



Models as the subject of research

Jeff Gray¹ · Bernhard Rumpe²

Published online: 27 August 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

It is said that in the times before Plato, mathematics was only a vehicle to solve practical problems. Plato himself, however, identified mathematics as a pure subject of research, with the observation that the beauty of mathematical theories was of deep intrinsic value to be developed and refined independent of practical application. This spawned mathematics as a field of research, which also led to many more practical applications as a result.

Computer science has a much younger history. Some early approaches to general-purpose programming (e.g., Charles Babbage’s mechanical general-purpose computer in the 1800s, the Analytical Engine, and Ada Lovelace’s first algorithms) are the most well-known examples. However, computer science (or better: informatics) really emerged in the twentieth century. Informatics started as a tool to help with other science and mathematics investigations. When the increasing complexity became clear, informatics became the target of its own research area with beautiful results in various subdomains.

Modeling, however, is much older. Conceptual models are omnipresent in philosophy, physics, chemistry, and many other inquiry-based disciplines. Physical manifestations of models were used frequently in building construction. The term “model” had an early historical connection to the construction of churches, where a model was a 1:10 reduced wooden version of the church to be built. From the great inventor Da Vinci, we still have many drawings that model surprising and smart machines—even though many of them have never been built. The authors of this editorial, however, are not aware of the topic of modeling as its own subject or research area in these earlier centuries.

The concept of a model needed more formalization when a specific semantics was required, such as modeling of systems and software. The most widely used definition of “model” was coined by Stachowiak only in 1973. Programming theory and semantics definitions needed a precisely defined notion of a well-formed piece of code. This notion also carried over to other forms of digitally communicated models. This includes Petri Nets in their various forms, automata and statecharts, class diagrams, action languages, and other forms of modeling languages. The formal methods domain (including logic) and the programming language domain (in particular, compiler construction) in informatics were the earliest to put some focus on the value of models and explicitly defined modeling languages. Software engineers and database administrators also relied upon models in various representations, focusing on the practical use of restricted forms of models. Interestingly, modeling tools have been a core product offered by commercial vendors since the beginning of the software industry. In fact, the first software product sold independently of a hardware package was Autoflow, which was a flowchart modeling tool developed in 1964 by Martin Goetz of Applied Data Research.

The use of models was widely discussed in the software engineering domain during the 1980s and 1990s, where a variety of different modeling languages (e.g., Booch’s clouds and Rumbaugh’s object models) were adopted. Around 1994, the overlap of common concepts across the various modeling languages pointed to the need for a unification of the different languages. The “methods wars”, which were intensely discussed at the OOPSLA conferences during the 1990s, were finally resolved into a standardization effort, first called the “Unified Method” and later leading to the “Unified Modeling Language” (UML). During this crucial period of unification, it became clear that defining such a standard would not be an easy task. A research community emerged that became interested in studying models and the UML as a core research subject area of its own.

In 1998, Jean Bézivin and Pierre-Alain Muller organized the first UML workshop, “The Unified Modeling Language. «UML»’98: Beyond the Notation” in Mulhouse, France,

✉ Bernhard Rumpe
bernhard.rumpe@sosym.org

Jeff Gray
jeff.gray@sosym.org

¹ University of Alabama, Tuscaloosa, AL, USA

² RWTH Aachen University, Aachen, Germany

June 3–4, 1998. Based on the success of the scientific event and the obvious need for further discussion and investigation, Robert France and Bernhard Rumpe (the founding creators of SoSyM; after Jean Bézivin came up with the idea) organized the follow-up event “«UML»’99—The Unified Modeling Language. Beyond the Standard” as the first international conference specific to software modeling, held in Fort Collins, Colorado (USA) in October 28–30, 1999. Some general questions that were discussed during this nascent period of modeling research included:

- What is a good UML modeling language?
- Concentration on core modeling constructs or elaborate comfort within UML?
- How to integrate orthogonal sublanguages?
- Semantics of UML sublanguages?
- How to compose, refine, and translate UML models?
- How to build reliable tooling for the UML with degrees of automation?
- How to teach UML?
- What are appropriate methods that use UML?
- How to use UML for variability modeling, components, or frameworks?
- How to positively improve the standard definition?

In 2005, the UML series of conferences was broadened to the general question of what defines a good modeling language. The focus remained on the development of software, but because software usually is embedded in a social, technical, or business context, various domains inside and related to informatics were more deeply involved. Therefore, the 2005 edition of the conference was renamed to “Model Driven Engineering Languages and Systems” using the acronym “MoDELS/UML” and held in Montego Bay, Jamaica, in October 2005. The above questions were discussed not only for the UML standard, which had at that time become relatively stable (and incomplete in terms of practical adoption, with many existing UML tools only partially satisfying many of the users needs). The MODELS conference name change also recognized the emergence of domain-specific modeling languages (DSMLs) that were not specifically tied to the UML. The increased interest in DSMLs helped to add a new research focus on language definition and semantics, as well as the usage and engineering of such languages (e.g., for testing, code synthesis, simulation, and artifact transformation).

Many recent publications use modeling as a contribution to a specific application domain, but much of the modeling research literature also concentrates on modeling languages as a core subject of study (rather than just a vehicle to study something else). As a consequence, many new conferences over the past decade have emerged that are centered around modeling themes, which have been

hosted by SoSyM as special issues (e.g., ECMFA, ICMT, and the DSM workshop series). We can safely state that since the early UML discussions at OOPSLA in the 1990s and the start of the MoDELS/UML series of conferences, models and modeling languages have become the subject of study of research on their own, leading to hundreds of Ph.D. dissertations across the world. An emerging focus has been to transfer the research results into practice by building industrial-strength and robust commercial modeling tools.

It is equally important to understand what models, denoted in explicit and purposeful modeling languages, can do for engineers or scientists beyond pure software development. It is time to understand how the digital models of informatics can be merged with the continuous calculus of control theory, the geometric models of mechanical engineering, and how to use behavioral modeling with uncertainty, underspecification, variants, and high-configurability of models at runtime. Physical and chemical laws can be understood as models of the world, and software controlling some part of it needs an integrated understanding.

With only a few decades of existence, there is much for the modeling research community to do. This is why we expect SoSyM to grow further and to become and remain the primary venue of articles focusing on modeling and modeling languages—on a more formal, as well as applied with a technological focus. We look forward to the future submission of exciting research to SoSyM!

1 Content of this issue

This issue contains an Expert’s Voice and 12 Regular papers as follows:

- *Expert’s Voice*
 - “Contents for a Model-Based Software Engineering Body of Knowledge” by Antonio Vallecillo, Loli Burgueño, Federico Ciccozzi, Michalis Famelis, Gerti Kappel, Leen Lambers, Sebastien Mosser, Richard Freeman Paige, Alfonso Pierantonio, Arend Rensink, Rick Salay, Gabriele Taentzer, and Manuel Wimmer.
- *Regular Papers*
 - “Multi-view approaches for software and system modeling: a systematic literature review” by Antonio Cicchetti, Federico Ciccozzi, and Alfonso Pierantonio.
 - “Privacy-enhanced BPMN: enabling data privacy analysis in business processes models” by Pille Püllonen, Jake Tom, Raimundas Matulevicius, and Aivo Toots.

- “Trade-off analysis for SysML models using decision points and CSPs” by Patrick Leserf, Pierre De Saqui-Sannes, and Jerome Hugues.
- “A comparative study of students and professionals in syntactical model comprehension experiments” by Mohamed El-Attar.
- “Modeling compliance specifications in linear temporal logic, event processing language and property specification patterns: a controlled experiment on understandability” by Christoph Czepa, Amirali Amiri, Evangelos Ntentos, and Uwe Zdun.
- “Integrated revision and variation control for evolving model-driven software product lines” by Felix Schwägerl and Bernhard Westfechtel.
- “Improving manual reviews in function-centered engineering of embedded systems using a dedicated review model” by Marian Daun, Thorsten Weyer, and Klaus Pohl.
- “Model execution tracing: a systematic mapping study” by Fazilat Hojaji, Tanja Mayerhofer, Bahman Zamani, Abdelwahab Hamou-Lhadj, and Erwan Bousse.
- “ChronoSphere: a graph-based EMF model repository for IT landscape models” by Martin Hausler, Thomas Trojer, Johannes Kessler, Matthias Farwick, Emmanuel Nowakowski, and Ruth Breu.
- “Broadened support for software and system model interchange” by Catalina Lladó and Connie Smith.
- “An approach for bug localization in models using two levels: model and metamodel” by Lorena Arcega, Jaime Font, Oystein Haugen, and Carlos Cetina.
- “A UML Profile for the Design, Quality Assessment and Deployment of Data-intensive Applications” by Diego Perez-Palacin, Jose Merseguer, Jose Requeno, Michele Guerriero, Elisabetta Di Nitto, and Damian Tamburri.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.