



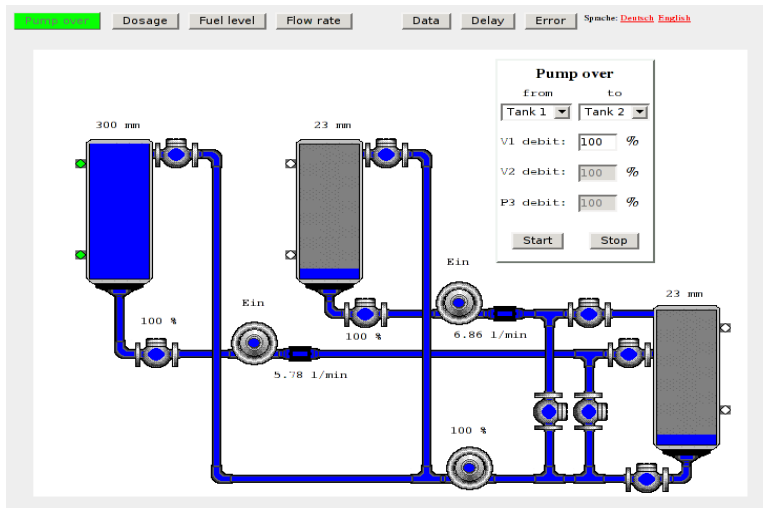
**TECHNISCHE
UNIVERSITÄT
DRESDEN**

**DRESDEN
concept**

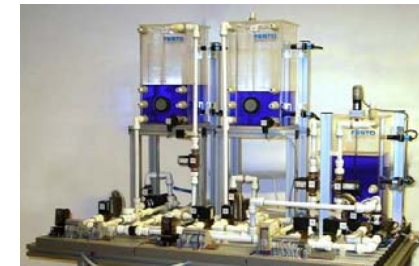


Fakultät Elektrotechnik und Informationstechnik Institut für Automatisierungstechnik

Modellbasierte Entwicklung von Visualisierungslösungen für die industrielle Automation



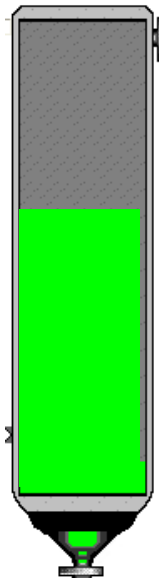
Bedienen und
Beobachten



- Hersteller- und Plattformabhängigkeit
- händische interaktive Entwicklung von Bedienoberflächen

Aktuelle Bibliotheken

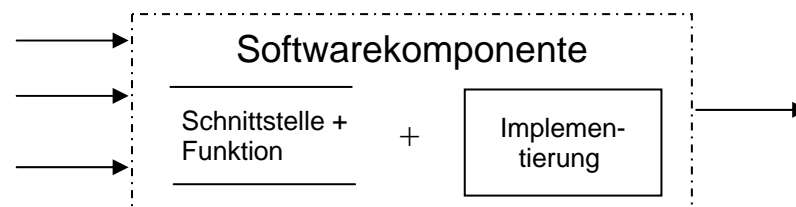
Funktion + Schnittstelle + Implementierung



Höhe =
(Füllstand * 2) + b

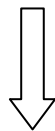
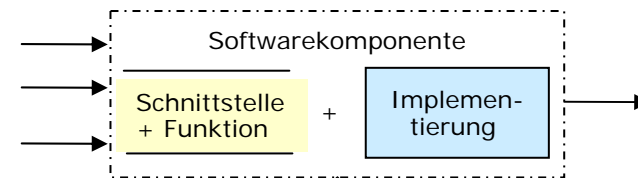
- X,Y-Pos
- Farbe
- Rahmen
- Füllstand

- C++
- Java
- WinCC
- .net
-



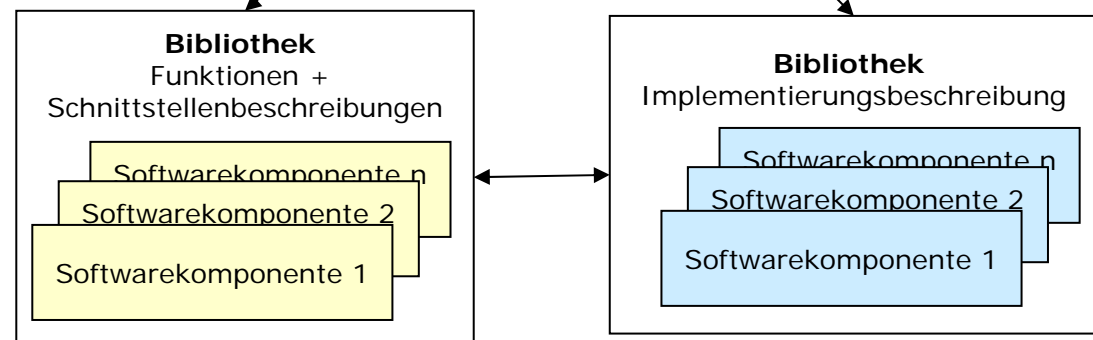
Technologieunabhängige Softwarekomponenten

- Trennung von Funktion und Implementierung
- Technologieunabhängige Beschreibung der Schnittstelle mittels XML
- Implementierungsvorschriften durch Transformation

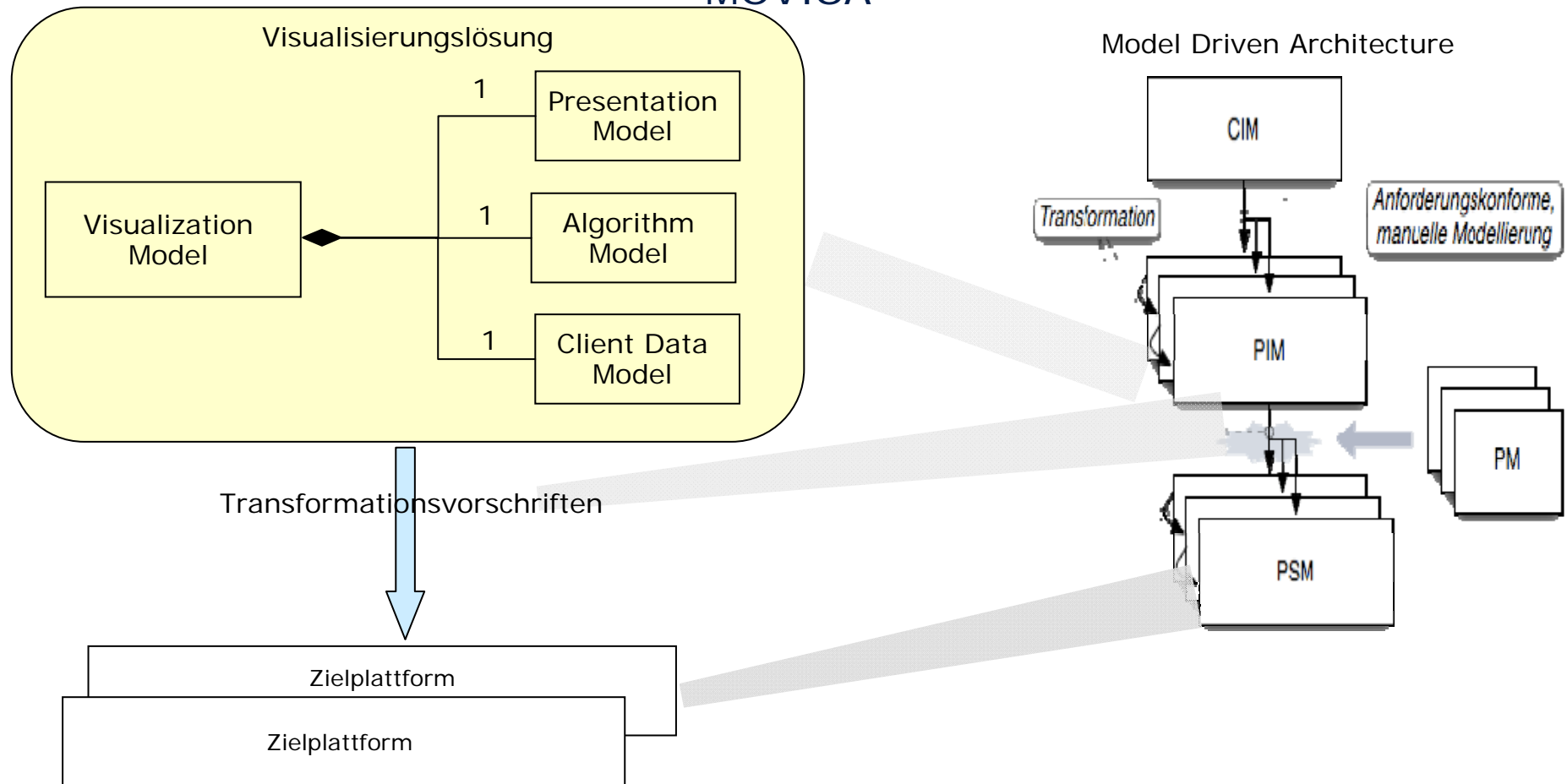


Modellgetriebene Technologien

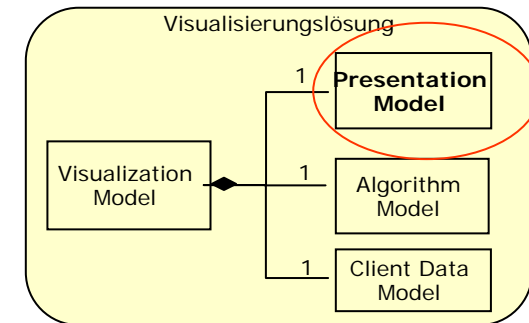
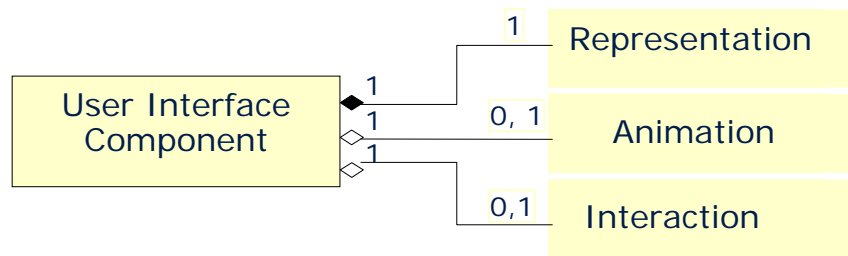
XML Extensible Markup Language




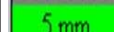



Model-Driven Development of Visualization Solutions in Industrial Automation - MOVISA



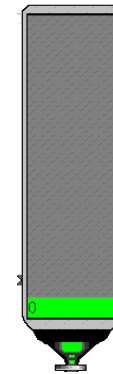
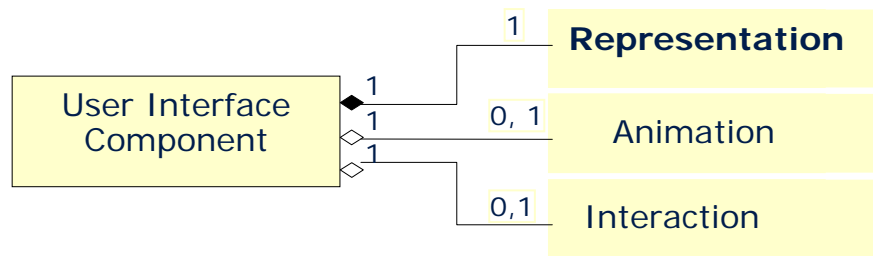
Presentation Model



- User Interface Komponenten (elementar, komplex)
- User Interface Eigenschaften
- Navigation Modell

	Kreuzen Sie die <input checked="" type="checkbox"/> Ventil 1 <input type="checkbox"/> Ventil 2 <input checked="" type="checkbox"/> Ventil 3				Umpumpen von: <input type="radio"/> Behälter 1 <input checked="" type="radio"/> Behälter 2 <input type="radio"/> Behälter 3		Füllhöhe Behälter 1: <input type="text" value="250"/> in mm
Schalter	Schalt- leiste	Anzeige	Menue	Bild	Checkbox	Trend	Eingabefeld

Presentation Model - allgemeine Bildungsvorschrift (Metamodell)

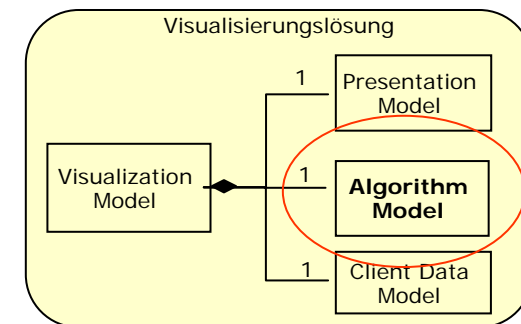
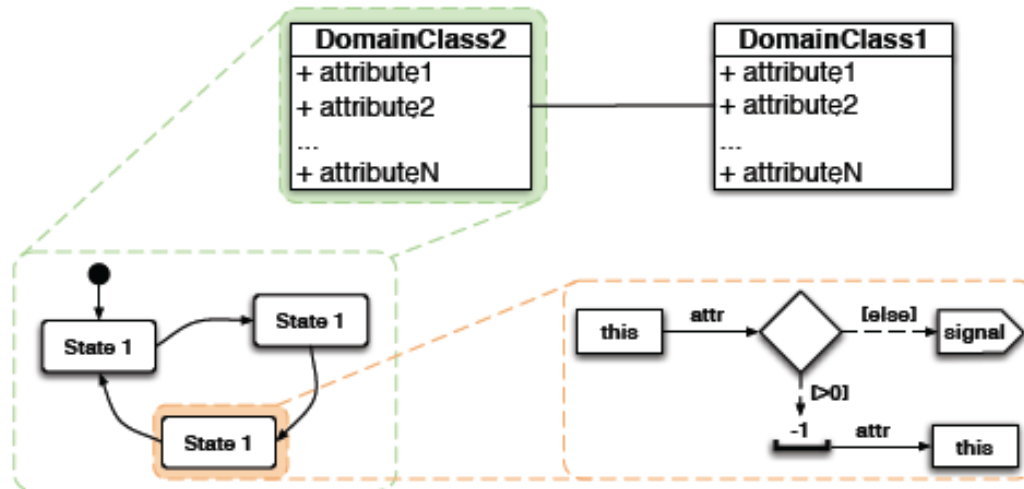


```

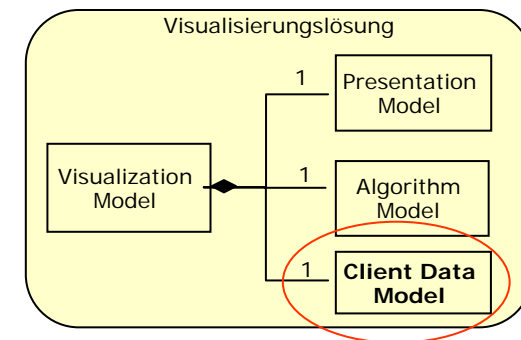
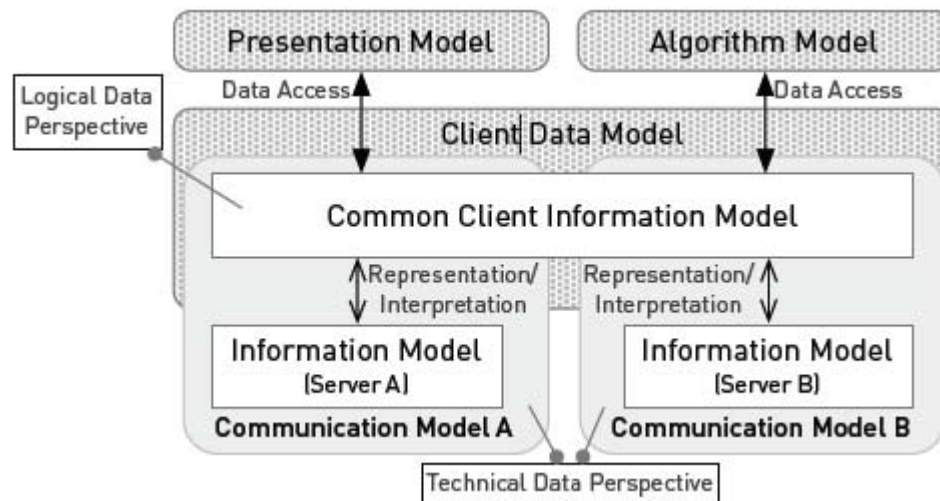
<xs:complexType name="positionType">
  <xs:annotation>
    <xs:documentation>position .....

```

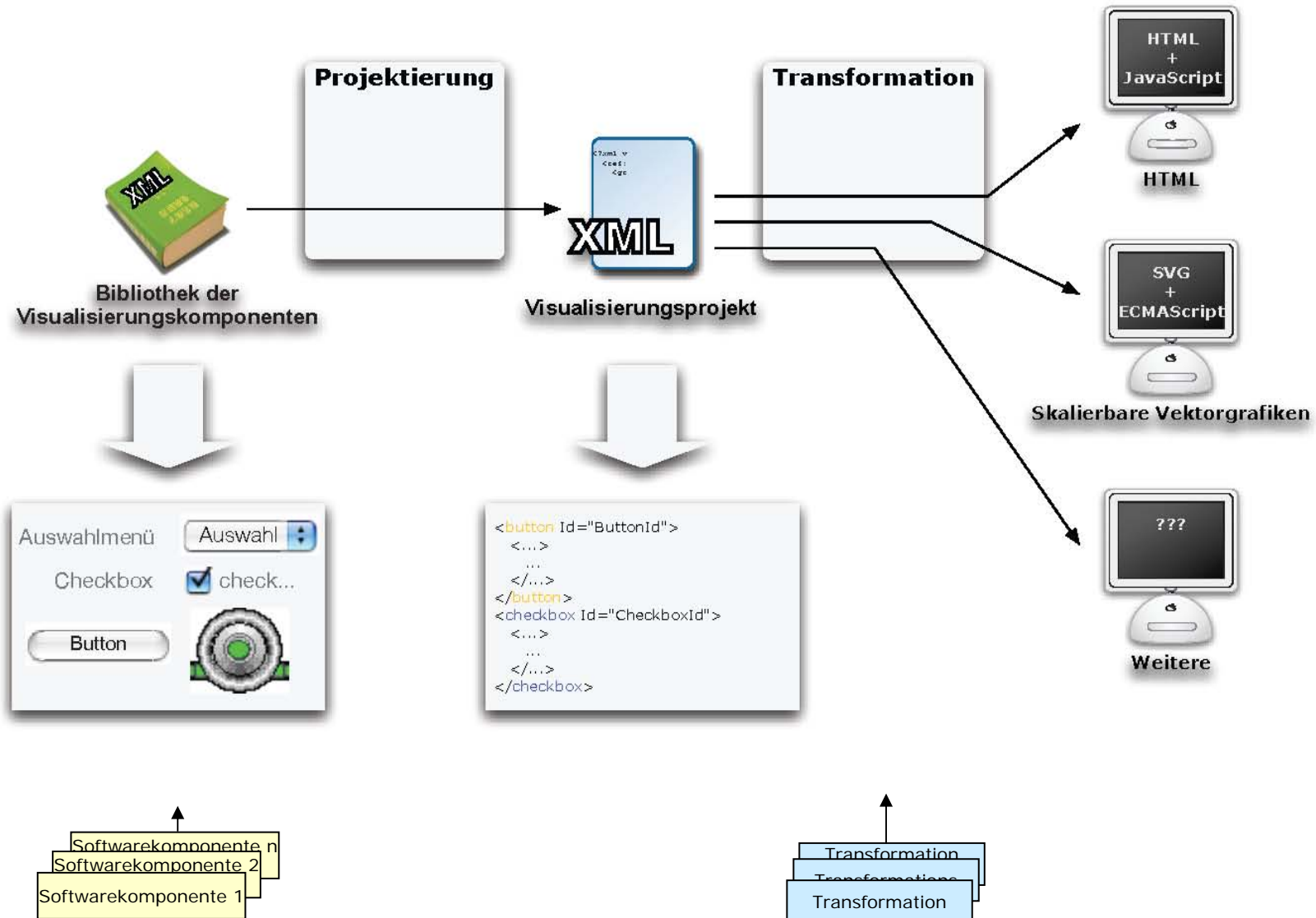
Algorithm Model



- Plattformunabhängige Programmiersprache
- Executable UML
- Class, State, Action

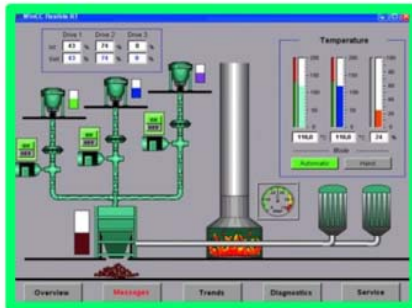


- Kommunikation mit Datenservern, Bereitstellung von Prozessdaten
- Herausforderung: Vielzahl von Kommunikationstechnologien
- Technical Data Perspective: technisches Interface
- Logical Data Perspective: Logisches Interface

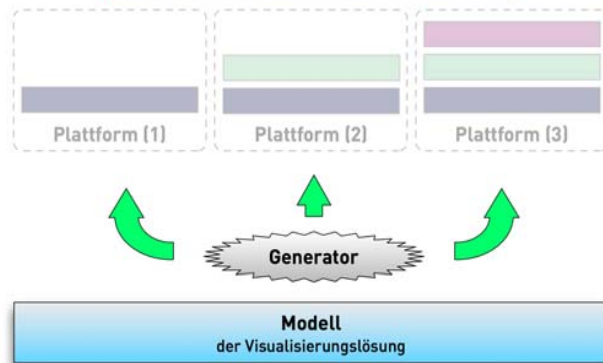


MOVISA im Überblick

Ziel: Visualisierungslösungen für verschiedene Zielplattformen (plattformunabhängig)

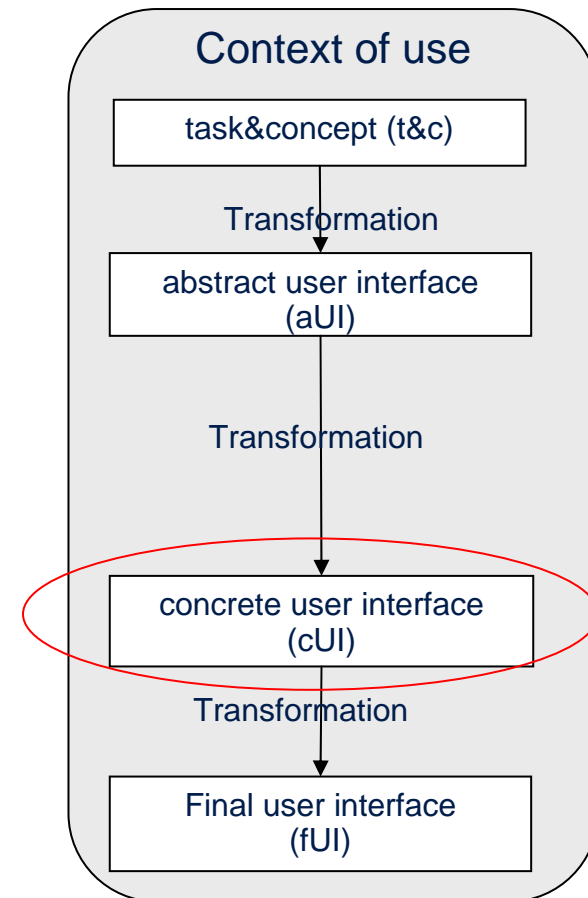
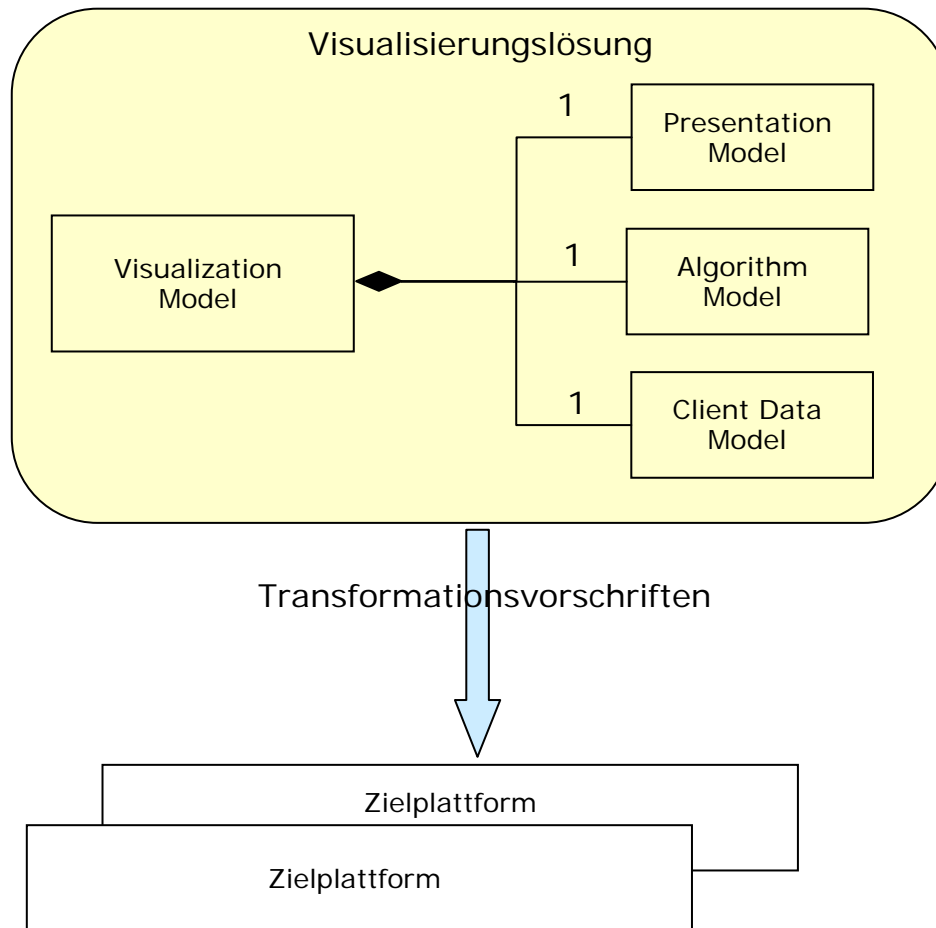


Ansatz: Technologieneutrale Beschreibung von Visualisierungslösungen, automatische Generierung des Quellcodes für verschiedene Zielsysteme



Weg Entwurf einer domänenspezifischen Sprache zur Modellierung der (1) *Benutzungsoberfläche*, der (2) benötigten *Prozessdaten* verschiedener Servertechnologien und (3) nutzerspezifische *Algorithmen*

Mittel MOF-konforme Modellierungskonzepte, Realisierung mit Eclipse (Ecore, EMF, GMF, EMFText, Xpand, Xtend, MWE)

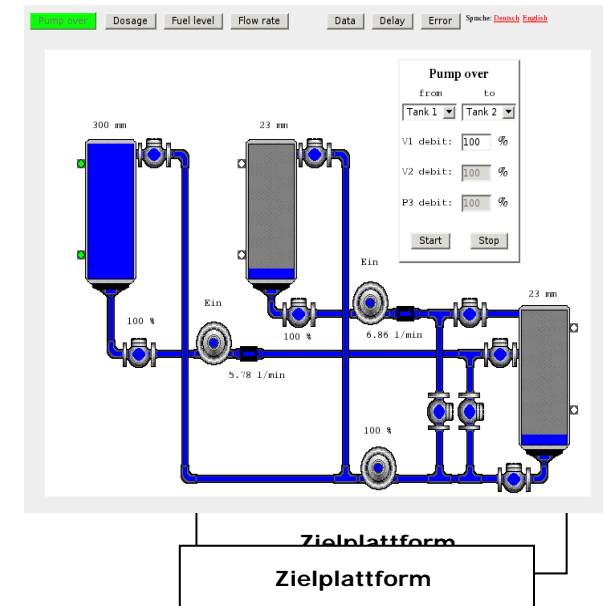


Stand

- getestete Metamodelle für Presentation, Algorithm, Client Data Model, (historical)
- Transformationsvorschriften für HTML, JavaScript, SVG, Java, (c#)
- Test von Prototypen (ca. 60 analoge Prozessvariablen)
<https://141.30.119.195:9443/learnet.app/URLoginView.jsp>

Ausblick

- Sicherheitsmechanismen (security Modell)
- Entwicklungswerkzeuge (Editor)
- Leistungstests
- Plattformmodelle



- Trennung funktionaler Aspekte von Implementierungsdetails
- Methodik zur Entwicklung von Bedienoberflächen unabhängig von Zieltechnologien und Zielplattform entwerfen
- keine zwingend händische interaktive Entwicklung von Bedienoberflächen
- einheitliche Entwicklungsumgebung für beliebige Zieltechnologien
- Nutzung MOF kompatibler Modellierungskonzepte
- Implementierung mit Eclipse ecosystem (EMF, GMF, EMFText, oAW, Epsilon)