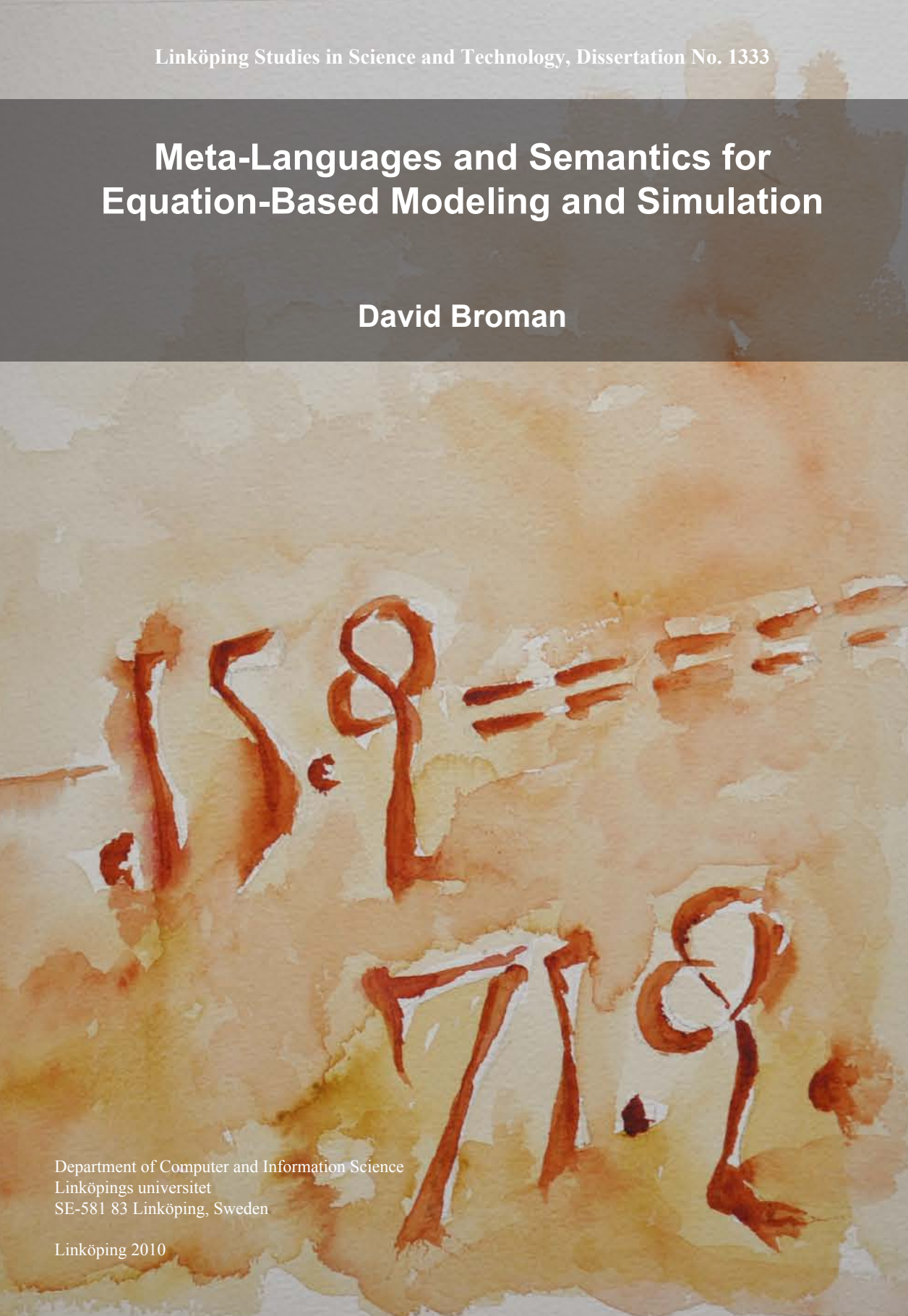


Linköping Studies in Science and Technology, Dissertation No. 1333

# Meta-Languages and Semantics for Equation-Based Modeling and Simulation

David Broman

A watercolor illustration in shades of brown and tan, featuring various mathematical symbols and equations. The symbols include a plus sign, a multiplication sign, a division sign, a square root symbol, and a pi symbol. The background is a textured wash of watercolor, with the symbols appearing as if they are floating or emerging from the paint.

Department of Computer and Information Science  
Linköpings universitet  
SE-581 83 Linköping, Sweden

Linköping 2010

Performing computational experiments on mathematical models instead of building and testing physical prototypes can drastically reduce the develop cost for complex systems such as automobiles, aircraft, and powerplants. In the past three decades, a new category of equation-based modeling languages has appeared that is based on acausal and object-oriented modeling principles, enabling good reuse of models. However, the modeling languages within this category have grown to be large and complex, where the specifications of the language's semantics are informally defined, typically described in natural languages. The lack of a formal semantics makes these languages hard to interpret unambiguously and to reason about. This thesis concerns the problem of designing the semantics of such equation-based modeling languages in a way that allows formal reasoning and increased correctness. The work is presented in two parts.

In the first part we study the state-of-the-art modeling language Modelica. We analyze the concepts of types in Modelica and conclude that there are two kinds of type concepts: class types and object types. Moreover, a concept called structural constraint delta is proposed, which is used for isolating the faults of an over- or under-determined model.

In the second part, we introduce a new research language called the *Modeling Kernel Language (MKL)*. By introducing the concept of higher-order acausal models (HOAMs), we show that it is possible to create expressive modeling libraries in a manner analogous to Modelica, but using a small and simple language concept. In contrast to the current state-of-the-art modeling languages, the semantics of *how to use* the models, including meta operations on models, are also specified in MKL libraries. This enables extensible formal executable specifications where important language features are expressed through libraries rather than by adding completely new language constructs. MKL is a statically typed language based on a typed lambda calculus. We define the core of the language formally using operational semantics and prove type safety. An MKL interpreter is implemented and verified in comparison with a Modelica environment.

