**Motivation**

- The shift from code-centric to model-centric software development places models as first class entities in model-driven development processes.
- A wide variety of modeling tools is available, supporting different development tasks (e.g., model checking, simulation, code generation) and often advocating different (possibly domain-specific) modeling languages.
- To use tools in combination and thus to fully exploit the potential of model-driven software development, seamless exchange of models is vital.

**Problem**

- Heterogeneity exists in textual representation, syntax, semantics, scope of modeling languages and exchange formats used by different tools.
- Integration is a cumbersome, error-prone and highly repetitive task not least when, e.g., new versions of modeling languages emerge.
- Inconsistencies are likely when concurrent development of models on basis of different tools takes place.

**Hypothesis**

- Ontologies can be used to resolve heterogeneity at a higher level of abstraction, forming the basis for semi-automatically deriving model transformations.
- A knowledge base can capture tool integration experience leading to reusable and qualitative solutions thus turning integration into an engineering discipline.
- Model versioning techniques can be developed based on semantically enriched descriptions of modeling languages.

**Goal**

- Development of concepts and techniques for establishing a semantic infrastructure supporting model-based tool integration.
- Prototypical implementation using state of the art technologies and standards, e.g., Eclipse's Modeling Framework (EMF) and Ecore, Inria's AAWA platform, and OMG's Ontology Definition Metamodel (ODM).
- Evaluation on basis of an industrial case study focusing on integrating Computer Associate's AllFusionGen modeling language and OMG's Unified Modeling Language.

---

**Project Partners**

- Vienna University of Technology (TU Wien)
- Johannes Kepler University Linz (JKU)

**Contact**

- UniAss. Dr. W. Schwinger
- Department of Telecooperation
- Johannes Kepler University Linz
- wieland.schwinger@jku.ac.at
- www.modelcvs.org

---

**Technical Architecture of ModelCVS**

**Conceptual Architecture of ModelCVS**

**Matching between Pseudo and Refactored Ontologies**

**Bridging Between Metamodels**

**Transforming Models Through ATL**

**Matching Fully Refactored Ontologies**

---

**Motivation**

- The shift from code-centric to model-centric software development places models as first class entities in model-driven development processes.
- A wide variety of modeling tools is available, supporting different development tasks (e.g., model checking, simulation, code generation) and often advocating different (possibly domain-specific) modeling languages.
- To use tools in combination and thus to fully exploit the potential of model-driven software development, seamless exchange of models is vital.

**Problem**

- Heterogeneity exists in textual representation, syntax, semantics, scope of modeling languages and exchange formats used by different tools.
- Integration is a cumbersome, error-prone and highly repetitive task not least when, e.g., new versions of modeling languages emerge.
- Inconsistencies are likely when concurrent development of models on basis of different tools takes place.

**Hypothesis**

- Ontologies can be used to resolve heterogeneity at a higher level of abstraction, forming the basis for semi-automatically deriving model transformations.
- A knowledge base can capture tool integration experience leading to reusable and qualitative solutions thus turning integration into an engineering discipline.
- Model versioning techniques can be developed based on semantically enriched descriptions of modeling languages.

**Goal**

- Development of concepts and techniques for establishing a semantic infrastructure supporting model-based tool integration.
- Prototypical implementation using state of the art technologies and standards, e.g., Eclipse’s Modeling Framework (EMF) and Ecore, Inria’s AAWA platform, and OMG’s Ontology Definition Metamodel (ODM).
- Evaluation on basis of an industrial case study focusing on integrating Computer Associate’s AllFusionGen modeling language and OMG’s Unified Modeling Language.

---

**Project Partners**

- Vienna University of Technology (TU Wien)
- Johannes Kepler University Linz (JKU)

**Contact**

- UniAss. Dr. W. Schwinger
- Department of Telecooperation
- Johannes Kepler University Linz
- wieland.schwinger@jku.ac.at
- www.modelcvs.org

---

**Technical Architecture of ModelCVS**

**Conceptual Architecture of ModelCVS**

**Matching between Pseudo and Refactored Ontologies**

**Bridging Between Metamodels**

**Transforming Models Through ATL**

**Matching Fully Refactored Ontologies**

---