

# Bridging Technological Spaces: Towards the Combination of Model-Driven Engineering and Ontology Technologies

---

## Zusammenfassung (Deutsch)

Die modellgetriebene Softwareentwicklung beabsichtigt die Spezifikation von Softwaresystemen durch Modelle zu vereinfachen und die automatisierte Entwicklung zu verbessern.

Die Modellierungssprachen und Werkzeuge, die zur Modellierung von Systemen und Anwendungsdomänen herangezogen werden, werden in modellbasierten technologischen Räumen zusammengefasst.

Ontologiebasierte technologische Räume enthalten Ontologiesprachen und Technologien zum Entwurf, der Anfrage und dem Schlussfolgern von Wissen.

Mit der Verbreitung des semantischen Webs werden Ontologien in der Entwicklung von Software zunehmend eingesetzt.

In dieser Arbeit werden zur Kombination von technologischen Räumen Brückentechnologien vorgestellt. Transformationsbrücken übersetzen Modelle, Abbildungsbrücken stellen Beziehungen zwischen Modellen verschiedener technologischer Räume her und Integrationsbrücken verschmelzen Räume zu neuen allumfassenden technologischen Räumen.

API Brücken erschaffen Interoperabilität zwischen Werkzeugen.

Diese Arbeit beschäftigt sich insbesondere mit der Kombination von modellbasierten und ontologiebasierten technologischen Räumen.

Nach einem Vergleich zwischen Sprachen und Werkzeugen der einzelnen Räume wird die Integrationsbrücke herangezogen um einen neuen gemeinsamen technologischen Raum zu erstellen, der den hybriden Gebrauch von Sprachen und den interoperablen Einsatz von Werkzeugen ermöglicht. Die Syntax und Semantik von Modellierungssprachen kann mit Hilfe von Ontologiesprachen spezifiziert werden. Die Korrektheit von Modellen wird durch den Einsatz von Ontologietechnologien gewährleistet. Ontologiebasierte Modellierungssprachen erlauben den Nutzen von Anfrage- und Schlussfolgerungstechnologien. Sie sind darüber hinaus so flexibel um verschiedene Anforderungen von Softwareentwicklern zu erfüllen.

Domänenspezifische Sprachen unterstützen neben der Spezifikation von Systemen auch die konzeptionelle Beschreibung von Domänen durch Modelle, die aus möglichen Laufzeitinstanzen und deren Typen bestehen. Integrierte Ontologiesprachen helfen eine formale Semantik für Domänenmodellierungssprachen zu definieren und Ontologietechnologien ermöglichen das Schlussfolgern über Typen und Instanzen.

Alle Ansätze in dieser Arbeit werden mit Hilfe eines Szenarios, in dem die Konfigurationen für Familien von Netzwerkgeräte modelliert werden, veranschaulicht. Ferner werden die Implementationen aller Brückentechnologien zur Kombination von technologischen Räumen und alle Werkzeuge für die ontologiebasierte Entwicklung von Modellierungssprachen illustriert.

## **Abstract (Englisch):**

Model-Driven Engineering (MDE) aims to raise the level of abstraction in software system specifications and increase automation in software development. Modelware technological spaces contain the languages and tools for MDE that software developers take into consideration to model systems and domains.

Ontoware technological spaces contain ontology languages and technologies to design, query, and reason on knowledge. With the advent of the Semantic Web, ontologies are now being used within the field of software development, as well.

In this thesis, bridging technologies are developed to combine two technological spaces in general. Transformation bridges translate models between spaces, mapping bridges relate different models between two spaces, and, integration bridges merge spaces to new all-embracing technological spaces. API bridges establish interoperability between the tools used in the space.

In particular, this thesis focuses on the combination of modelware and ontoware technological spaces. Subsequent to a sound comparison of languages and tools in both spaces, the integration bridge is used to build a common technological space, which allows for the hybrid use of languages and the interoperable use of tools. The new space allows for language and domain engineering.

Ontology-based software languages may be designed in the new space where syntax and formal semantics are defined with the support of ontology languages, and the correctness of language models is ensured by the use of ontology reasoning technologies. These languages represent a core means for exploiting expressive ontology reasoning in the software modeling domain, while remaining flexible enough to accommodate varying needs of software modelers.

Application domains are conceptually described by languages that allow for defining domain instances and types within one domain model. Integrated ontology languages may provide formal semantics for domain-specific languages and ontology technologies allow for reasoning over types and instances in domain models.

A scenario in which configurations for network device families are modeled illustrates the approaches discussed in this thesis. Furthermore, the implementation of all bridging technologies for the combination of technological spaces and all tools for ontology-based language engineering and use is illustrated.