

80. Software Factories - Metamodelling in Heterogeneous Technical Spaces

Prof. Dr. rer. nat. Uwe Aßmann
Institut für Software- und
Multimediatechnik
Lehrstuhl Softwaretechnologie
Fakultät für Informatik
TU Dresden
<http://st.inf.tu-dresden.de/teaching/most>
Version 19-0.2, 02.01.20

1) Heterogeneous technical spaces

Obligatorische Literatur

- ▶ Uwe Aßmann, Steffen Zschaler, and Gerd Wagner. Ontologies, meta-models, and the model-driven paradigm. In Coral Calero, Francisco Ruiz, and Mario Piattini, editors, *Ontologies for Software Engineering and Technology*. Springer, 2006.
 - http://www.springer.com/computer/swe/book/978-3-540-34517-6?cm_mmc=Google_-_Book%20Search_-_Springer_-_0
- ▶ Steffen Staab, Tobias Walter, Gerd Gröner, and Fernando Silva Parreiras. Model driven engineering with ontology technologies. In Uwe Aßmann, Andreas Bartho, and Christian Wende, editors, *Reasoning Web*, volume 6325, *Lecture Notes in Computer Science*, pages 62-98. Springer, 2010.
 - <http://www.uni-koblenz.de/~staab/Research/Publications/2010/reasoningweb2010.pdf>

Other Literature

- ▶ Kurtev, I., Bezivin, J., Aksit, M.: Technological Spaces: An Initial Appraisal. In: International Symposium on Distributed Objects and Applications, DOA Federated Conferences, Industrial track, Irvine. (2002)
- ▶ Model-based Technology Integration with the Technical Space Concept. Jean Bezivin and Ivan Kurtev. Metainformatics Symposium, 2005.

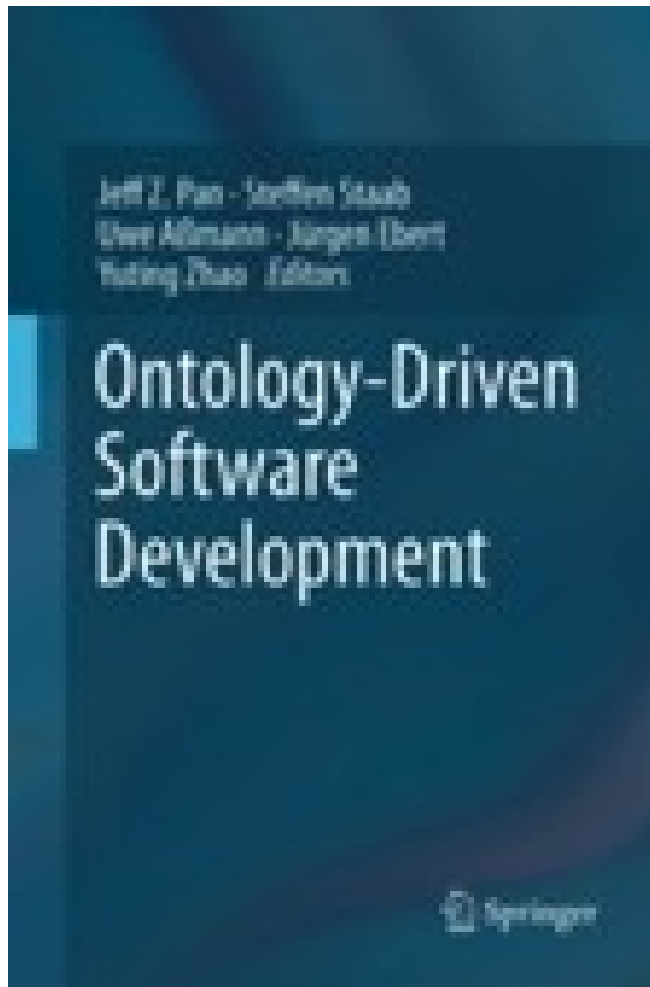
- ▶ Gašević, Dragan, Djuric, Dragan, Devedžic, Vladan. Model Driven Engineering and Ontology Development, 2nd ed., 2009, ISBN 978-3-642-00281-6
 - http://www.springer.com/computer/swe/book/978-3-642-00281-6?cm_mmc=Google-_-Book%20Search-_-Springer-_-0
- ▶ [Kendall] D. T. Chang and E. Kendall. Metamodels for RDF Schema and OWL. Proceedings of the First International Workshop on the Model-Driven Semantic Web (MDSW 2004), Monterey, USA, September 21, 2004.

- Jendrik Johannes. Component-Based Model-Driven Software Development. PhD thesis, Technische Universität Dresden, Fakultät Informatik, December 2010. <http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-63986>
 - This PhD thesis lays the ground for component models and composition technology for modeling languages.
 - www.reuseware.org
- Birgit Grammel. Automatic Generation of Trace Links in Model-driven Software Development. PhD thesis, Technische Universität Dresden, Fakultät Informatik, February 2014

The Book of the MOST Project for Multi-TS Development

5

Model-Driven Software Development in Technical Spaces (MOST)

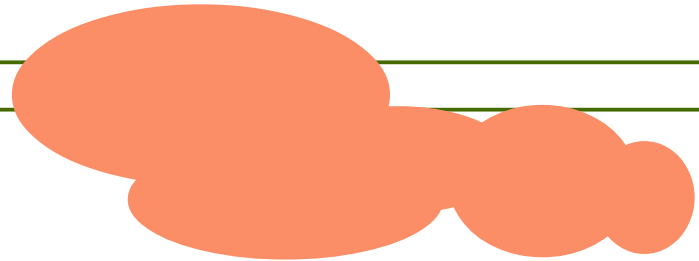


MOSTPROJECT

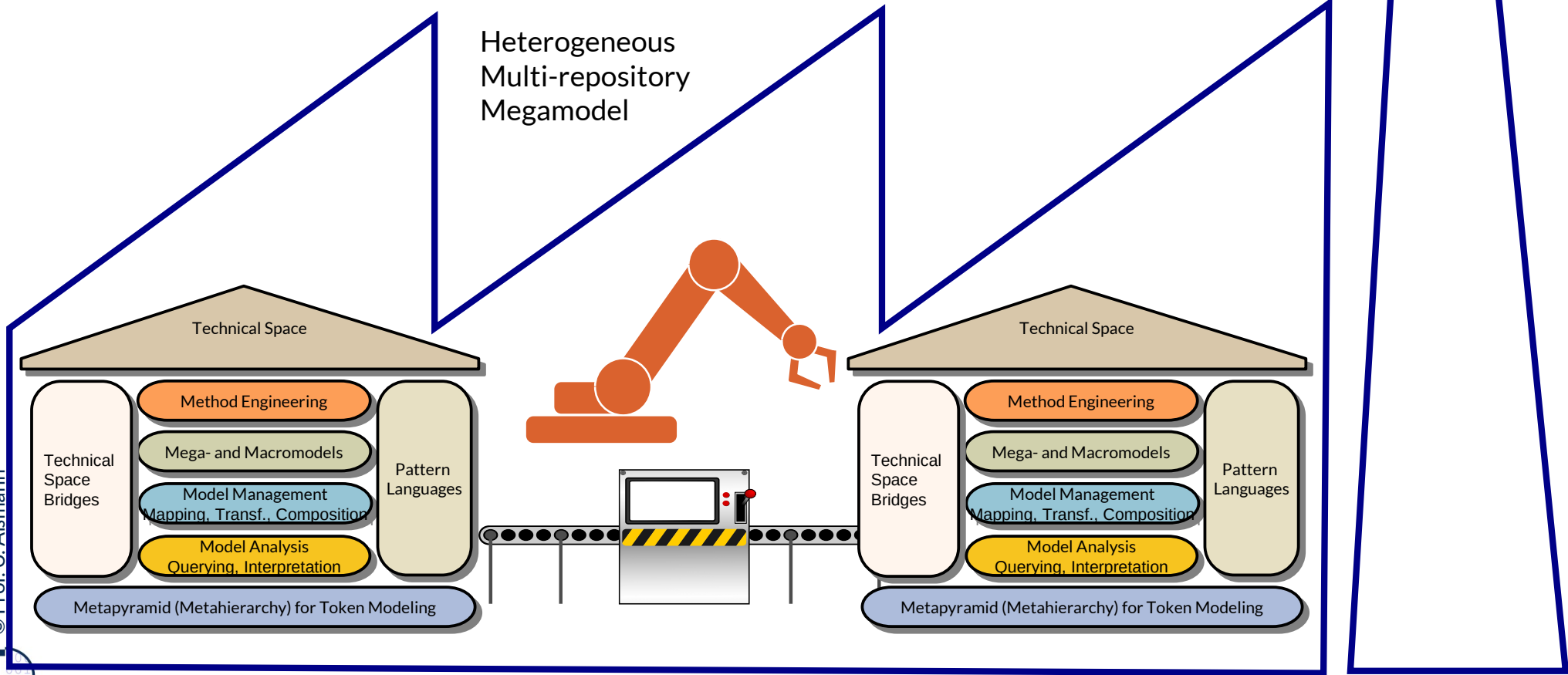
<http://most-project.eu>

80.1 MDSD in Multiple Technical Spaces (Multi-TS Development)

Q13: A Software Factory's Heart: the Multi-TS Megamodel

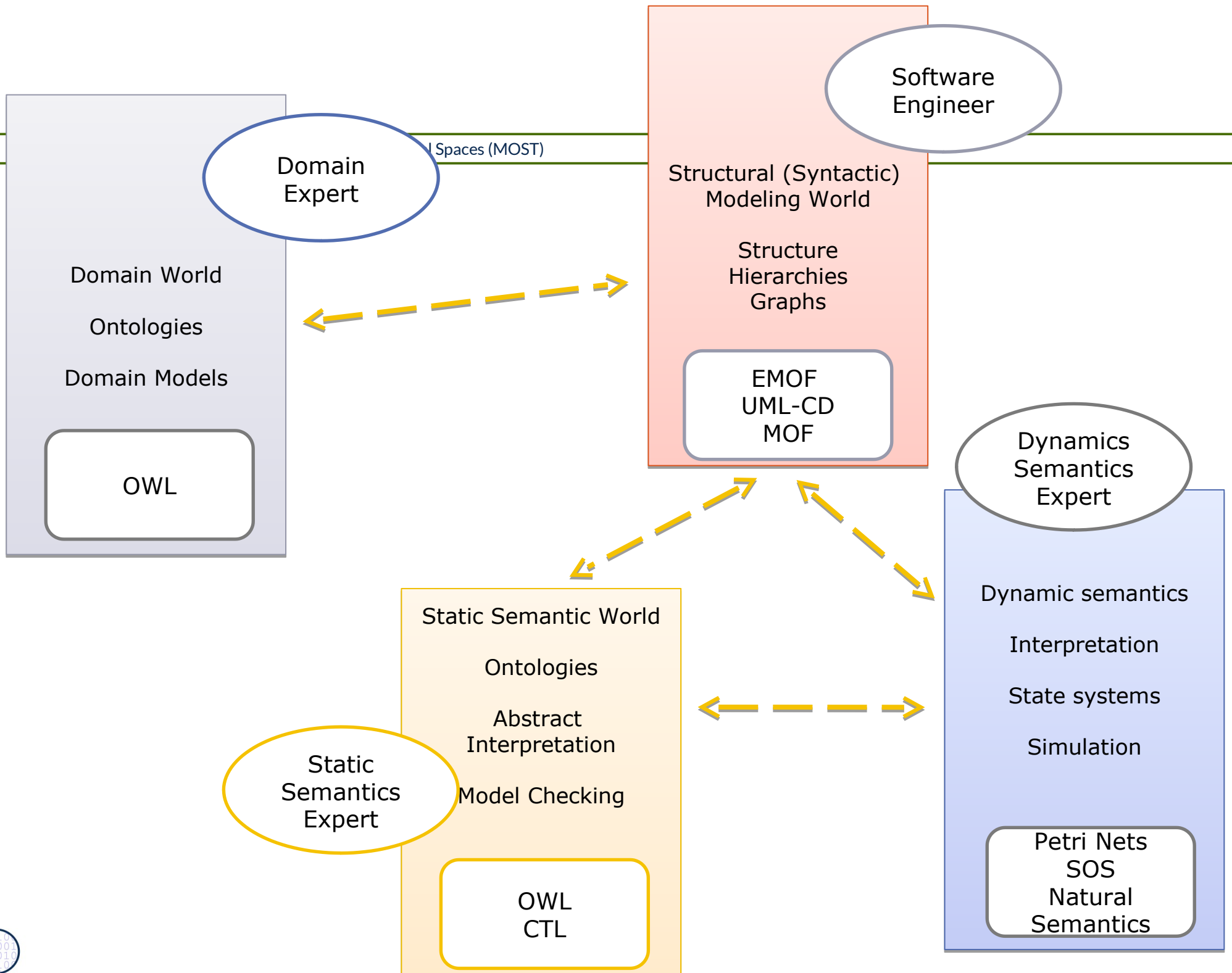


Software Factory

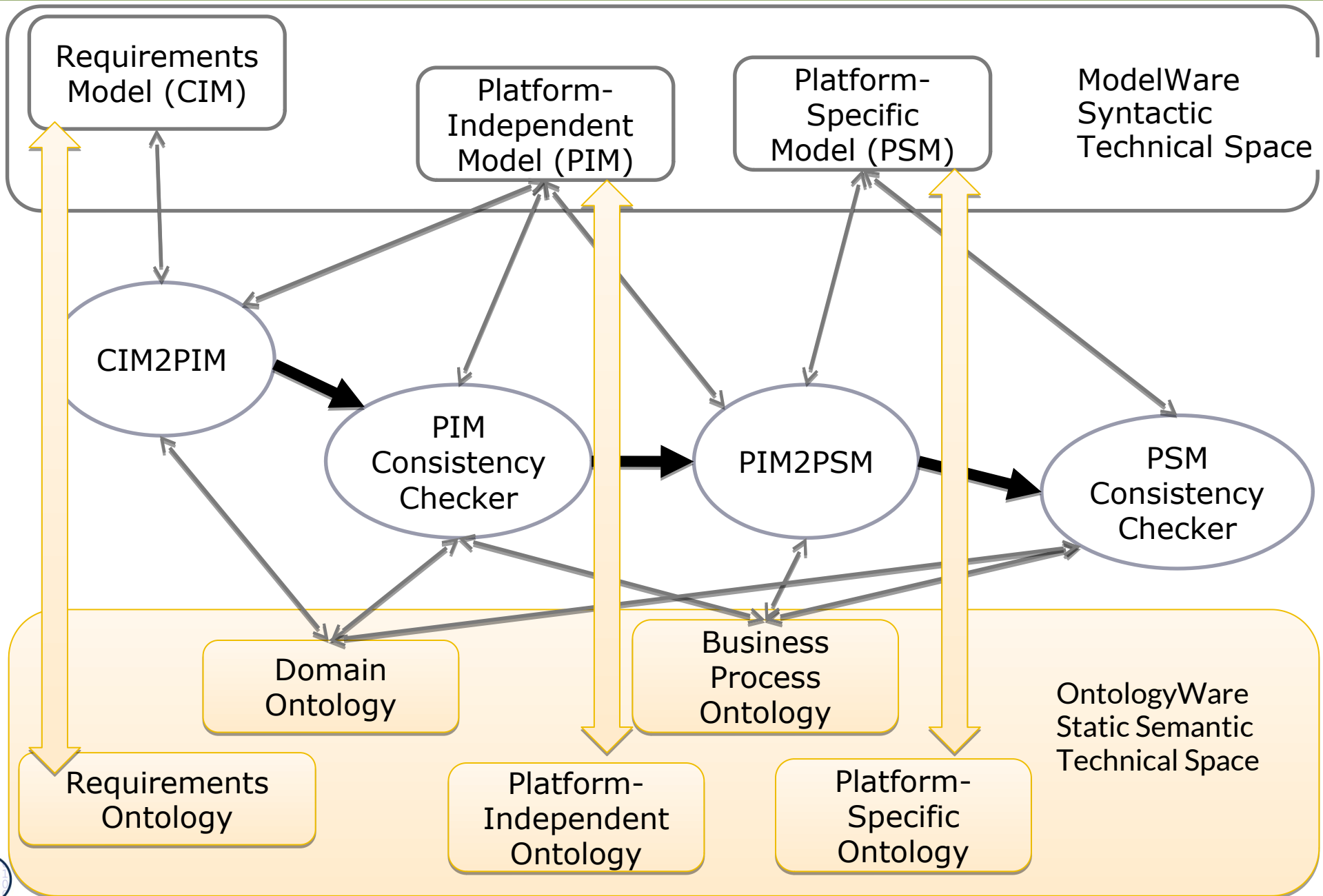


The Problem: Heterogeneous Software Lives in **Several** Technical Spaces

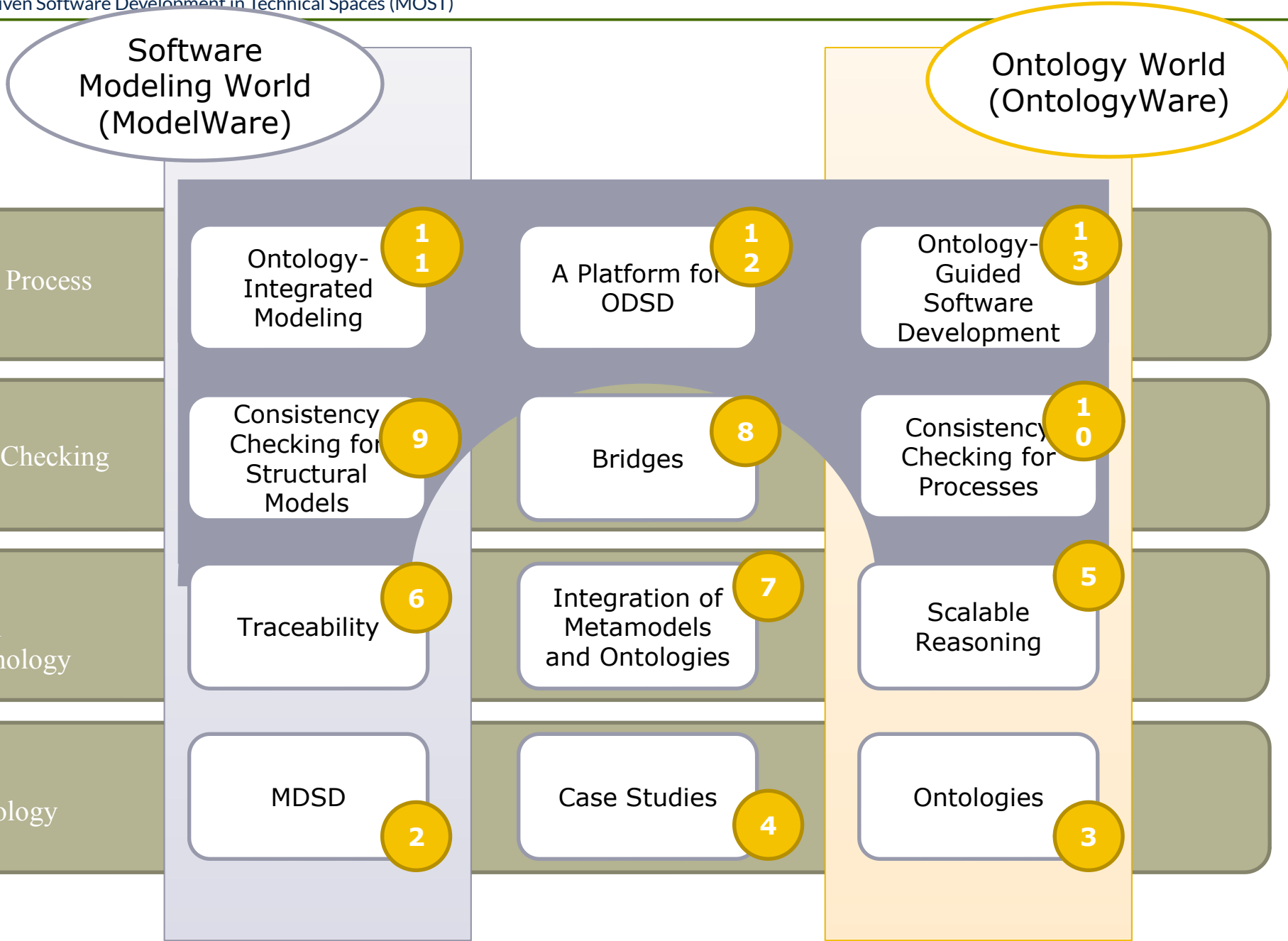
- ▶ Modern cars, cloud robots (Kiva robots), and other CPS live in several technical spaces:
 - Syntactic technical space
 - Static semantic technical spaces
 - Dynamic semantic technical space (usually one)
 - Domain world technical space



Example: A Heterogeneous MDA (From MOST Project)

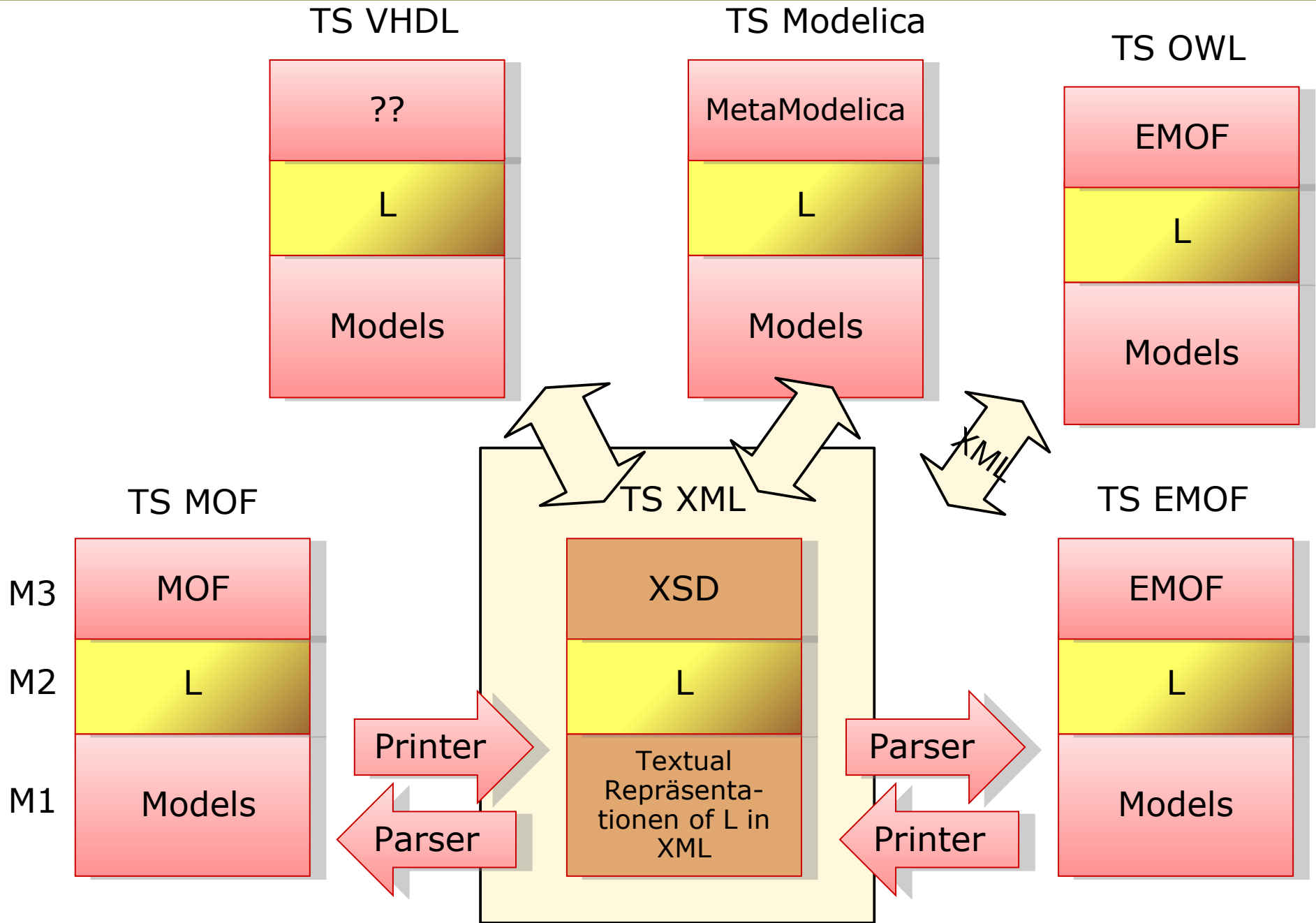


MOST: Tasks for Bridging between Syntactic and Semantic TS



80.2 Applications Working in Multiple Technical Spaces

Language Mapping between Several TS via XML Data Exchange via Link Trees



Multi-Technical-Space MDSD Tools (Software Factories)

A **multi-TS toolkit** is a toolkit using several technical spaces at the same time.

A **(heterogeneous) software factory** is a multi-TS toolkit.

- ▶ Today, most MDSD toolkits work in one technical space.
- ▶ However, industrial software development usually is heterogeneous and several technical spaces must be used (XML, Java, C++, UML, csv, ...)
- ▶ PreeVision, ASCET are software factories

A **software factory** produces heterogeneous software product lines in several technical spaces.

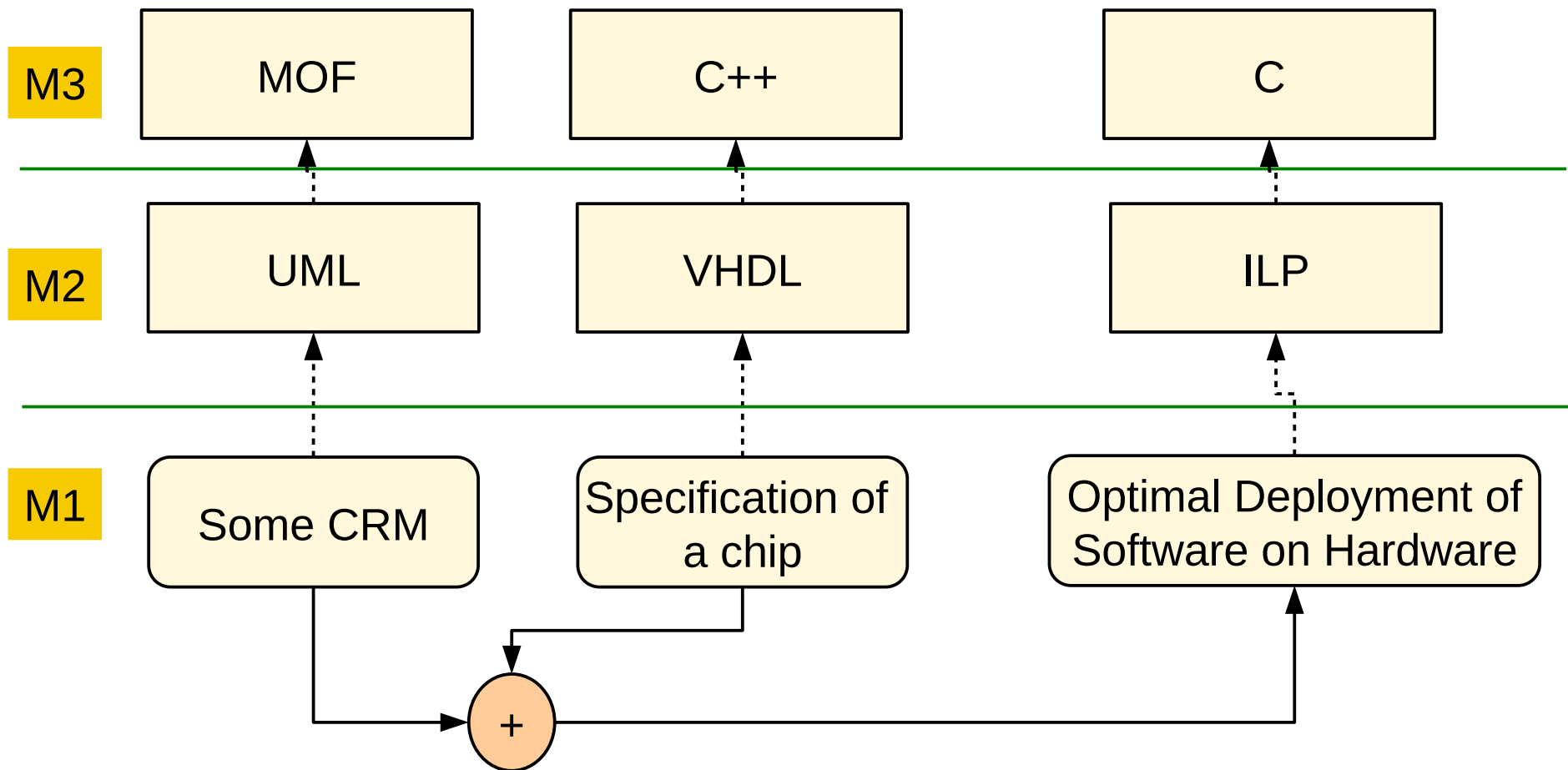
Multi-Technical-Space MDSD Tools (Software Factories)

- ▶ Bezivin's Model Engineering Metapher:
- ▶ “The world consists of different villages connected by streets. Every sort of engineer maintains one or several “model villages” (technical spaces) or “model towns” (or technological spaces)”
- ▶ The task of model engineering is to build bridges and streets in the modeling landscape

Model Engineering is the engineering of software factories – the engineering with several technical spaces in multiple technological spaces

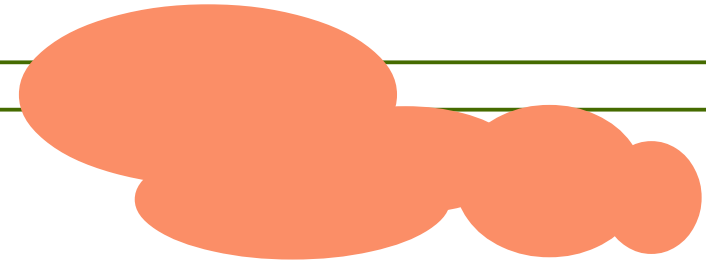
Example: Hardware Design

- ▶ To automate the optimization of software systems you need
 - A language to describe software systems (e.g., UML in MOF)
 - A language to describe hardware (e.g., VHDL in C++)
 - A language to express the optimization problem (e.g., ILP in C)

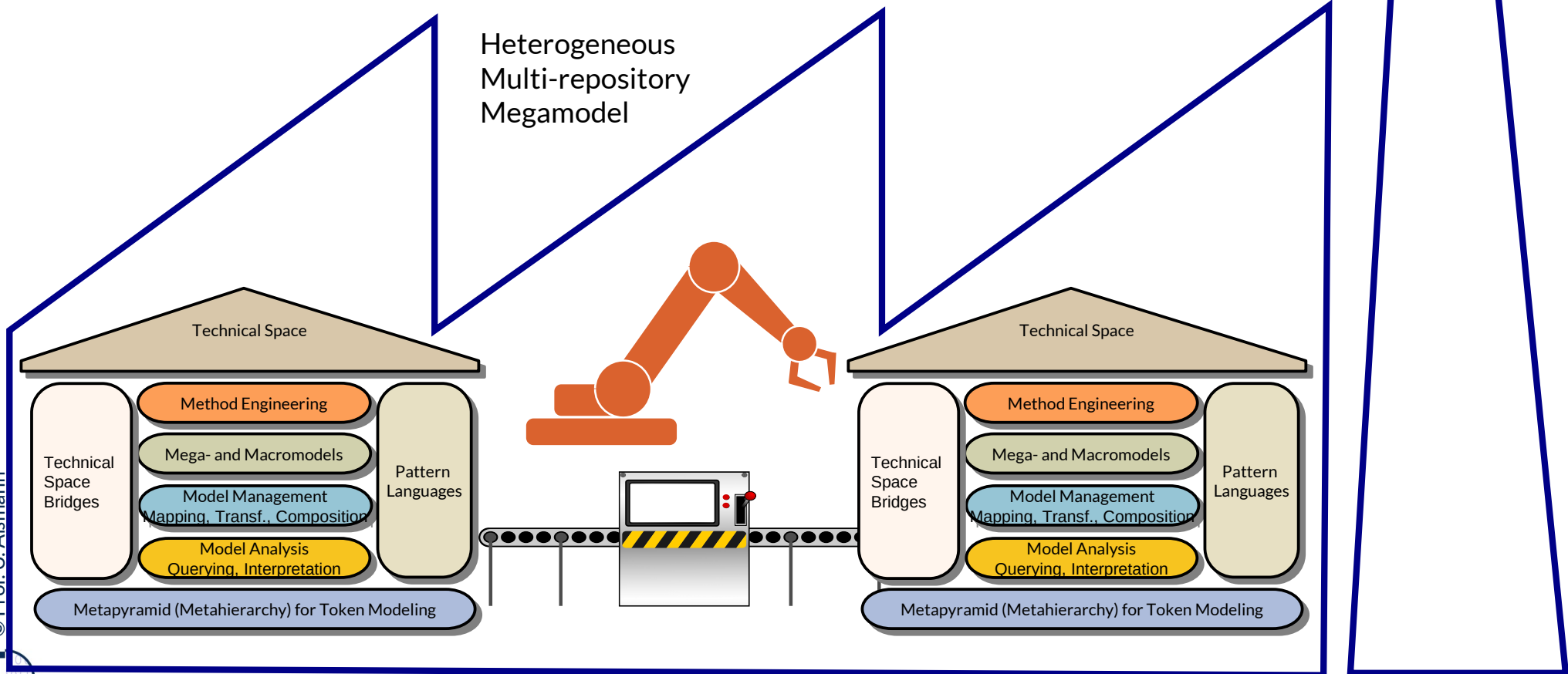


80.2 Software Factories (Wrapup)

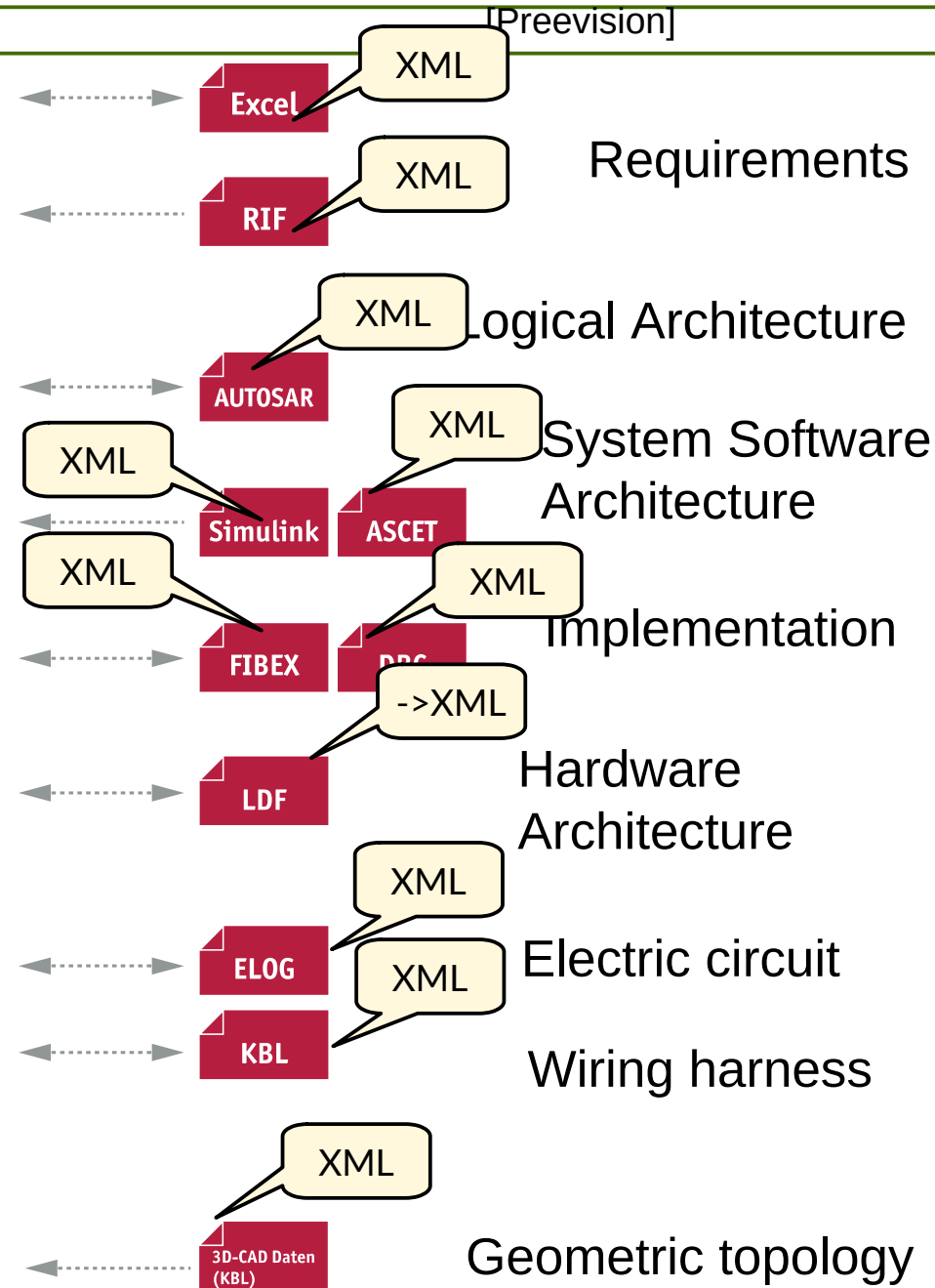
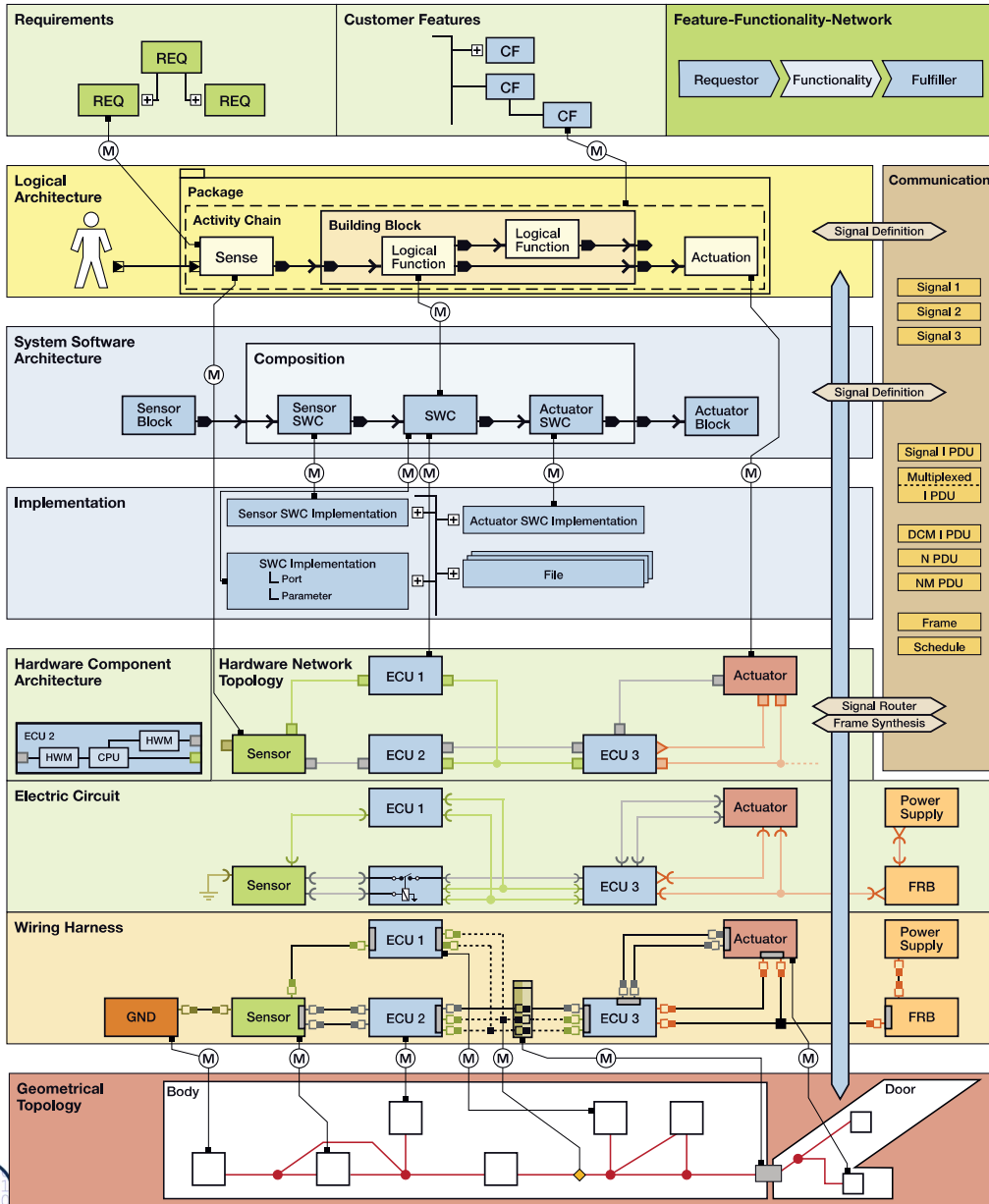
Q13: A Software Factory's Heart: the Multi-TS Megamodel



Software Factory



Remember the Big Example: Car Design with PREEVision (Vector): Interoperability with XML Link Trees



Engineering of Multi-Technical Space Megamodels

- ▶ Engineering of Technical Spaces and Megamodels is one of the most important topics of the future of software development
- ▶ Dresden has modern technologies and tools
 - Transformation tools (such as RACR)
 - Invasive composition
 - Metacomposition tools (Reuseware)
 - CROM (Role-based metalanguages)
 - Round-Trip Engineering and Role-based tools

Join us

The End

- ▶ Explain why future toolkits to design complex things will be multi-TS software factories
- ▶ What is different in the handling of a multi-TS megamodel compared to a 1-TS megamodel?
- ▶ Which technical space would you choose to exchange data in a software factory? Why?
- ▶ Why will all engineering disciplines do software factories in 50 years from now?

The End