Model-Integrated Computing (MIC)
(or MDA, MDE, MDD, MBD,…)

Janos Sztipanovits & Gabor Karsai
ISIS, Vanderbilt University

US-EU Review
July 8, 2005
Paris, France

VANDERBILT UNIVERSITY
INSTITUTE FOR SOFTWARE INTEGRATED SYSTEMS

Chess

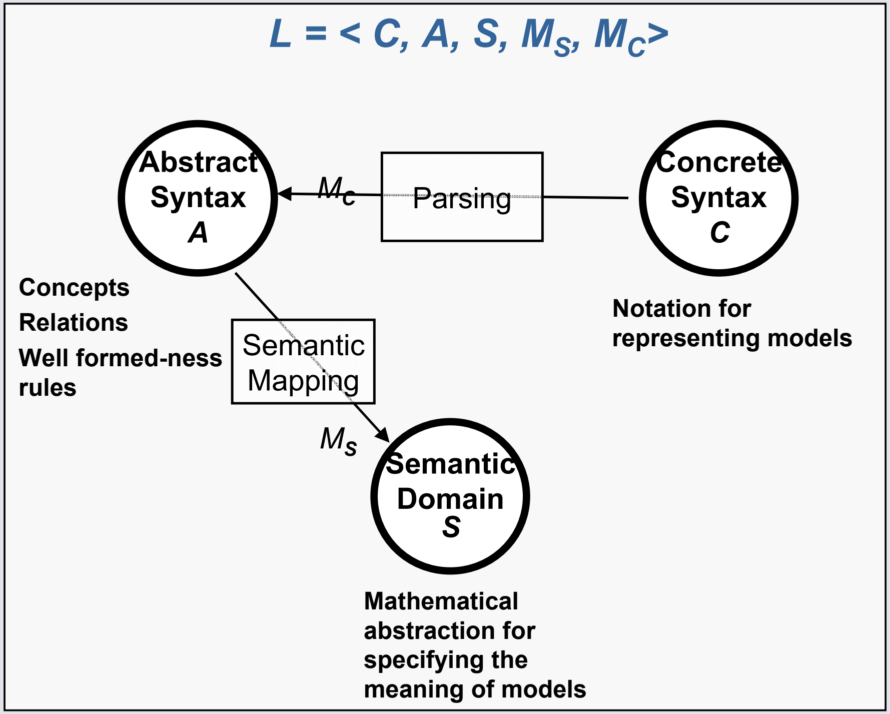
Model-Based Design

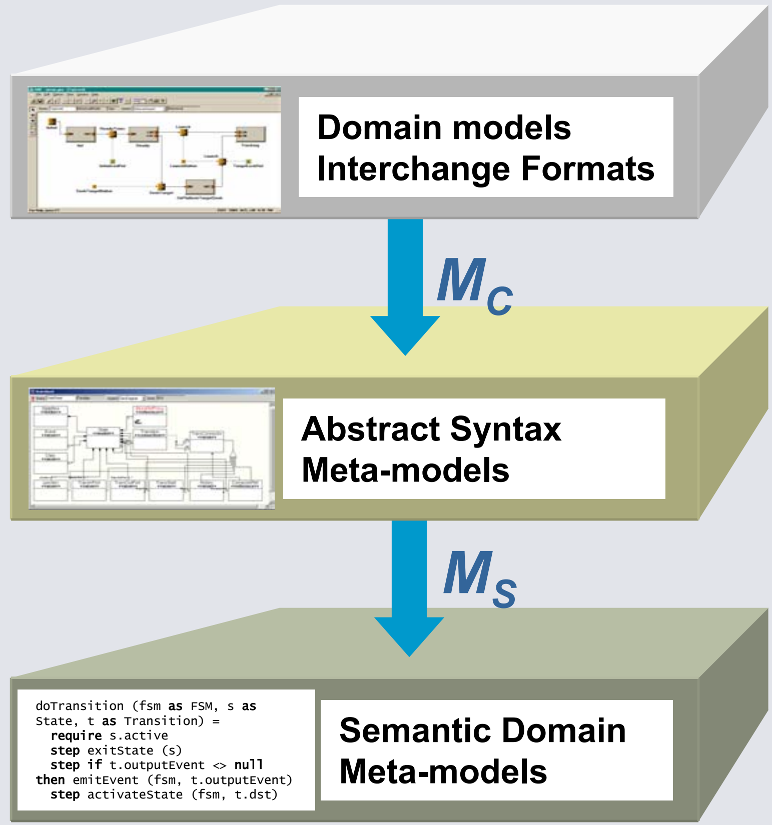
Model-based design focuses on the *formal representation, composition, and manipulation of models* during the design process.

Research Agenda in Model-Based Design

1. Composition of Domain Specific Modeling Languages
2. Model Transformation
3. Model Synthesis

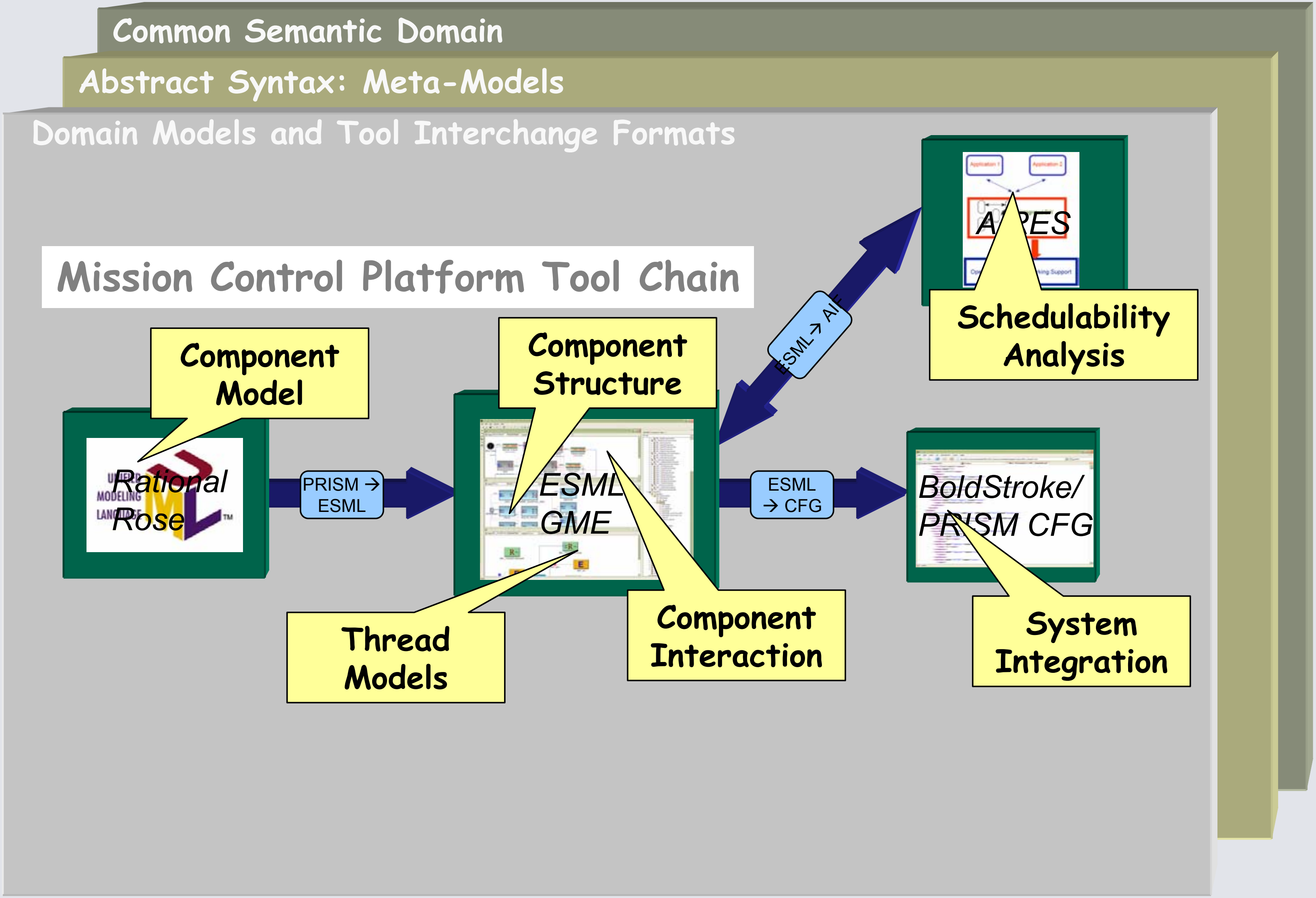
Domain Specific Modeling Languages (DSML)



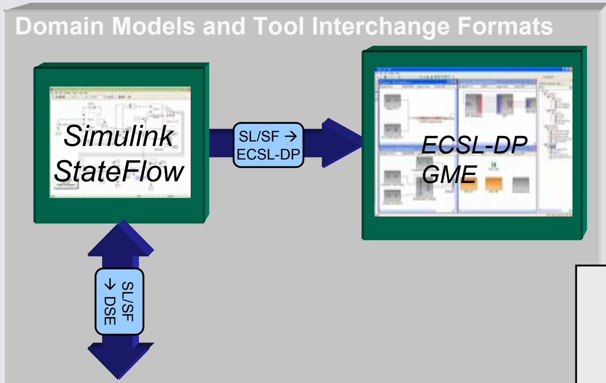


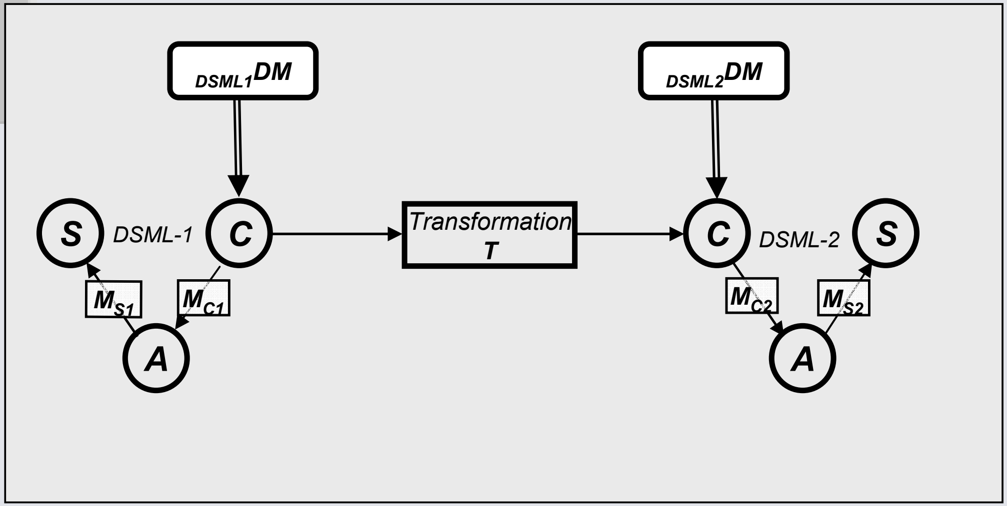
DSML-s are the foundations for model-based design

Mission Control Platform (MCP) Tool Chain from "Above"



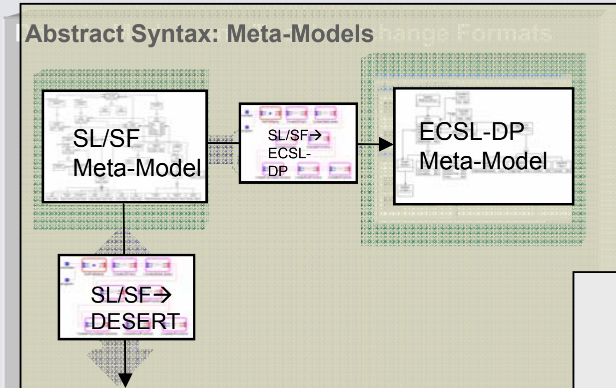
Constructing Design Flow: Modeling and Transformations



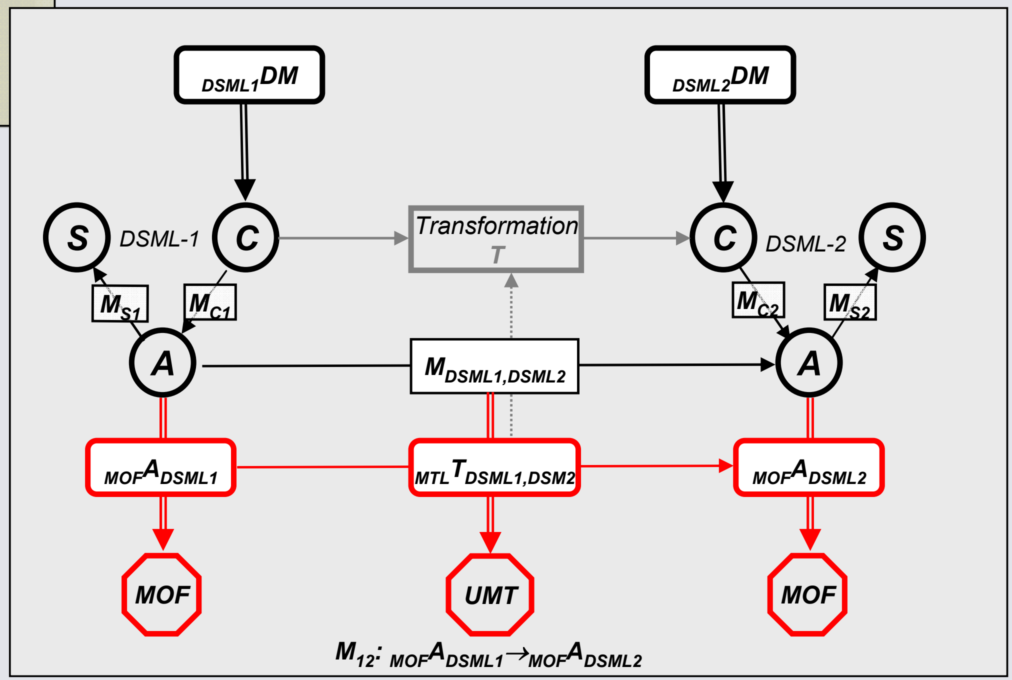


- Large influence of concrete syntax
- No clear role of semantics
- It is not clear what are we doing?

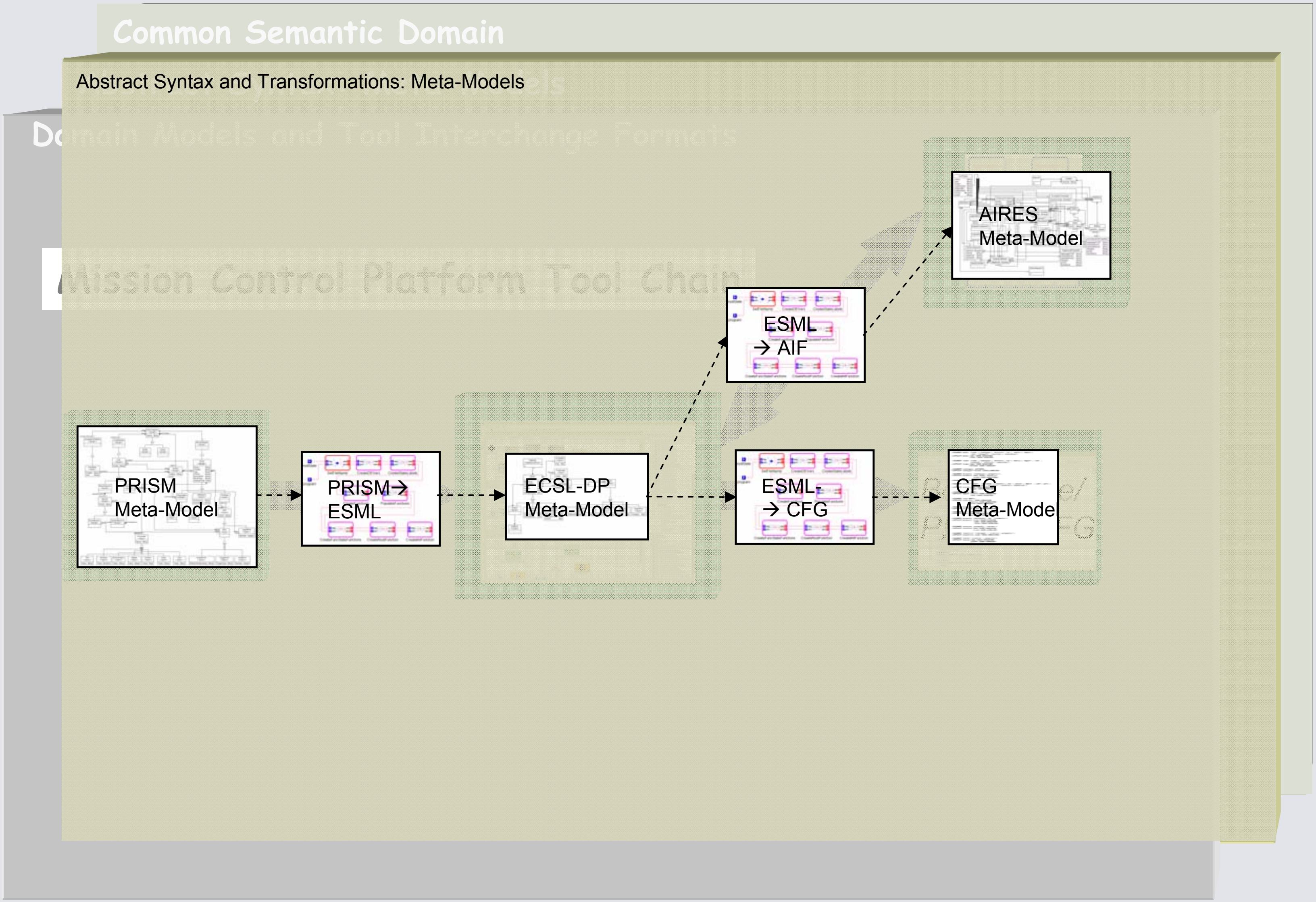
Abstract Syntax Metamodeling



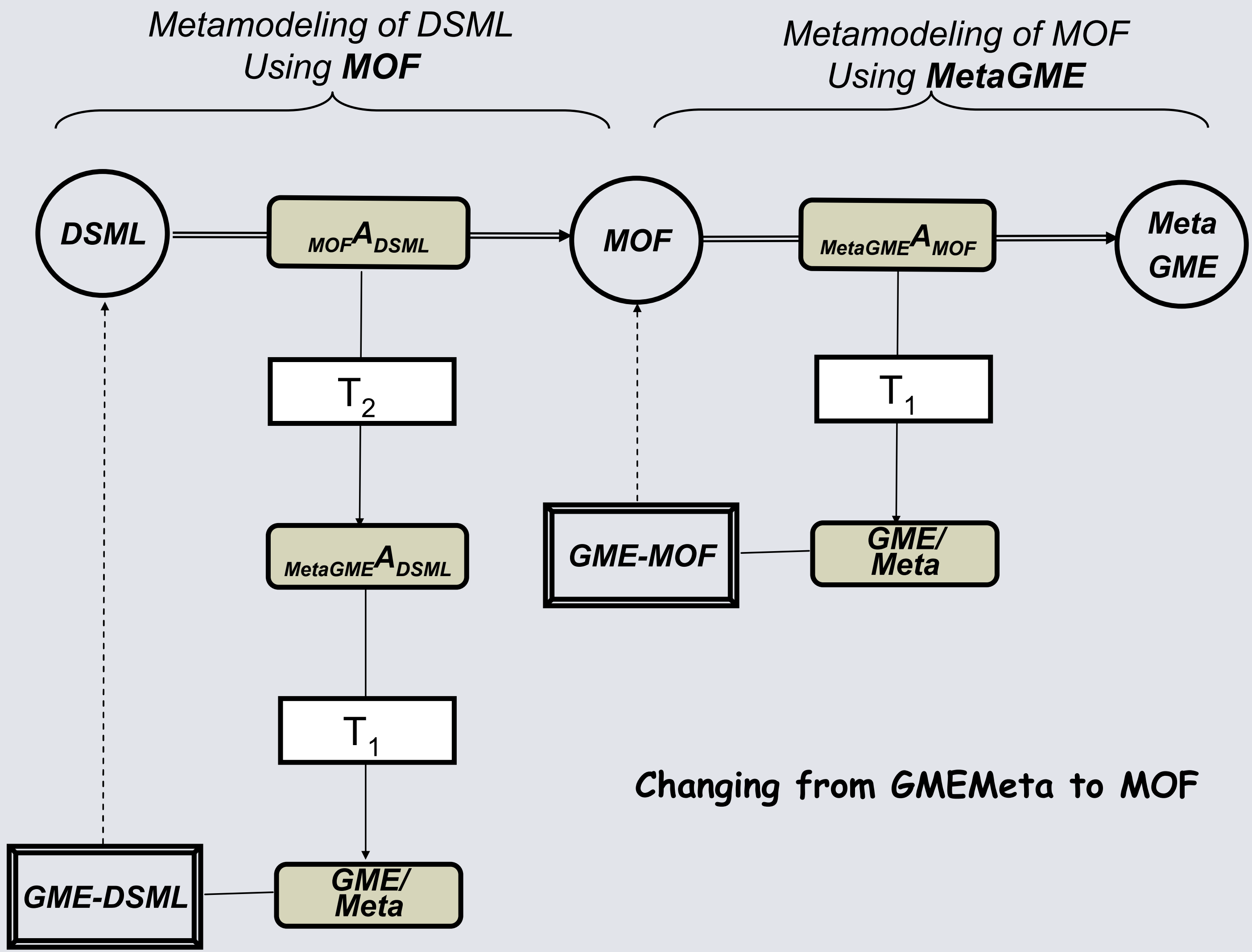
- Gives structural semantics for the models
- Set-valued Semantic Domain for the metamodels and transformations



Mission Control Platform at the "Middle" Layer

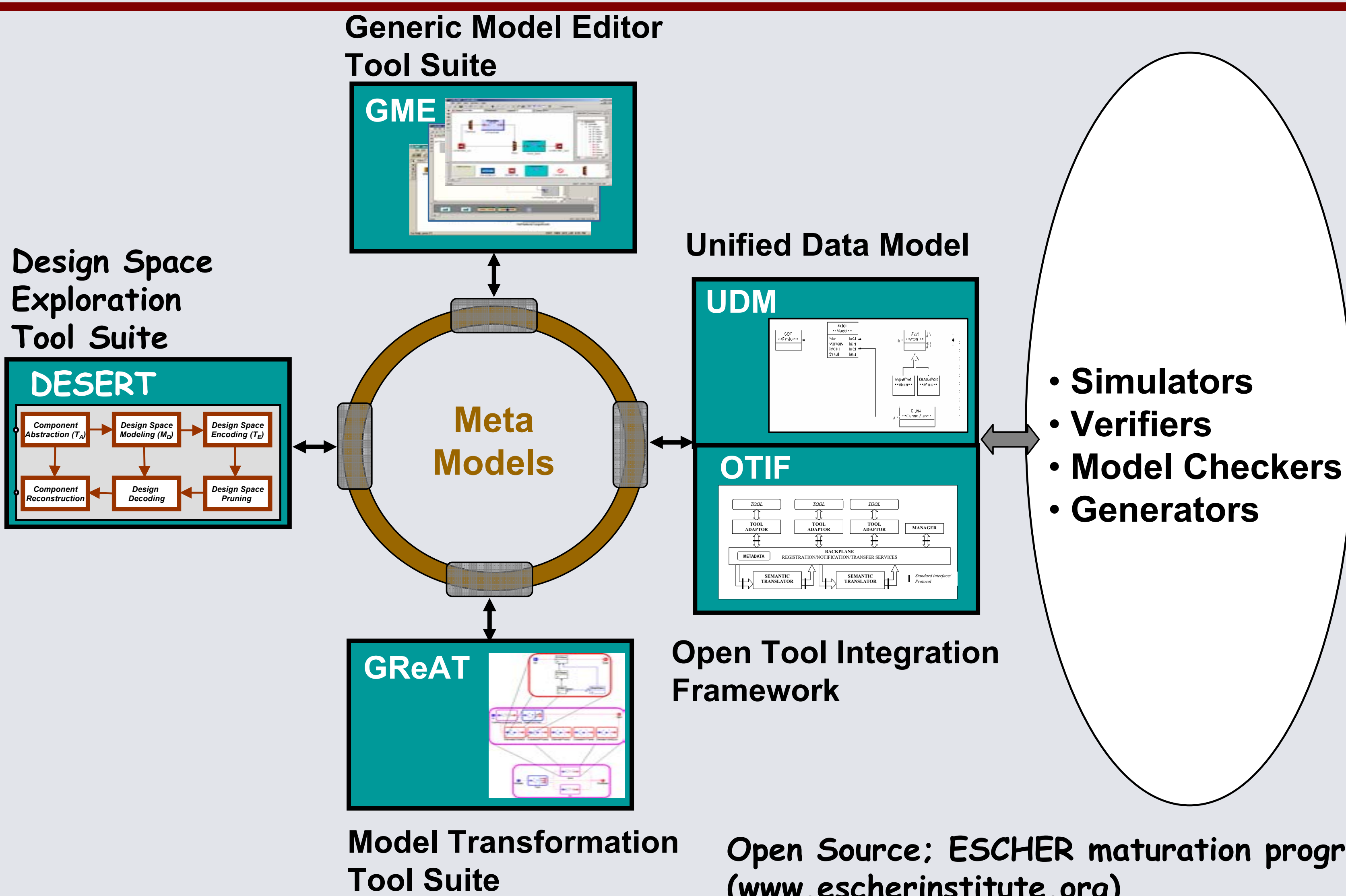


Metamodeling Languages



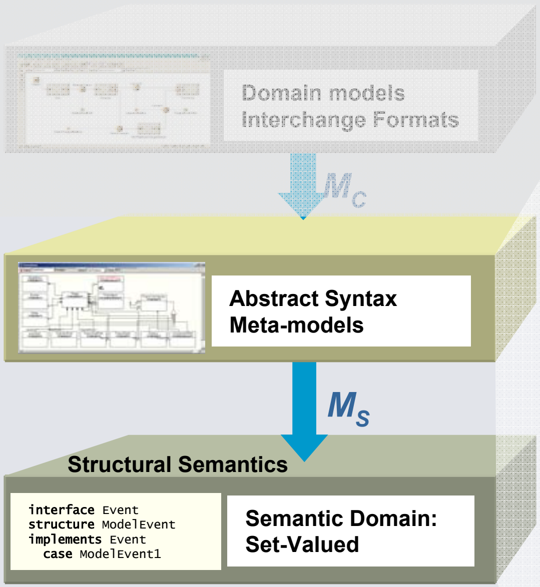
Changing from GMEMeta to MOF

Metaprogrammable Tool Suite



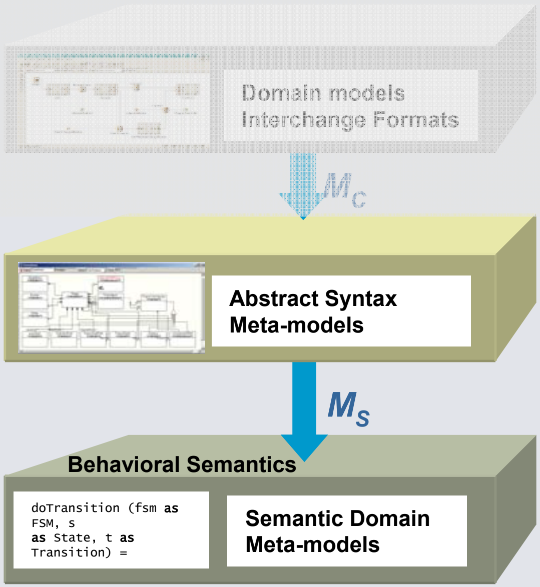
Open Source: ESCHER maturation program
(www.escherinstitute.org)

Ongoing Research on DSML-s and Model Transformations

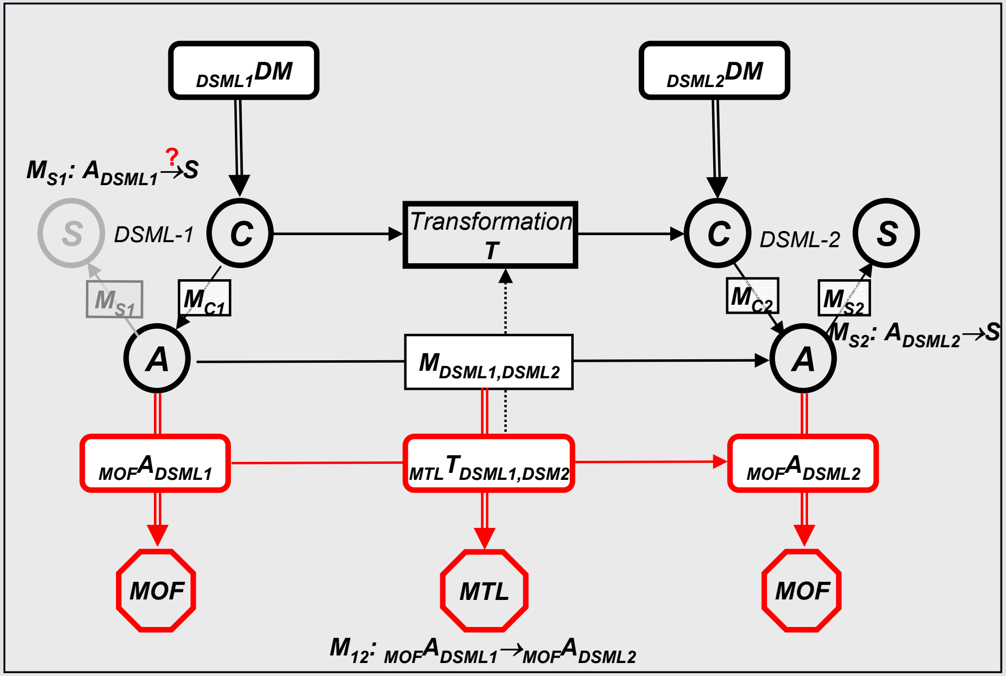


- Compositional construction of Metamodels (inheritance, packages, libraries, operators)
- Compositional construction of Model Transformations
- Multiple Aspect Modeling and modeling of aspect inter-dependences:
  - constraint-based,
  - transformation-based
- Formal semantics of transformations
- Reasoning about properties of transformations
- Platform modeling and use of embedded platform models in transformations
- Pushback reasoning in transformations
- Generation of efficient code from graph transformations
- Transformations for embedded system platforms
- Using graph transformations for embedded component adaptation
- Embedding graph transformations in the run-time platform

Metamodeling and Model Transformation Use Cases

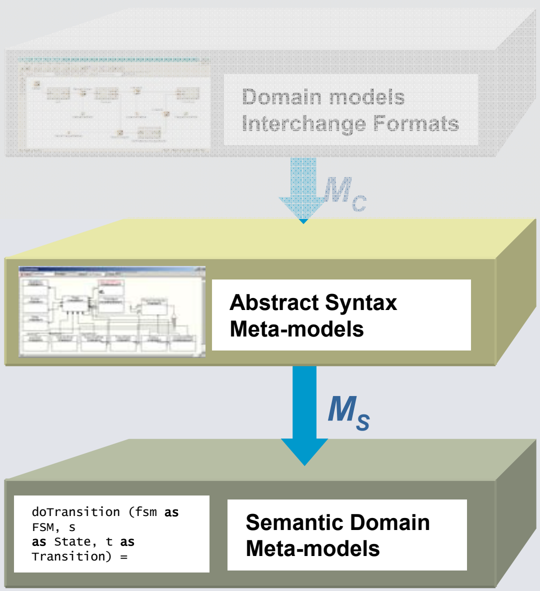


1. Transformational Specification of Behavioral Semantics

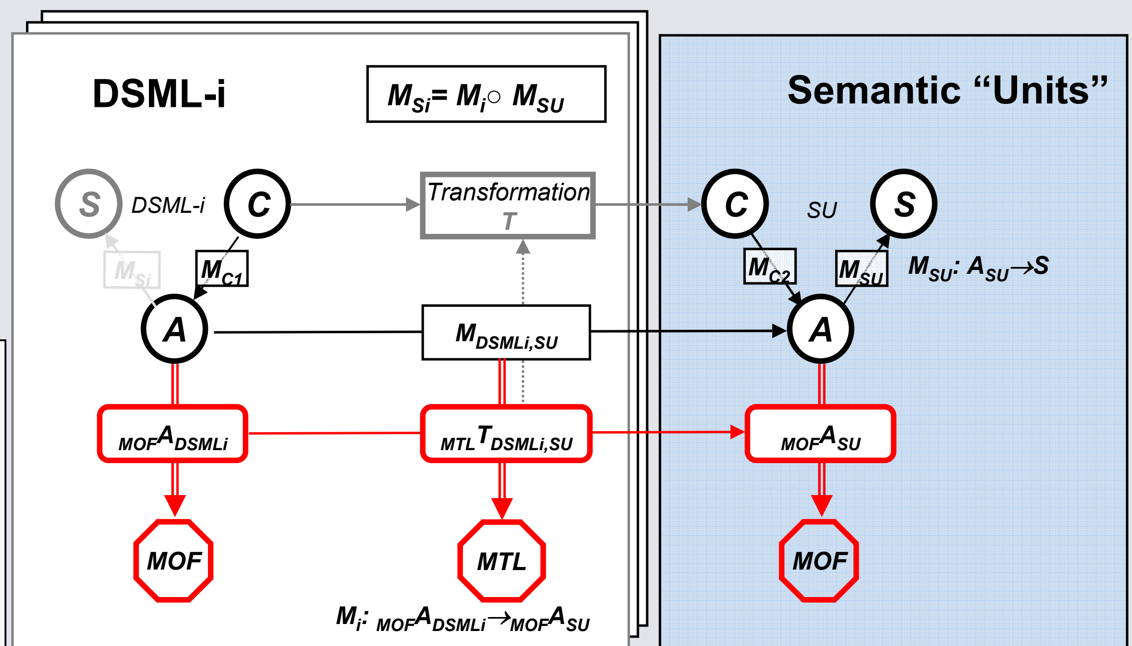


$$M_{S1} = M_{S2} \circ M_{12}$$

Metamodeling and Model Transformations Use Cases



2. Semantic Anchoring of DSML-s

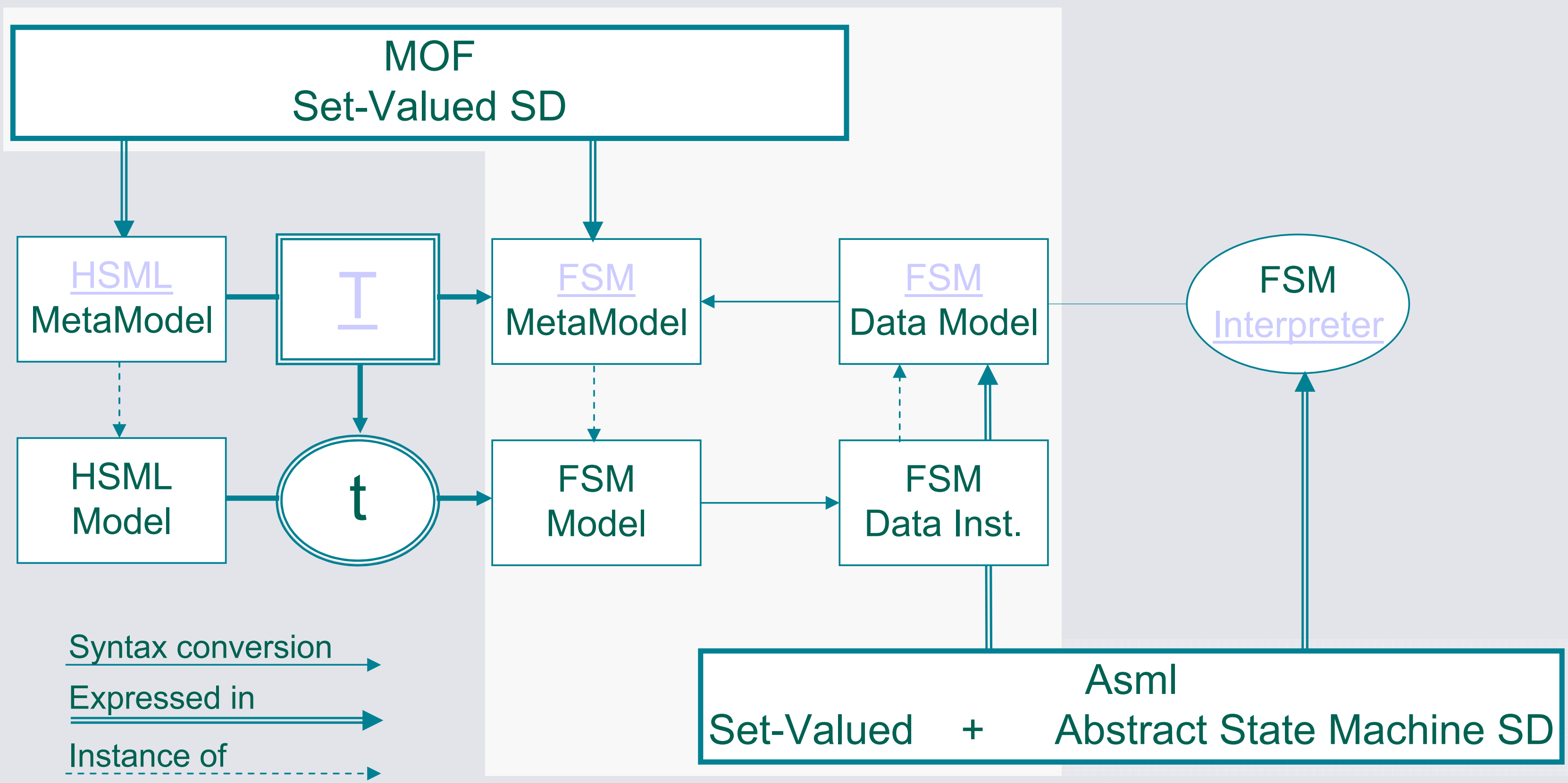


- The "Semantic Units" are selected abstract semantics such as MoC-s
- DSML-s or their aspects are anchored to the MoC-s using transforma-tions
- The "Semantic Units" are specified in a formal framework

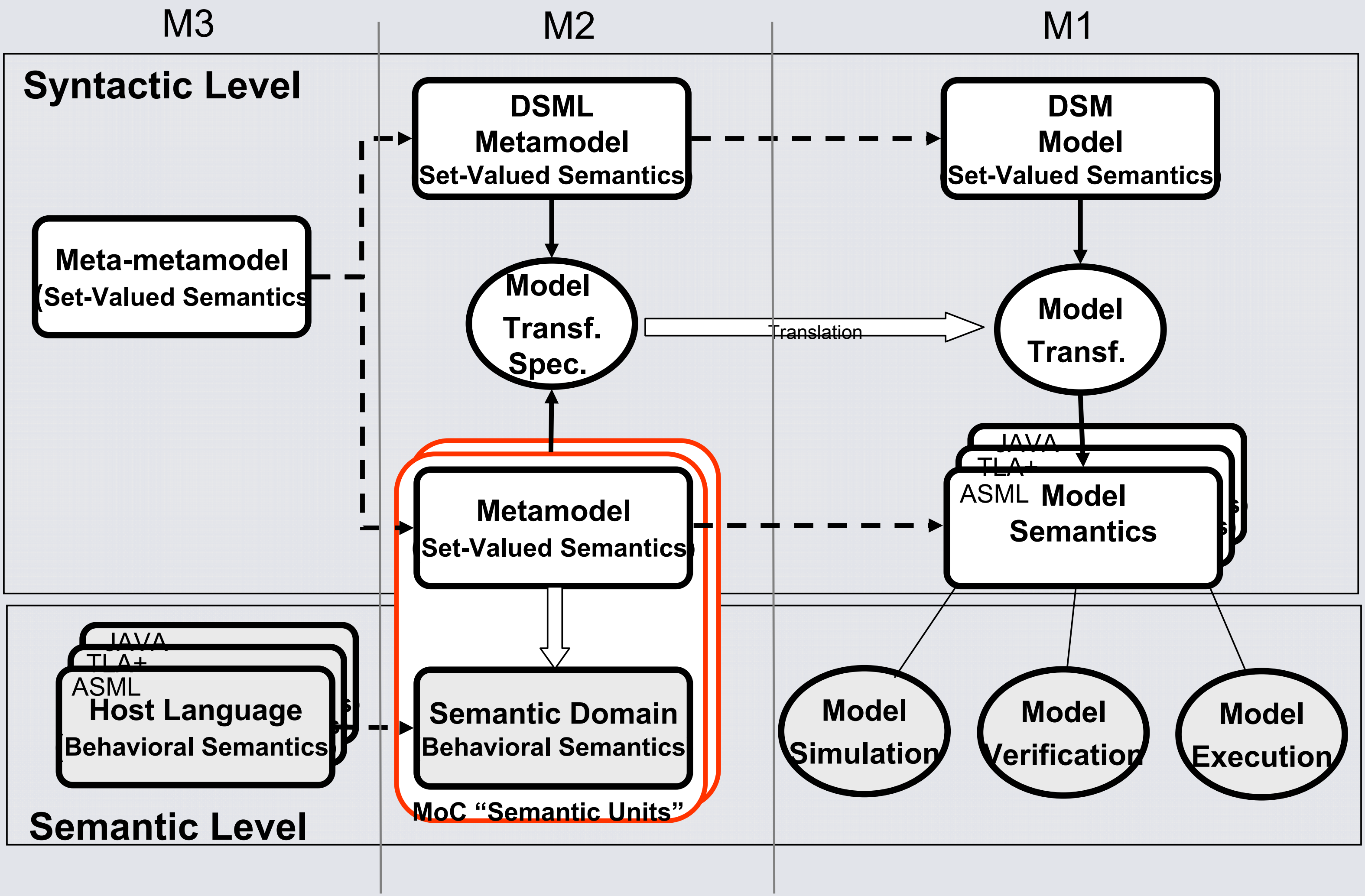
More On Semantic Anchoring

- Step 1
  - Define the DSML metamodel $\langle A, C, M_c \rangle$
- Step 2
  - Select a proper MoC as a "semantic unit" (MoC library): $L_i = \langle A_i, C_i, M_{Ci}, S_i, M_{Si} \rangle$
- Step 3
  - Anchor the semantics: $M_A = A \rightarrow A_i$
  - DSML semantics: $L = \langle A, C, M_c, S_i, M_A \circ M_{Si} \rangle$

Experimental Framework

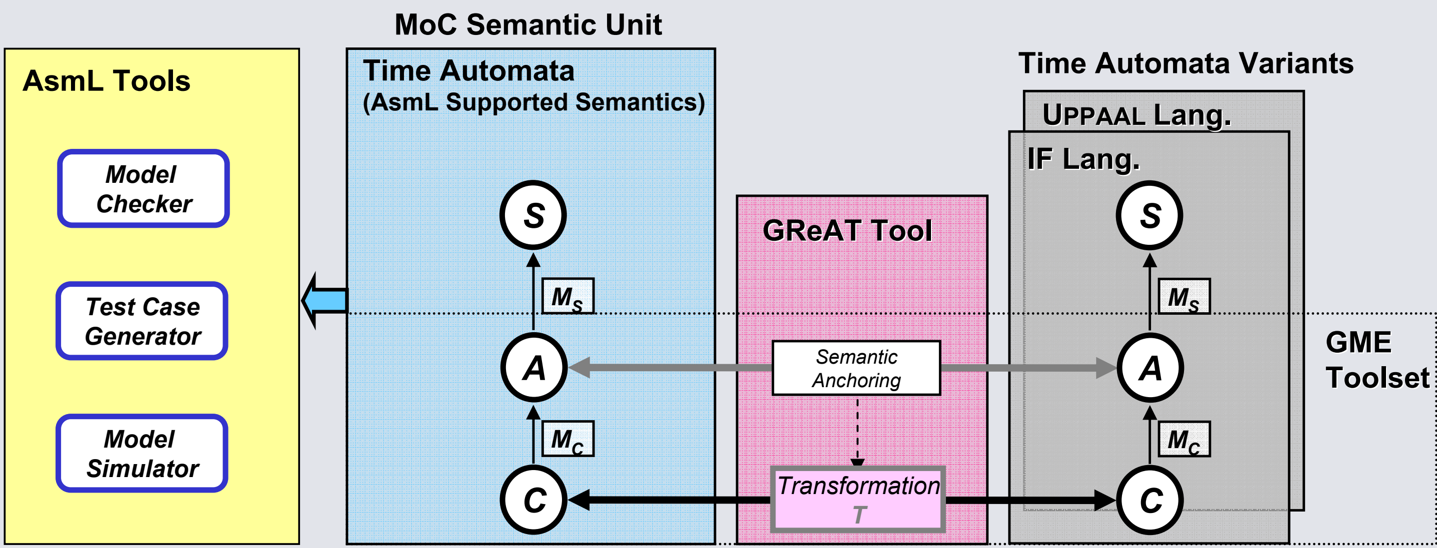


Long-Term Vision: Semantic Anchoring Architecture of DSML-s

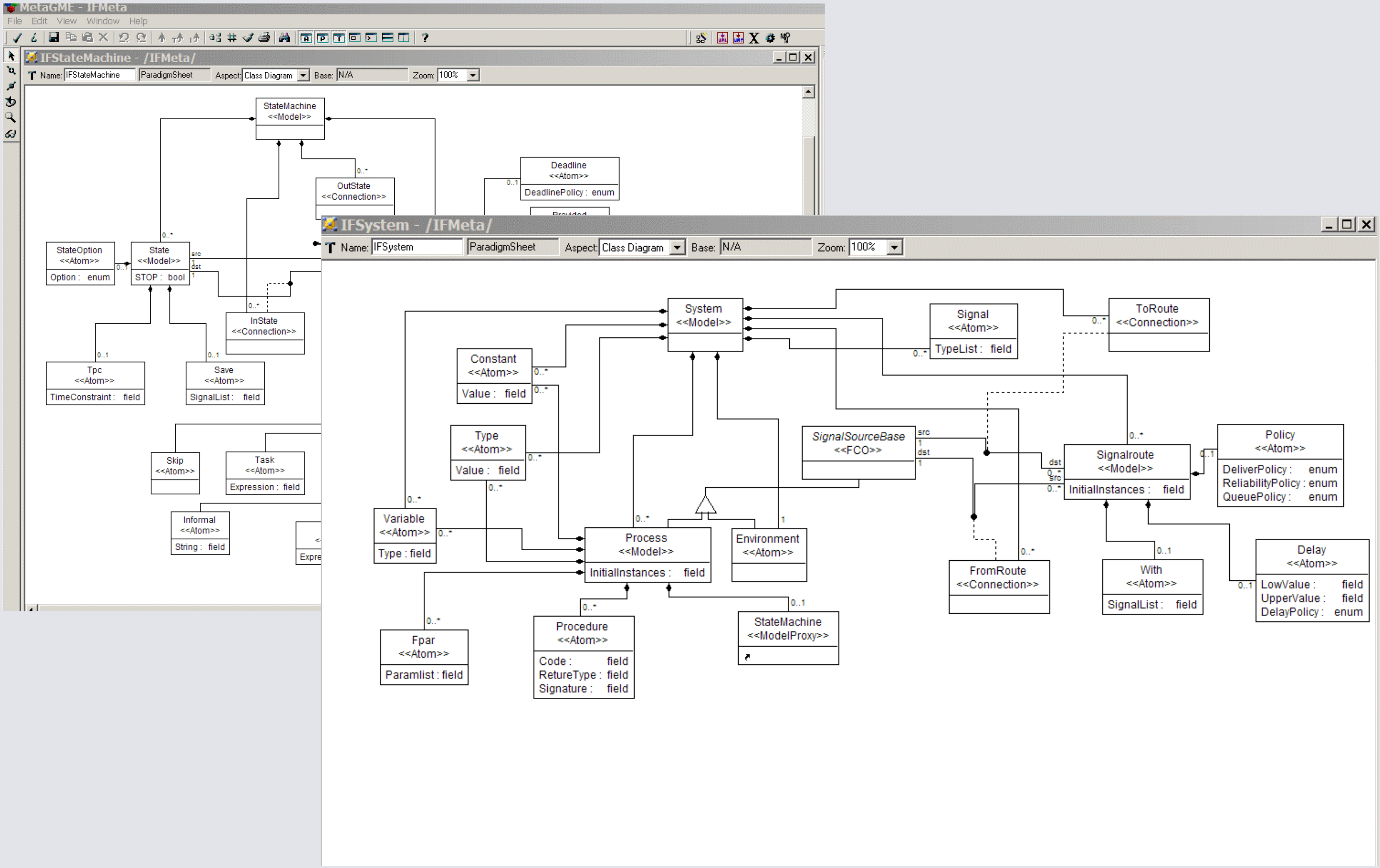


A "Semantic Unit" for Time Automata Based Modeling Languages

- Common semantic domain for varied timed automata based modeling languages: (In cooperation with VERIMAG)
  - Guard
  - Priority
  - Synchronization



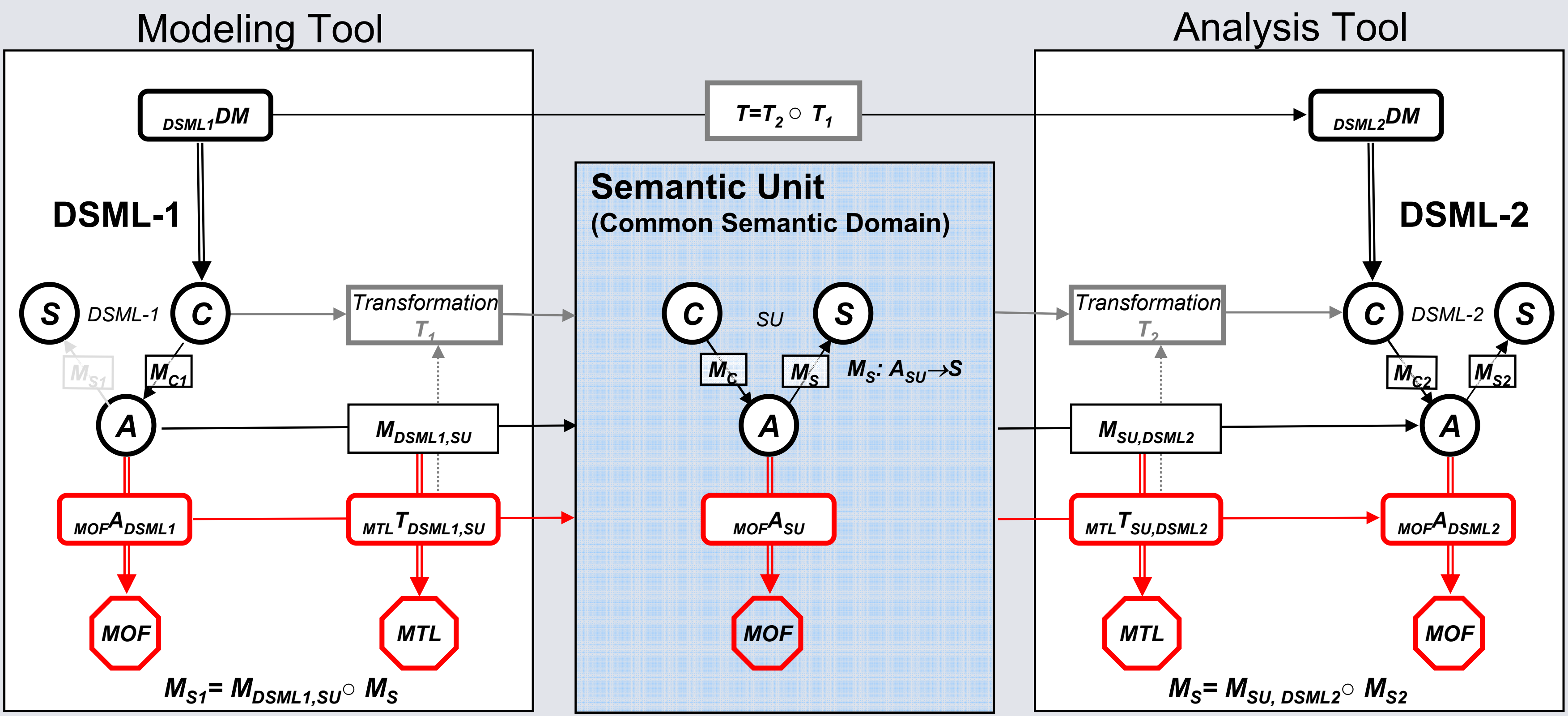
"Evidence": IF Metamodels



Metamodeling and Model Transformation Use Cases

VANDERBILT UNIVERSITY
INSTITUTE FOR SOFTWARE INTEGRATED SYSTEMS

3. Semantic Integration of Tools



Ongoing Cooperations

- Artist 2 (Prof. Sifakis, Verimag)
  - MIC-based modeling interface for IF;
  - Semantic Unit specification for Time Automata
  - Modeling Language Design Environment
- DECOS (Prof. Kopetz, TUV)
  - MIC-based modeling environment for DECOS
  - Automotive Design Tool Chain
  - DESERT Applications