WoM⋆: An Open Interactive Platform for Describing, Exploring, and Sharing Mathematical Models

Jean-Marie Gaillourdet[1], Thomas Grundmann[1], Martin Memmel[2], Karsten Schmidt[1], Arnd Poetzsch-Heffter[1], and Stefan Deßloch[1]

[1] Department of Computer Science, University of Kaiserslautern, Germany
{jmg,thg,kschmidt,poetzsch,dessloch}@cs.uni-kl.de
[2] Knowledge Management Department, DFKI GmbH, Trippstadter Str. 122, D-67663 Kaiserslautern, Germany and
University of Kaiserslautern, Germany
martin.memmel@dfki.de

Abstract. Mathematical models play an increasingly important role in science and engineering. In this paper, we present the *WoM*, a platform for building up knowledge repositories for searching, exploring, combining, and sharing such models. In contrast to similar efforts, *WoM* supports a well-defined semi-structured representation for mathematical models, which acts as a solid foundation for intelligent web-based presentation, browsing, search, simulation/visualization, and Web community capabilities. We envision *WoM* to provide a foundation for future design flows and engineering processes using standardized, composable, and computer-supported models.

1 Introduction

Mathematical models play an increasingly important role in science and engineering. Supported by the power of modern software and computer technology, they allow for a better analysis, visualization, and comprehension of natural phenomena and designed artifacts. According to Pieter Eyckhoff [1], a mathematical model is "a representation of the essential aspects of an existing system (or a system to be constructed) which presents knowledge of that system in usable form". Following this explanation, we distinguish three kinds of models:

– *Descriptive* models explain the essential aspects of existing systems such as physical, sociological, or economical systems.
– *Constructive* models describe systems to be constructed as part of engineering tasks or processes.
– *Abstract* models are used for modeling a certain class of phenomena, but are not (yet) applied to a specific system. Examples are special classes of differential equations or labeled transition systems.

⋆ The project *WoM* is supported by the Rhineland-Palatinate research center $(CM)^2$ – Center for Mathematical and Computational Modelling.

The Web of mathematical models, *WoM*, is a platform to host such models on the Web. The mission of *WoM* is to help improving the accessibility, usability, precision, tool support, classification, and comparability of mathematical models. Further goals are to improve interfacing with, composition of, and intelligent search for models. To achieve these goals, we address two central questions: How should mathematical models be represented? How should they be managed and presented in an open model platform?

Model representation. In our approach, a model and its representation should combine several aspects reflecting different requirements:

1. An *informal description* introduces the model and explains which phenomena or artifacts are modeled. It may be supported by graphical or video visualizations of the model.
2. The *mathematical description* characterizes the model in terms of its mathematical properties. It explains the parameters of the model. In principle, it should be expressible in a formal language.
3. *Software support* allows simulating the model with different parameter settings. The software should be realized in a component-based way such that models can be composed.
4. *Linking and metadata* relate the model to other models and related documents, and provide support for classification and structured search.

To address these requirements, we developed a structured model representation schema. Based on this schema, we started to build up a model repository and a community. The community will be crucial for the process of refining, extending, and strengthening the mathematical models as well as the underlying schema.

Model platform. The Model platform allows to store, relate, query, search, use, and compose models. We have identified four central requirements for such a platform:

1. Community support: Construction, collection, classification, and linking of models is only possible with support from a user community. Accordingly, the platform should encourage participation, and foster the development of a self-sustainable community.
2. Model construction: The platform should support the schema-conform construction and modification of models. It should combine wiki functionality with an offline editing possibility. In particular, models need a platform independent representation that can be down- and uploaded.
3. Web accessibility: Models and all their functionality should be accessible and usable on the Web. In particular, simulations should be possible without download and shareable between users.
4. Evolvability: To stepwise realize our vision, many changes of the schema for model representation and ontologies for classifications have to be managed in the future. In particular, the existing model representations have to evolve together with the schema and ontologies in a consistent way. This is only achievable with mechanical support by the platform.

Overview and vision. This paper describes the current state of development of the *WoM*. At the heart of our approach is a semi-structured, schema-based representation of mathematical models, which is described in Section 2. An overview of our *WoM* platform approach and its main components is provided in Section 3. We specifically focus on the model repository, the model visualization and simulation component, and the Web access and community support. Related work is discussed in Section 4.

Our goal is to make the notion of "mathematical models" more explicit and precise and to build up knowledge repositories for searching, exploring, combining, and sharing such models. We envision *WoM* to provide a foundation for future computer-guided design flows and an intelligent engineering support for standardized, composable, and computer-processable models.

2 A Schema for Models and their Relations

At the heart of our approach is a structured representation of models. After a thorough discussion with our first user community, we started with a still simple schema for models that addresses the requirements listed in Section 1. Having an explicit, content-related schema distinguishes the *WoM* approach from classical wiki platforms and enables stronger computer-support for model use and management. We envision that the schema will be refined in the future (see below).

A model in the *WoM* has to follow the *XModel schema*, which defines an XML representation of a model. The components of an XModel are shown in Fig. 1 and described in the following paragraphs.

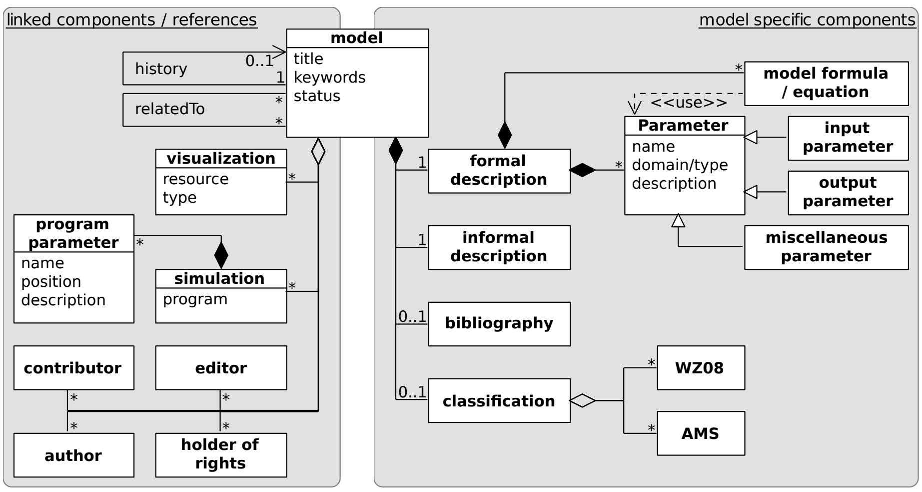

Fig. 1. The XModel schema

An XModel – an instance of the XModel schema – has to have a *title* and consists of an *informal description* as well as a *formal description*. While the informal description is a textual representation of the model (optionally with images), the formal description defines a model using mathematical expressions.

The formal description consists of a set of *input parameters* I, *output parameters* O, and *miscellaneous parameters* M. The latter may serve as constants or other variables. Each parameter is defined by a *name*, a *domain/type*, and optionally a short (informal) *description*. The central part of the formal model description are the *model formulas/equations* that define the relationships between the parameters. All parameters should occur in the formulas or equations.

In addition to these mandatory components, a model may also contain *visualizations* and *simulations*. The visualizations can range from simple images to complex videos. Simulations are provided by software packages that allow for online experimentation with the model. Their parameters should correspond to the parameters of the formal model; additional parameters to control particular aspects of the simulations are possible. Different techniques for implementing simulations are supported (see Section 3).

The XModel schema also includes metadata such as the model's *authors*, *contributors*, *holders of rights*, and *editors*. Other metadata are *keywords*, the model's *status* that can be stable, experimental or checked, and a *bibliography*, whose structure is based on the BibTeX syntax. Using a *classification*, a model can be related to standard classifications. Currently, *WoM* supports the AMS 1991 Mathematical Subject Classification and the WZ08 - Classification of Economic Activities. A model may also be *related to* other models in the *WoM*, which is a symmetric relation.

The XModel schema is designed in a way that allows to evolve the models over time. For example, models may be classified according to a new classification ontology or may be extended with new simulations and visualizations. Of course, changes in other parts of the model are possible as well. Consequently, a model may have different versions and provides access to its *history*.

The XModel schema is internally defined based on XML Schema. As LaTeX serves as the default input language (in particular for formulas), we also provide LaTeX support for the XModel schema.

3 A Platform for the Web of Models

Addressing the platform requirements mentioned in the introduction, we developed a web-based platform[3] for the *WoM* to view, store, edit, annotate, manage, and explore models. This section describes its architecture and explains further aspects of the model repository, the integration of visualization and simulation, and the community support.

[3] A snapshot of a development prototype is available here: `http://angren.cs.uni-kl.de/WoMstatic/`. Use *womguest* as username and password to login.

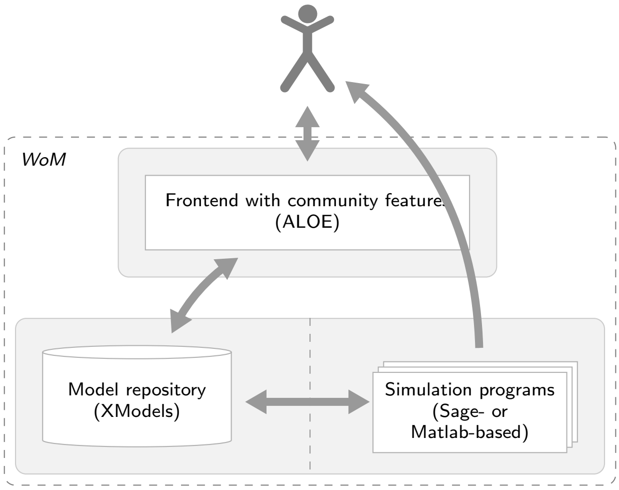

Fig. 2. Architecture of the Web of Models

3.1 The Architecture of the Web of Models Platform

The architecture is depicted in Fig. 2. It is centered around a repository supporting storage, editing, and versioning of models and their relationships (see Section 3.2). We expect the use of a schema to be crucial in order to support *evolvability*, i.e., the evolution of the form of model representations and their relationships. Every evolution step of the schema has to be complemented with a transformation that updates all models in the repository such that they conform to the new schema.

The platform also provides an environment to execute model-specific *simulation* programs. The (underlying) gray rectangles denote possibly distributed computation sites where these simulations are hosted (cf. Section 3.3).

Community features and *Web accessibility* are provided by a separate component (see Section 3.4). This component also manages further model annotations that are added by end users outside the scope of the model representation.

3.2 Model Repository

The model repository stores the models and their different versions and integrates them into a linked web. This is done (1) by using the metadata in the models and (2) by using the external classifications. The relationships between models and their integration into the classifications support browsing and search of related models.

The separation of model repository, user management, and storage of user generated annotations facilitates the investigation of evolution aspects of models. Furthermore, we work on the long-term goal of combining models and their simulation programs in an automatic way. A well-defined data representation is crucial for this.

3.3 Visualization and Simulation

Mathematical models are often quite complex. Therefore, the need to provide tools for better understanding, easier application and simpler checking of applicability of a model arises naturally when creating or working with a descriptive mathematical model. Consequently, an environment designed to store and publish such models should provide such capabilities. *WoM* supports these use cases by allowing model representations to be enhanced by visualizations and simulation programs. Usually, the model author provides them together with the model. Whereas visualizations in form of graphics or video are essentially an enhancement of the informal description, the simulation programs allow exploring the model with different parameter sets and visualizing the results. The *WoM* supports two kinds of simulation programs:

- pre-fabricated simulation programs
- open simulation programs

Pre-fabricated simulation programs are embedded into the Web user interface of a model. They accept a set of parameters and produce results that are presented to the user, e.g. as a picture or a movie. *Open* simulation programs are embedded into an interactive environment. It allows the user to explore and to experiment with the model in interactive sessions, in which the simulation and visualization procedures can freely be called by the user. We have realized such an integration by adapting the Sage framework [2]. It provides a web-based Python interpreter and a very rich library of mathematical algorithms and data structures.

In the current implementation, the model repository and simulation environment run on the server infrastructure of the *WoM*. For the next version, we plan to support the physical and organizational separation of the model repository and simulation programs. This would allow model authors to keep their simulation programs private, e.g., for IP reasons. Or they are simply in a better position to provide the necessary computational resources to execute a simulation program. Also, decoupling the simulation environment from the model repository simplifies security and scaling issues, as well as adaptation to new languages, libraries or other tools used in the implementation of simulation programs.

The fact that *WoM* provides the possibility to integrate simulation programs enables authors to write remotely usable simulation programs without in-depth knowledge of Web technologies. Since we anticipate authors to be mathematicians or engineers, but not Web developers, we expect that *WoM* simplifies the publishing of models for them.

3.4 Community Support and Web Accessibility

Through the integration of the ALOE system into the *WoM* infrastructure, Web accessibility is ensured, and a variety of social media technologies are provided. ALOE[4] is a web-based and generic social resource sharing platform developed

[4] `http://aloe-project.de`

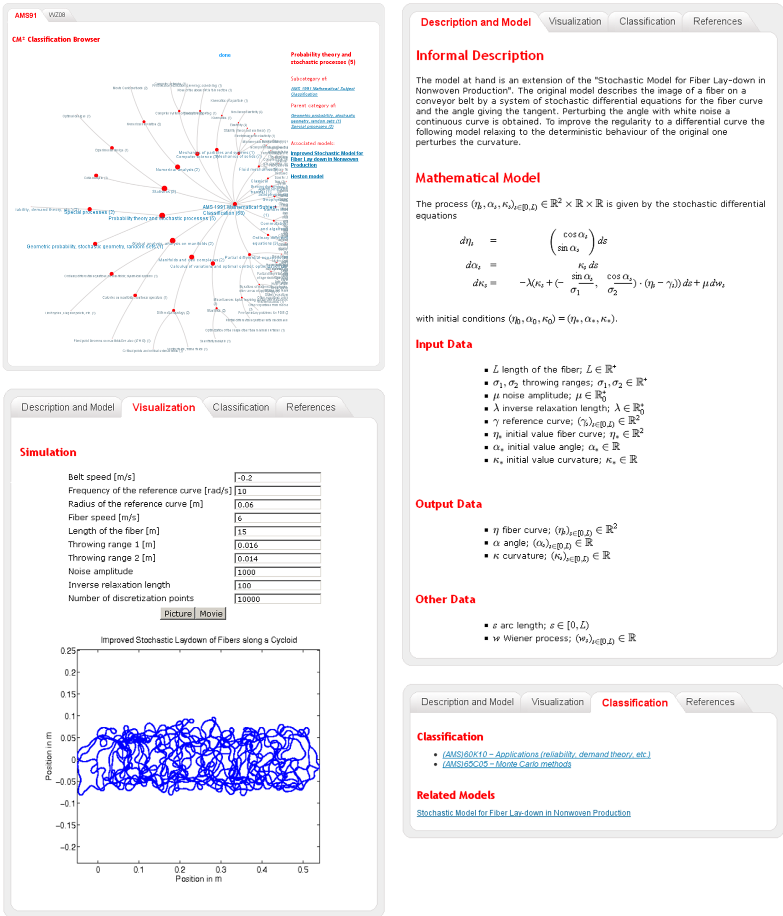

Fig. 3. Screenshot of the interactive classification browser and several facets (description, visualization, and classification) of a mathematical model within *WoM*

at the Knowledge Management group of DFKI. It allows for contributing, sharing, organizing, and accessing arbitrary types of digital resources such as text documents, music, or video files. Users are able to either upload resources (using the system as a repository) or by referencing a URL (using the system as a referatory). Furthermore, the platform offers common user management features and Web 2.0 interaction possibilities: users can tag, rate, and comment on resources, they can maintain resource portfolios, join and initiate groups, get reports about activities by means of feeds or information mails, etc. Moreover, using groups that can be open or closed (i.e., joining requires the permission of a group administrator), it is possible to defined fine-grained access rights for resources and annotation facilities. Thus, model contributors can decide who is able to see, annotate or enhance their models.

To allow the usage of ALOE in as many scenarios as possible, and to foster the adoption of as many users as possible, ALOE was designed as a server-based application where information is exchanged via HTTP. On the one hand, the system's functionalities are offered via a graphical user interface that can be accessed with any common Web browser that can connect to the ALOE server. On the other hand, a Web Service API is offered that allows to access the ALOE functionalities. Therefore, SOAP was chosen as a standard and platform-independent protocol[5]. ALOE thus realizes a *social backbone* that allows to introduce social media paradigms in existing (heterogeneous) infrastructures (see [3,4,5] for more information about the system architecture and use cases).

For *WoM*, a specifically adapted instance of ALOE is being developed. Among others, these adaptations concern the integration with the *WoM* model repository, rights management issues, support for the visualization of mathematical models, and browsing facilities for the supported classifications (see Fig. 3).

4 Related Work

There are several active and less active communities bringing math into the Web. A proprietary platform based on Mathematica/WorlframAlpha is available under [6]. The main focus is on providing visual aids for mathematical algorithms. Each (often interactive) *demonstration* is based on a CDF file (computable document format), which can only be processed by proprietary tools. Some unstructured text comments are attached to them as well as some author information. Demonstrations may point to other "related" demonstrations, but not necessarily vice versa. The platform is missing any community features, model structuring, and alternative visualization capabilities.

Based on several freely available software projects, a number of math-related wikis have emerged. Many of them heavily use the OMdoc [7] format[6] and the OpenMath and MathML standards [8]. However, this documentation format concentrates on the representation of mathematical equations, their transformation, and referencing. Sample projects are [9] (outdated) and [10] – "The Encyclopedia Sponsored by Statistics and Probability Societies". These platforms provide AMS classifications, user communities, and LaTeX-based integration of new content. However, "models" are still unstructured, i.e., plain LaTeX documents and visual aids are limited to simple images. Another prototype *JOBAD* (JavaScript API for OMDoc-based Active Documents) [11] concentrates on the "Web" part of this platform. The integration of other (ad-hoc called) Web Services linked via keywords or explicit user input, on-the-fly conversion of units, or document visualization style are the main goals of this server-client approach for math documents. Unfortunately, there are no efforts to integrate interactive models or to structure them in any way.

Similar to a math encyclopedia, PlanetMath [12] collects all kinds of math-related and LaTeX-sourced documents and makes them available as HTML pages.

[5] see `http://www.w3.org/TR/soap` for more information

[6] OMdoc is an inter lingua for mathematical communications.

Although the models can be downloaded in an intermediate XML format, there is no clear structuring to support model linkage. Only keywords, which are automatically extracted from the text, are used for references – wiki style. Some community features such as comments and history are supported, but with the exception of images, no visual aids or interactive playgrounds are provided.

Platforms such as ActiveMath [13] may communicate with computer algebra systems or formalize mathematical expressions in order to annotate or simply present them in the Web. They may also provide learning platforms allowing flash programs and applets to be embedded. Special platforms focus on proof languages and proof checker capabilities for all math-related expression (see e.g. [14]). But they do not support the notion of mathematical models and their relationships, nor do they provide any community features.

Similar to *WoM*, the OKSIMO project (formerly known as Plath Earth Simulator [15]) wants to model day-to-day problems and make the partially interactive models available on the Web. However, a propriety Fcl input language, i.e., a visual programing language, is required to submit new models. The platform does not provide a structured model repository like *WoM*.

In summary, the described approach have a different focus, but usually share some technical aspects. For instance, JOBAD uses a similar model repository approach, namely TnTBase – a database assembled from Subversion and Berkeley DB XML. Except platforms aiming at formalizing mathematical expressions, all (web-oriented) platforms support LaTeX-based inputs without any pre-defined additional structure. A distinguishing feature of the *WoM* approach is that it supports its models/entries by an explicit schema.

5 Conclusions

We presented the Web of Models, a platform for storing, searching, exploring, and sharing mathematical models. For exploration, the platform supports the simulation and visualization of models. Models are represented as instances of the so-called XModel schema. From a conceptual point of view, the schema is helpful to define the scope of the *WoM* and to support the comparison and classification of models. This is important for standardizing models, for intelligent search, and for integrating models into computer-guided engineering processes. From a technical point of view, it improves model consistency, enables to generate documentation of the simulation procedures from the model description, and forms the backbone for the automated transformation of the model repository in the contexts of evolution steps.

Our next practical goals are to enlarge the user community and to increase the model collection. In the future, we plan to investigate the refinement of the schema in three ways:

- a deeper integration of models and simulations
- a further formalization of the models by expressing them in a formalized language of set theory or higher-order logic

– the introduction of compositional models, i.e., models that are composed from other models by linking input and output parameters or by combining structural models with flow models.

Another area of future work is to transfer the structured wiki approach to other areas, e.g., to software components.

Acknowledgements Many thanks go to the authors of the mathematical models currently available in *WoM* and to our students contributors Johannes Schildgen, David Poetzsch-Heffter, and Mathias Weber.

References

1. Eyckhoff, P.: System Identification: Parameter and State Estimation. J. Wiley (1974)
2. Stein, W., et al.: Sage Mathematics Software. The Sage Development Team. (2011) `http://www.sagemath.org`.
3. Memmel, M., Tomadaki, E., Wolpers, M.: An approach to enable collective intelligence in digital repositories. In Luca, J., Weippl, E.R., eds.: Proceedings of the World Conference on Educational Multimedia, Hypermedia & Telecommunications, ED-MEDIA 2008, Vienna, AACE (2008) 1803–1811
4. Wolpers, M., Memmel, M., Giretti, A.: Metadata in architecture education - first evaluation results of the mace system. In Cress, U., Dimitrova, V., Specht, M., eds.: EC-TEL 2009 - Learning in the Synergy of Multiple Disciplines. Volume 5794 of Lecture Notes in Computer Science., Springer (2009) 112–126
5. Memmel, M., Schirru, R.: Sharing digital resources and metadata for open and flexible knowledge management systems. In Tochtermann, K., Maurer, H., eds.: Proceedings of the 7th International Conference on Knowledge Management (I-KNOW), Know-Center, Graz, Journal of Universal Computer Science (September 2007) 41–48 ISSN 0948-695x.
6. Wolfram: Wolfram demonstrations project `http://demonstrations.wolfram.com`.
7. Kohlhase, M.: Omdoc: Open mathematical documents. In Vries, F., Attwell, G., Elferink, R., Tödt, A., eds.: Open Source for Education (November 2005) 137–143
8. Caprotti, O., Carlisle, D.: Openmath and mathml: semantic markup for mathematics. Crossroads **6** (November 1999) 11–14
9. MathWeb: `http://www.mathweb.org`.
10. StatProb: The encyclopedia sponsored by statistics and probability societies `http://statprob.com`.
11. Giceva, J., Lange, C., Rabe, F.: Integrating web services into active mathematical documents. In Carette, J., Dixon, L., Coen, C., Watt, S., eds.: Intelligent Computer Mathematics. Volume 5625 of LNCS. Springer Berlin / Heidelberg (2009) 279–293
12. PlanetMath: `http://planetmath.org`.
13. Melis, E., Goguadze, G., Libbrecht, P., Ullrich, C.: ActiveMath – a learning platform with semantic web features (2009) `http://activemath.org`.
14. Naumowicz, A., Kornilowicz, A.: A brief overview of Mizar. In Berghofer, S., et al., eds.: TPHOLs 2009. (2009) LNCS 5674.
15. Doeben-Henisch, G.: The planet earth simulator project - a case study in computational semiotics. In: AFRICON, 2004. 7th AFRICON Conference in Africa. Volume 1. (September 2004) 417–422 `http://oksimo.inm.de/`.