

# An Ontology for Formal Methods

Martin Wirsing  
Institut für Informatik  
Universität München  
[wirsing@lmu.de](mailto:wirsing@lmu.de)

Bernd Krieg-Brückner  
Bremen Institute of Safe and Secure Systems  
Universität Bremen  
[bkb@Informatik.Uni-Bremen.DE](mailto:bkb@Informatik.Uni-Bremen.DE)

## 1 Semantic Structuring with Ontologies

The aim of the MMISS project is to set up a multi-media Internet-based adaptive system for education in the area of Formal Methods. At the core of the system is the development of lectures and classes in the area of safe and secure systems. The system provides a formal framework for integration of the teaching material based on a semantic structure for relating the different parts of the teaching material. The semantic structure is particularly important for understanding the area for formal methods, for learning the differences and similarities between the formal techniques and knowing which techniques are suitable for particular applications.

Ontologies provide the means for establishing this semantic structure. An ontology is a formal explicit description of concepts in a domain of discourse. It specifies the properties of each concept by describing the various features and attributes of the concept. Although ontologies exist for many applications we are not aware of any ontology for formal methods. However, we base our ontology on several approaches for classifying and defining topics related to formal methods such as the ACM classification scheme [ACM 98], Astesiano and Reggio's work on defining a schema for formal development techniques [AR 00], Clarke and Wing's survey on formal methods [CW 96] and Steffen's framework for formal methods tools [SMB 97].

For developing an ontology for formal methods we use the modeling language UML which is the actual de facto standard language for software development. As an ontology describes domain concepts abstractly by means of classes, subclasses and slots, UML seems to be particularly well-suited for the diagrammatic representation of the ontology [KCHDBKS 01].

## 2. The Ontology for Formal Methods

For describing the ontology of Formal Methods and its instances in UML we use class and object diagrams. The class diagrams serve as representation for the abstract notions such as Domain, Engineering Method, Formal Method, Formalism, Language and Tool. The object diagrams represent the instances of the abstract notions. For example, Computer Science topics are instances of the class Domain.

The most general notion for describing a topic for research or teaching is the notion of domain. The class of Engineering Methods (see Figure) is a specialization of the class Domain. An Engineering Method is established in the context of (zero,) one or more Domains, it has Tools which support the Method and pragmatics for applying the Engineering Method where the pragmatics are described by a Process (cf.

[AR 00]). The class Formal Method is a specialization of Engineering Method with the particular feature that any instance of Formal Method is based on a Formalism. Formal Methods are classified into Specification, Verification and Analysis Techniques (cf. [CW 96]). A Formalism has one or more associated Languages and a Theory consisting of definitions and theorems.

**References**

[ACM 98] ACM Computing Classification System [1998 Version], <http://www.acm.org/class/>  
 [AR 00] E. Astesiano, G. Reggio: Formalism and method. TCS 236, 2000, 3-34.  
 [CW 96] Edmund M. Clarke and Jeannette M. Wing: Formal Methods: State of the Art and Future Directions, ACM Computing Surveys 28 (4), 1996, 626-643.  
 [KCHDBKS 01] P. Kogut, S. Cranefield, L.Hart, M. Dutra, K. Baclawski , M Kokar, J. Smith: UML for Ontology Development, 2001.  
 [SMB 97] B. Steffen, T. Margaria, V. Braun: The Electronic Tool Integration Platform: Concepts and Design. Int. Journal on Software Tools for Technology Transfer (STTT) 1, 1997, 9-30.

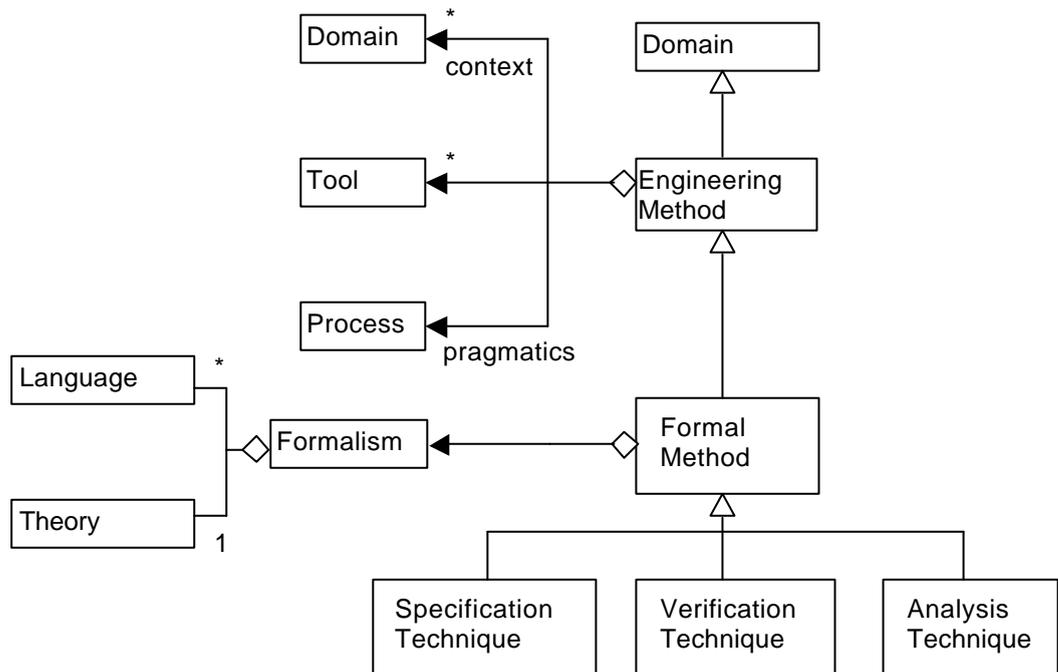


Figure: Semantic Structure of Engineering Method and Formal Method