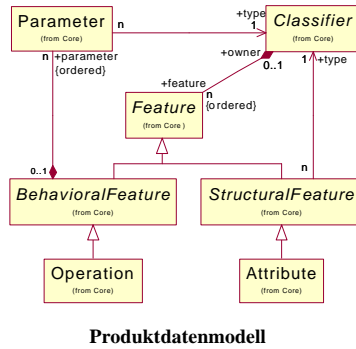
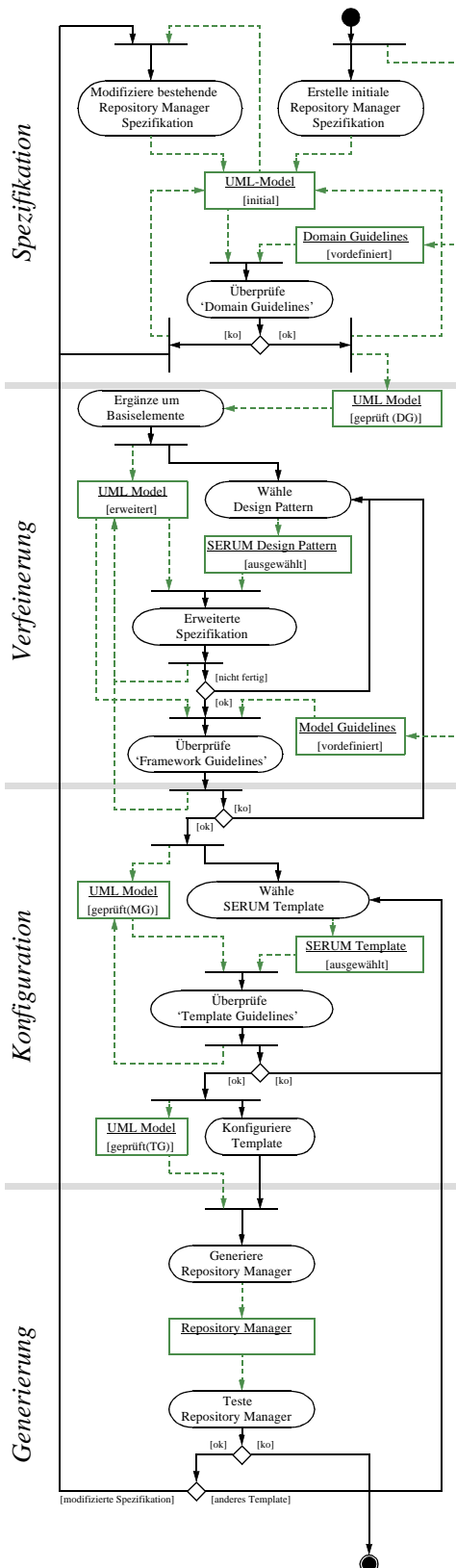


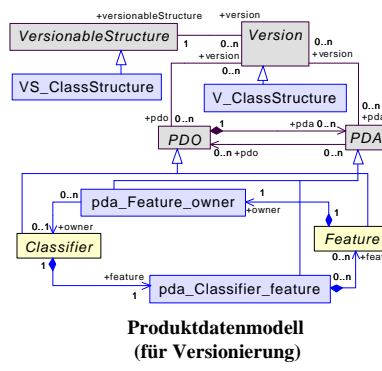
# SERUM-Prozeß

Generating Software Engineering Repositories using UML



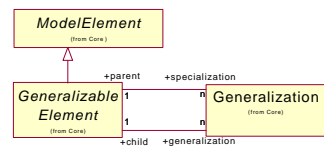
## Spezifikation basierend auf dem UML-Metamodell

- q standardisiert (OMG), weit verbreitet, objektorientiert
- q graphische Modellierung und Spezifikation
- q umfangreiche Modellierungselemente verfügbar
  - m Struktur (Class, Interface, DataType, Association, ...)
  - m Verhalten (StateMachine, ActivityGraph, ...)
- q deskriptive Sprache zur Spezifikation weiterer Integritätsbedingungen
  - m Object Constraint Language (OCL)



## SERUM Framework: technologieunabhängiger Teil

- q vordefinierte Basiselemente
- q vordefinierte Verfeinerungsoperationen (SERUM DesignPattern)
  - m Parameter: Welche Teile des Modells sollen verfeinert werden?
  - m Vorbedingungen: Darf das DesignPattern angewendet werden?
  - m Zusicherungen: Was darf zukünftig nicht mehr geändert werden?
  - m Skript: Wie wird das Modell verändert?
- q Modellierungsrichtlinien
  - m Domain Guidelines  
Entspricht die Spezifikation den Vorgaben aus der Domäne?
  - m Framework Guidelines  
Entspricht die Spezifikation den Richtlinien des Frameworks?
  - m Template Guidelines  
Entspricht die Spezifikation den Anforderungen des Generators?



## SERUM Framework: technologieabhängiger Teil

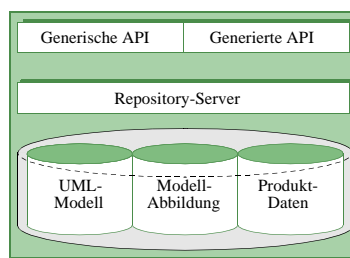
- q unvollständiger Repository Manager
- q Konfiguration kompatibler Komponenten (SERUM Template)
  - m Rahmenarchitektur
  - m Regeln zur Code-Generierung
  - m vorgefertigte Code-Templates zur Vervollständigung
  - m fertige Komponenten zur Komposition
  - m Parameter für Generierungsregeln, Code-Templates und fertige Komponenten

```

Informix SQL

create row type data_generalizable_element_ty
(...)
under data_model_element_ty;
create table generalizable_element_ta
of type generalizable_element_ty
(...)
under model_element_ta;
    
```

## Generierungsregeln



## Repository Manager

## Repository Manager generieren

- q Erzeugung von Code mit den Generierungsregeln
  - m Typen, Tabellen, Views, Trigger, UDRs, ...
  - m API-Klassen, ...
- q Vervollständigung der Templates
- q Parametrisierung der Komponenten
- q Kombination der entstandenen Artefakte