



# **Modellbasierte Werkzeugintegration**

Seminar Modellgetriebene Softwareentwicklung  
Zwischenpräsentation

Name: Ngoc Diep Phan

## Warum Werkzeugintegration?

- Bestimmte Werkzeuge unterstützen nur ausgewählte Phasen der SW-Entwicklung.
- Die Modelle sind konzeptionell stark mit einander verzahnt, können aber nicht gemeinsam benutzt werden.
- Gleiche Spezifikationen müssen oft mehrmals erfasst werden  
=> ineffizient und fehleranfällig

## Lösung:

- Kopplung der Entwicklungswerkzeuge, damit die Modelle reibungslos ausgetauscht werden können.
- Werkzeugintegration: Die gemeinsame Nutzung unterschiedlicher Werkzeuge
- Interoperabilität: Mehrere Werkzeuge können dasselbe Modell verwenden, um den reibungslosen Austausch von Daten und Informationen zu erreichen.

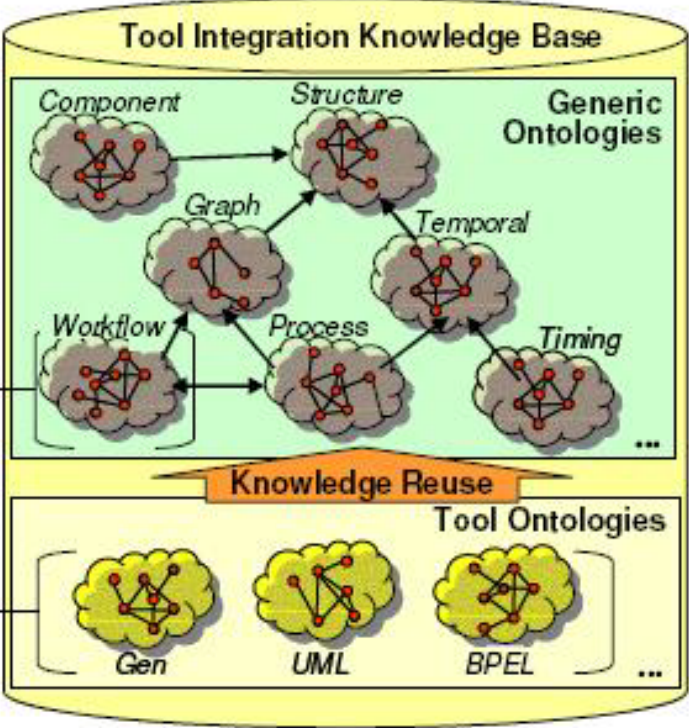
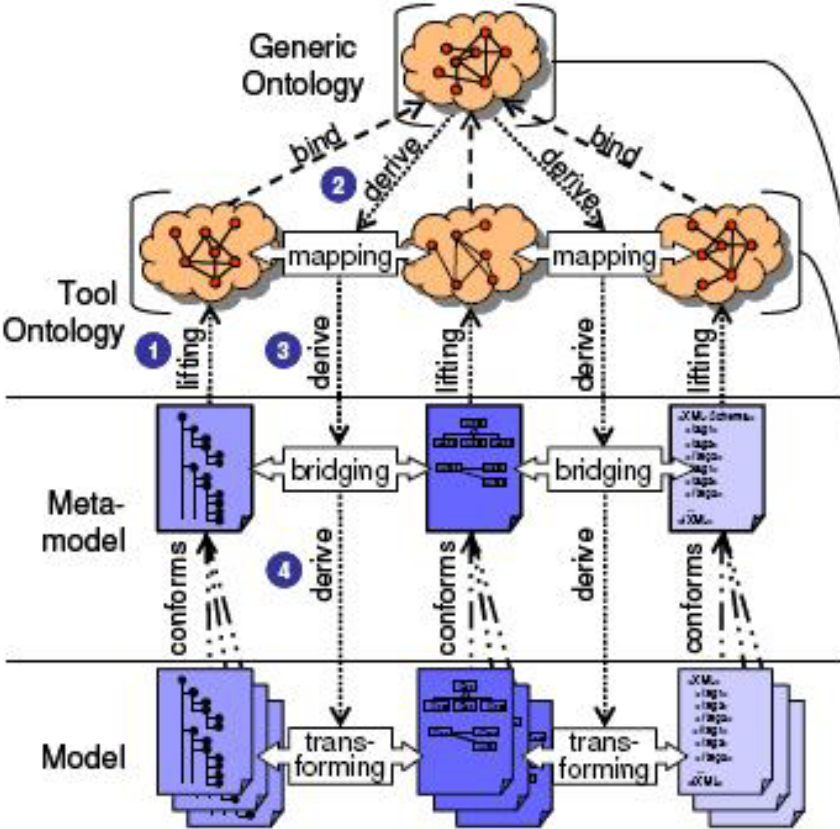
## Ansätze

- **Die Daten werden als Modelle betrachtet. Die Transformation von Modellen erfolgt auf ausgewählten Ebenen der M3M.**
  - **Beispiel: ModelCVS, objectiF**
- **Andere versuchen, einen gemeinsamen Standardformat zu benutzen, um den Datenaustausch zu ermöglichen.**
- **Standardformate:**
  - **XMI (XML Metadata Interchange)**
  - **CDIF (CASE Data Interchange Format )**
  - **GXL (Graph eXchange Language)**

## **ModelCVS (A Semantic Infrastructure for Model-based Tool Integration )**

- **Ermöglicht eine transparente Transformation von Modellen zwischen verschiedenen Werkzeugen durch:**
  - **Gemeinsame Benutzung der Modelle**
  - **Modellsynchronisation**
  - **Modellversionierung**
- **Die Integration erfolgt in der Metamodell-Ebene.**

# ModelCVS



# objectiF

- **die vorhandenen Modelltransformationen in objectiF anpassen**
- **mit objectiF für objectiF neue Modelltransformationen entwickeln**
- **Vorteil:**
  - **Erweiterbarkeit**
  - **Anpassbarkeit**
  - **kostengünstige Weiterentwicklung und Wartung**

# GXL (Graph eXchange Language)

- eine *XML-Subsprache*
- bietet eine standardisierte Notation für den Austausch von Graphinstanzen und Graphschemata, die beide mit derselben *XML Document Type Definition (DTD)* kodiert werden.
- kann als Generalisierung dieser Formate angesehen werden:
  - ATerms
  - Relation Partition Algebra
  - Rigi Standard Format
  - daVinci
  - GML/Graphlet
  - XGMML
  - Graph(X)ML
  - GTXL