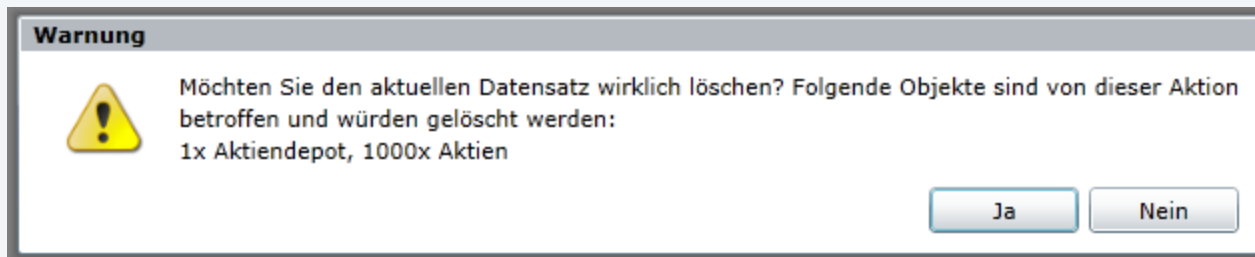
Fallstricke: Anwendungsschicht

Validierung: Gültigkeit des Objektgraphs

- Szenario: Ein Aktiendepot wird gelöscht. Für die Aktien ist die Relation mit einem Aktiendepot verpflichtend.



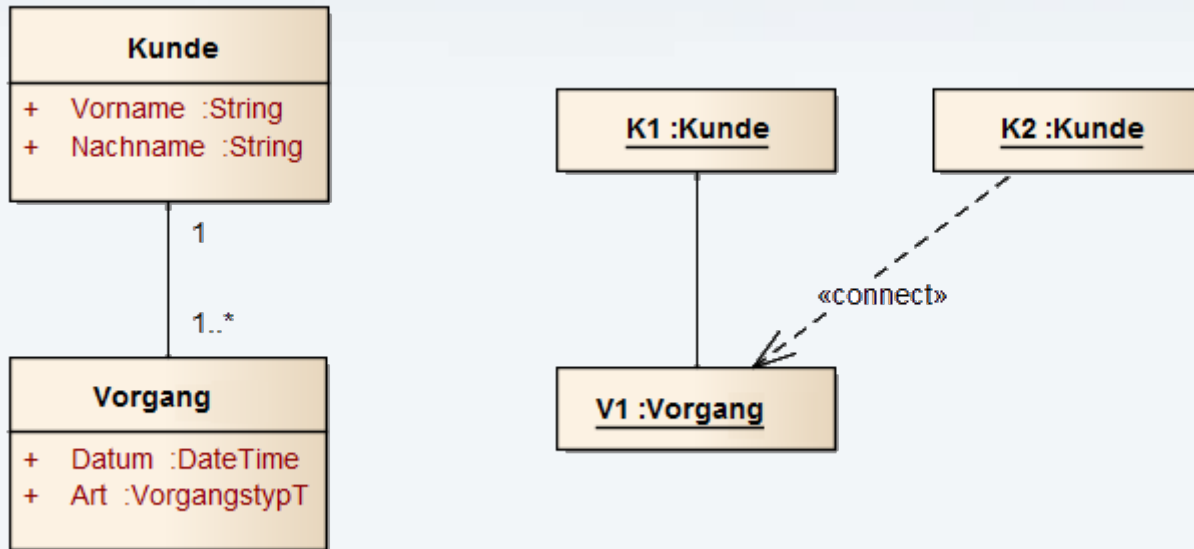
- Damit das System valide bleibt, müssen die Objekte vom Typ „Aktie“ gelöscht oder zuerst einem anderen Depot zugeordnet werden
- Objektgraph-Validierung
 - Eine MDSD-Lösung sollte solche Situationen erkennen und entsprechend behandeln



Fallstricke: Anwendungsschicht

Validierung: Gültigkeit des Objektgraphs

- Szenario: Ein bestehender Vorgang „V1“ wird einem anderen Kunden „K2“ zugeordnet. Der Kunde „K1“ wird durch diese Aktion invalide.



- Solche Situationen müssen ebenfalls erkannt werden
 - In diesem Fall sollte diese Benutzeraktion mit einer Fehlermeldung unterdrückt werden

Fallstricke: Persistenzschicht

■ Automatisiertes OR-Mapping

- Nach dem Modell-First-Ansatz werden das Datenbank-Schema und die benötigten OR-Mapping-Informationen automatisch generiert
- Abbildung von Vererbungshierarchien
 - Für die Abbildung von Vererbungshierarchien auf ein Datenbankschema existieren verschiedene Vorgehensweisen
 - Tabelle pro Vererbungshierarchie (viele leere Felder, keine Join-Operation)
 - Tabelle pro Unterklasse (Join-Operation nötig)
 - Tabelle pro konkrete Klasse (Selektionen auf Oberklassen sind aufwendiger)
 - Es sollte konfigurierbar sein, welche Strategie zur Abbildung einer Vererbungshierarchien verwendet wird (Optimierungsmöglichkeit)
 - Zur Abstraktion sollte „Tabelle pro konkrete Klasse“ die Standardvorgehensweise sein

Fallstricke: Persistenzschicht

■ Referentielle Integrität

- Konzept dient zur Sicherung der Datenintegrität
- Fremdschlüssel-Beziehungen müssen immer gültig sein
 - Nach der RI-Regel dürfen Datensätze nur auf existierende Datensätze verweisen
- Eine MDSD-Lösung sollte die RI-Regel stets berücksichtigen
- Eine kaskadierende Löscho- oder Änderungsweitergabe hingegen sollte nicht verwendet werden
 - Die Anwendungsschicht muss solche Vorgängen kontrollieren können
 - Auswirkungen von Löschvorgängen müssen erkennbar sein

■ Pflichtfelder auf der Datenbankebene

- Die NULL-Marke sollte für alle Datenbank-Felder zulässig sein
- Ermöglicht die Persistierung von „ungültigen“ bzw. „halbfertigen“ Objekten
- Einsatzszenario
 - Schrittweise Erfassung eines Datensatzes über einen Workflow-Prozess

Fazit

- Die Implementierung von MDSD-Lösungen ist kompliziert
- Die Menge der möglichen Eingabemodelle für den Generator ist unbegrenzt
 - Mit Vererbungshierarchien und allen Arten von Assoziation muss umgegangen werden können
- Eine MDSD-Lösung sollte keine statische Präsentationsschicht generieren
 - Generische GUIs sind deutlich flexibler
 - Möglichkeiten zur Anpassung und Konfiguration der GUI sind ausschlaggebend für die Akzeptanz einer MDSD-Lösung
- Schnittstellen und Erweiterungspunkte zur Anpassung des Standardverhaltens werden in allen Softwareschichten benötigt

Vielen Dank für Ihre Aufmerksamkeit!

Inhouse-Schulungen



Wir bieten Inhouse-Schulungen und Beratung durch unsere IT-Experten und -Berater.

Schulungsthemen

- Softwarearchitektur (OOD)
- Requirements Engineering (OOA)
- Nebenläufige & verteilte Programmierung

Gerne konzipieren wir auch eine individuelle Schulung zu Ihren Fragestellungen.



Sprechen Sie uns an!
Tel. 0231/61 804-0, info@W3L.de

W3L-Akademie



Flexibel online lernen und studieren!

In Zusammenarbeit mit der Fachhochschule Dortmund bieten wir

zwei Online-Studiengänge

- B.Sc. Web- und Medieninformatik
- B.Sc. Wirtschaftsinformatik

und 7 Weiterbildungen im IT-Bereich an.



Besuchen Sie unsere Akademie!
<http://Akademie.W3L.de>