The Web of Mathematical Models*: A Schema-based, Wiki-like, Interactive Platform

Thomas Grundmann, Jean-Marie Gaillourdet, Karsten Schmidt, Arnd Poetzsch-Heffter, Stefan Deßloch
Department of Computer Science
University of Kaiserslautern, Germany
{thg,jmg,kschmidt,poetzsch,dessloch}@cs.uni-kl.de
Martin Memmel
Knowledge Management Department, DFKI GmbH, Trippstadter Str. 122
D-67663 Kaiserslautern, Germany and
University of Kaiserslautern, Germany
martin.memmel@dfki.de

Abstract

In science and engineering mathematical models are increasingly important to describe natural phenomena and design artifacts. Our goals is to make the notion of "mathematical models" more explicit and precise as well as to build up knowledge repositories for searching, exploring, combining, and sharing models. With the *Web of Mathematical Models*, *WoM*, we provide a platform to host such models on the Web. Models follow an explicit, content-related schema.

1 Introduction

In science and engineering mathematical models are increasingly important to describe natural phenomena and design artifacts. Not least because of the power of modern software and computer technology, which allows for better analysis, visualization, and comprehension of natural phenomena and designed artifacts. Based on Eykhoff's definition, in which 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" [6], we distinguish three kinds of models:

- *Descriptive models* that explain the essential aspects of existing systems such as physical, sociological, or economical systems;
- *Constructive models* that describe systems to be constructed as part of engineering tasks or processes and
- *Abstract models* that 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.

Our goal is to make the notion of "mathematical models" more explicit and precise as well as to build up knowledge repositories for searching, exploring, combining, and sharing such models. With the *Web of Mathematical Models*, *WoM*, we provide a platform to host such models on the Web. Its mission is to help improving the accessibility, usability, precision, tool support, classification, and comparability of mathematical models, and thus, *WoM* provides a foundation for future computer-guided design flows and an intelligent engineering support for standardized, composable, and computer-processable models. To achieve our goals, we had to answer the following two central questions:

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

1. How to represent mathematical models?
2. How to manage and present them in an open model platform?

To answer the first question, we established a small initial user community of applied mathematicians and identified the following requirements:

- A model should have an *informal description* that introduces the model and explains which phenomena or artifacts are modelled. It may be supported by graphical or video visualizations of the model.
- A *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.
- *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.
- *Linking and metadata* relate the model to other models and related documents, and provide support for classification and structured search.

These requirements are fulfilled by an XML-based schema. This schema does not only structure the model definition, but also relates it to visualizations – graphics or videos – and (interactive) simulation programs. It supports two kinds of simulations:

- *pre-fabricated simulations* that are embedded into the Web user interface and create – based on a set of parameters – a result, e.g., a picture or a movie, and
- *open simulation programs* that are embedded into an interactive environment, which allows the user to experiment with and to explore the model in interactive sessions.

WoM provides the possibility to integrate 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.

To answer question two, we identified four requirements which enable us to store, relate, query, search, use, and compose models:

- *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.
- *Model construction:* The platform should support the schema-conformal construction and modification of models. It should combine Wiki functionality with an offline editing possibility. In particular, models need a declarative and platform independent representation that can be down- and uploaded.
- *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.
- *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.

Addressing these last four requirements, we developed a web-based model platform[1]. It is centered around a repository for our schema-based model representation. We expect the schema to be crucial for evolvability.

Web accessibility is ensured by the integration of the ALOE system into the *WoM* infrastructure. ALOE[2] is a web-based and generic social resource sharing platform developed at the Knowledge Management group of DFKI[3]. ALOE allows contributing, sharing, organizing, and accessing arbitrary types of digital resources such as text documents, music, or video files. Users are able either to upload resources or to reference them using URLs. Furthermore, the platform offers common user management features and Web 2.0 interaction possibilities like tagging, rating, and commenting on resources.

This paper focuses on the schema used for the representation of mathematical models (Section 2). It extends the overview paper [7]. In Section 3, we discuss related work. Section 4 contains the conclusion.

2 A Schema for Models and their Relations

Our approach is built around a structured representation of models. Based on an XML-based schema, we started to build up a model repository. Having an explicit, content-related schema distinguishes the approach of *WoM* from classical Wiki platforms and enables stronger computer-support for model use and management. The schema is called XModel Schema and is internally defined by an XML Schema. LaTeX serves as the default input language – in particular for formulas. Thus, we provide a LaTeX implementation of the XModel Schema. Future versions may also support other input languages like MathML. In this section we introduce the current state of the XModel Schema.

The *XModel Schema* defines an XML representation of a model. The components of an XModel – an instance of the XModel Schema – are shown in Figure 1 and described in the following paragraphs.

Describing a Model. Besides a *title*, an XModel essentially consists of an informal description and a formal description. The *informal description* is basically a textual representation of the model. It is a compound of *paragraphs*, *lists*, (simple) *tables*, and *images*. In addition, elements to *emphasize* content and for *referencing* are supported as well as *mathematical expressions*. Although they are permitted, these expressions do only have a descriptive character and are not associated with any special role.

In contrast, the *formal description* defines a model using mathematical expressions, which have special meaning. They are used to define the model's 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 a short textual *description*. These parameter blocks are followed by the *model's definition*, which describes how the model is implemented using the beforehand defined parameters. Therefore, this consists of text content similar to the informal description's one. To (re)use these parameters, special elements to refer to a parameter's name, domain/type, and description are added (*parameterName*, *parameterType*, *parameterDescription*). But the essential part of the definition block is a set of *model equations/formulas* R that define the relationships between the parameters, such that for each $o \in O$ at least one equation/formula $r(I' \cup M') \in R$ exists with $I' \subseteq I$ and $M' \subseteq M$. R may also contain equations/formulas that allow to bind some miscellaneous parameters $m \in M$ for further use in other equations.

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

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

[3] German Research Center for Artificial Intelligence

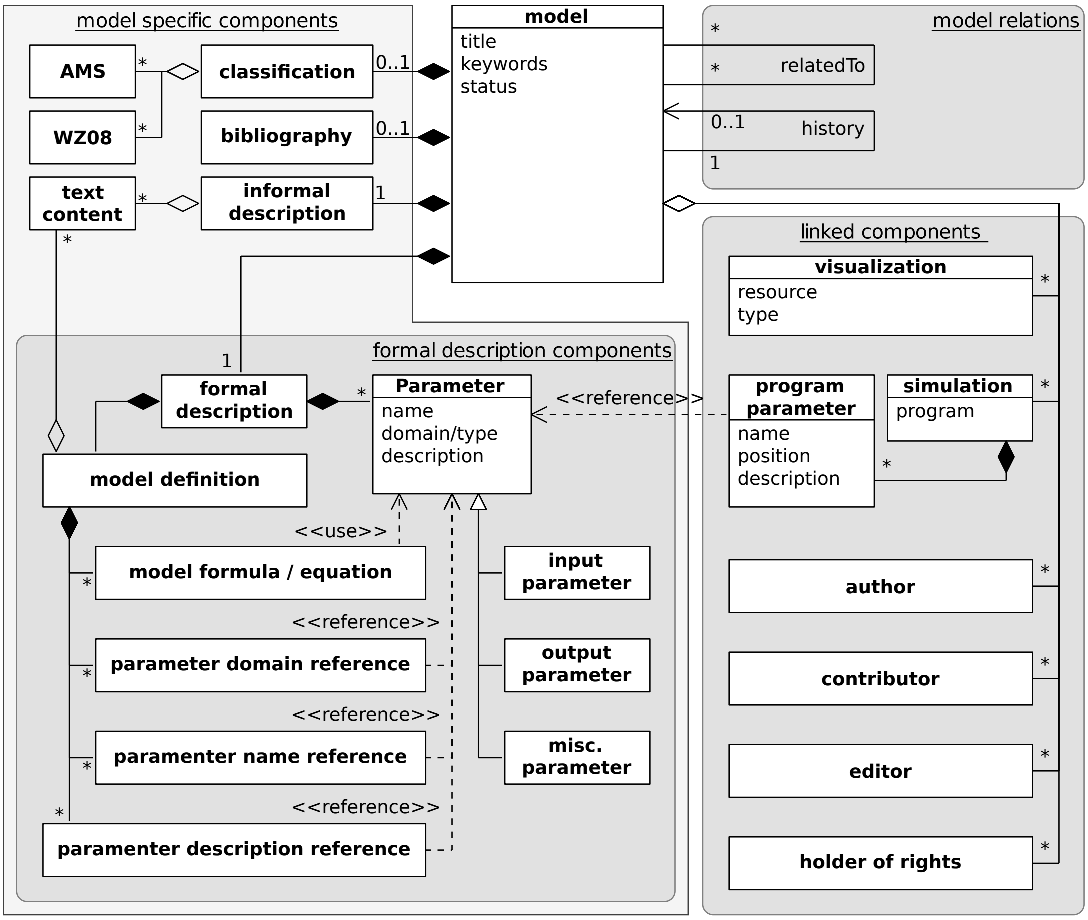


Figure 1: The XModel Schema

Visualizations and Simulations. In addition, a model may 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. They usually take some input parameters and – in the case of pre-fabricated simulations – return an image or video as result. These *program parameters* should correspond to the (input) parameters of the formal model. Therefore, the formal description's parameters can be referenced by the program parameters. Thus, the program parameter's role (name, description, and possibly domain/type) can be easily inferred. In addition to these correlated parameters, a simulation may also have parameters to control particular aspects of the simulations. Currently Matlab- and Sage-based simulations are supported. The Sage framework [16] provides a web-based Python interpreter and a very rich library of mathematical algorithms and data structures.

While the previously described way to use simulations by referencing an external program resource, for instance a Matlab or Sage file, is already supported by the *WoM*, we are currently developing *model-embedded simulations*, which are (Sage-based) programs that allow to define a simulation directly inside the XModel. Therefore, the XModel Schema allows code snippets that are parts of the formal description's parameters and equations/formulas. An accordingly extended schema is depicted in Figure 2. It shows the extension of the parameters and equations with the new element *simulation code* that encapsulates the additional code. Using IDs, each simulation code element is associated with the simulation to which it belongs. Because the order of the snippets in the description may be different to their order in the program, additionally, each simulation code element has to be numbered. For code, which does not

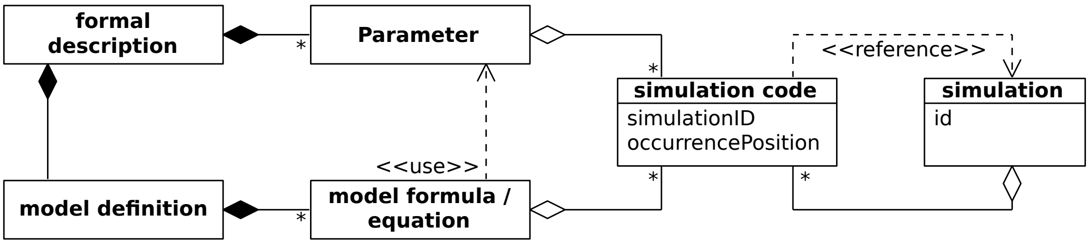


Figure 2: XModel Schema with integrated simulations

directly belong to a formal description's component, such snippets can be defined within the body of the corresponding simulation element.

Metadata. The XModel Schema also includes metadata such as the model's *authors*, *contributors*, *holders of rights*, and *editors* as well as *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 [1] and the WZ08 - Classification of Economic Activities [4]. 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*.

Processing an XModel Document. As mentioned in the beginning of this section, we currently provide a LaTeX document class, which roughly describes the syntax of XModel. The LaTeX source is translated into an XModel document. Because of that, LaTeXML [12] in conjunction with BibTeXML [2] are used to translate the LaTeX document into an intermediate XML document, which is finally transformed into an XModel document using XSLT. Based on a valid XModel representation, further processing operations like a transformation into an XHTML representation or back into a LaTeX document are possible. The entire processing chain is depicted in Figure 3.

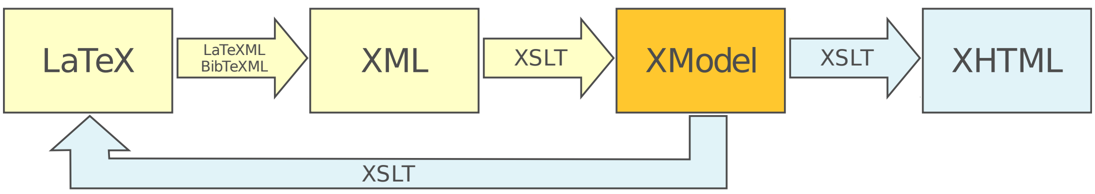


Figure 3: The processing chain from LaTeX to XHTML

In Figure 4, the result of such an additional transformation is shown. It exemplarily illustrates the XHTML representation of a model for the "Horizontal Shot". Furthermore, it shows the integration of a pre-fabricated Matlab simulation in section *4. Simulations*. A separate link to a corresponding Sage simulation is shown at the end of that section. Figure 5 gives an impression of the Sage interface.

Horizontal Shot

[stable]

Visualization Simulation

Created by D. Poetzsch-Heffter [WWW, @] ITWM
Keywords: shot; mechanics; physics; school

1. Informal Description

The horizontal shot describes the ballistic trajectory of a masspoint (i.e. it moves without friction) on the planet earth. This model is one of the basic models of the mechanics theory founded by Isaac Newton.

2. Formal Description

The basic situation of the model is that the masspoint is s_{y_0} metres above the ground and is shot with an initial speed named v_x parallel to the ground.

$$v_y(t) = gt$$
$$s_y(t) = \frac{1}{2}gt^2 + s_{y_0}$$
$$v_{tot}(t) = \sqrt{v_x^2 + v_y(t)^2}$$
$$\alpha(t) = \arctan\left(\frac{|v_y(t)|}{|v_x|}\right)$$

As you can see, v_y, s_y, v_{tot} and α depend on the input parameter t. By using the following equation it is also possible to determine these values using the way parallel to the ground s_x as input parameter:

$$t = \frac{s_x}{v_x}$$

2.1. Input Parameters

- s_{y_0} The initial distance of the masspoint to the ground; $s_{y_0} \in \mathbb{R}_0^+$
- v_x The x-direction (parallel to the ground) part of the speed; $v_x \in \mathbb{R}^+$
- t The elapsed time; $t \in \mathbb{R}_0^+$
- s_x The way in x-direction (parallel to the ground); $s_x \in \mathbb{R}_0^+$

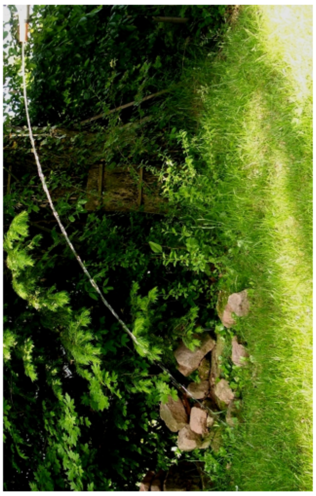
2.2. Output Parameters

- v_y The speed towards the ground; $v_y \in \mathbb{R}^+$
- s_y The distance to the ground; $s_x \in \mathbb{R}$
- v_{tot} The absolute value of the masspoint's speed; $v_{tot} \in \mathbb{R}^+$
- α The angle between the masspoint's direction of movement and the ground; $\alpha \in [0-90)$

2.3. Miscellaneous Parameters

- g The gravitational acceleration; $g = -9.81\frac{m}{s^2}$

3. Visualizations



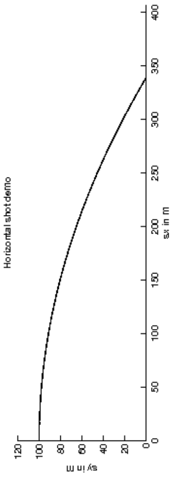
A ballistic trajectory shown via a water jet

4. Simulations

The movie of the following simulation is a realtime movie. This means the duration of the movie is exactly the time the masspoint would need to reach the ground in reality.

The initial distance of the masspoint to the ground s_{y_0} $[m]$ 100
The x-direction (parallel to the ground) part of the speed v_x $[m/s]$ 75

Picture Movie



I have prepared an interactive environment for plotting diagrams of horizontal shots (shown under this passage). Use function plot_trajectory(sy0, vx) to plot the curves.
Open SageMath simulation window

5. Classification

(AMS)70B05 — Kinematics of a particle

6. References

1. *Isaac Newton:* **Philosophiæ Naturalis Principia Mathematica** (1687)

7. Related models

Vertical shot General shot

Figure 4: Web representation of the “Horizontal Shot” XModel

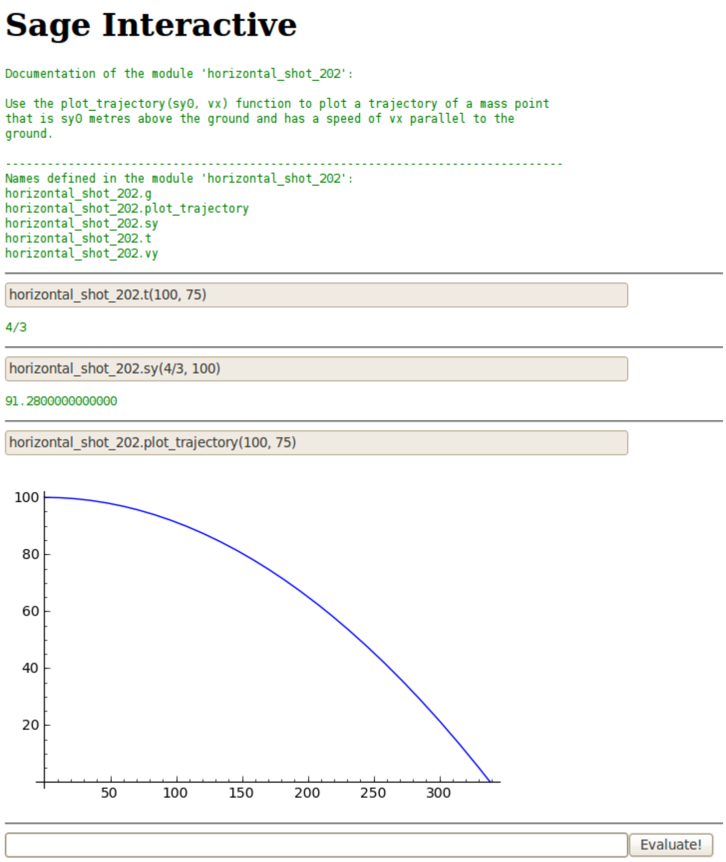


Figure 5: The Sage simulation provided by the Sage module which is part of the model "Horizontal Shot".

3 Related Work

Several communities work on different aspects to support math in the Web. Some of them focus on providing visual aids for mathematical algorithms, like the proprietary platform available under [17], which is based on Mathematica/WolframAlpha. *Demonstrations*, which are mainly interactive, are based on a CDF[4] file that can only be processed by proprietary tools. For various demonstrations unstructured text comments and author information are given. Demonstrations may be referenced to "related" demonstrations, but that is not necessarily a symmetric relation. It also misses any community features, model structuring, and alternative visualization capabilities.

Another approach are math-related Wikis, which are based on several freely available software projects. A wide range of them use the OMdoc [9] format[5]. It concentrates on the representation of mathematical equations, their transformation, and referencing. Others use the OpenMath and MathML standards [3]. Sample projects are [10] (outdated) and [15] – "The Encyclopedia Sponsored by Statistics and Probability Societies". While they are providing AMS classifications, user communities, and LaTeX-based integration of new content, they lack a clear model structure. For instance plain LaTeX documents and visual support are limited to ordinary images. Another prototype *JOBAD* (JavaScript API for OMDoc-based Active Documents) [8] concentrates on the integration of other (ad-hoc called) Web

[4]Computable Document Format
[5]OMdoc is an inter lingua for mathematical communications.

Services. They are linked via keywords or explicit user input. This project's main goals are on-the-fly conversion of units, and document visualization style. Neither interactive models nor structured models are provided.

The PlanetMath project [14] collects – similar to a math encyclopedia – all kinds of math-related and LaTeX-sourced documents and makes them available as HTML pages. The downloadable intermediate XML format does not provide any clear structuring or support for model linkage, besides keywords, which are automatically extracted from the text. While community features like comments and history are supported, neither visual assistance, except for images, nor interactive playgrounds are provided.

Platforms like ActiveMath [11] are capable of communicating with computer algebra systems or formalizing mathematical expressions in order to annotate or simply present them in the Web. Some of them provide learning platforms that allow flash programs and applets to be embedded. Special platforms (see e.g. [13]) are specialized in proof languages and proof checker capabilities for all math-related expressions. But none of them provide any community features nor do they support the notion of mathematical models and their relations.

While the *WoM* and the OKSIMO project (formerly known as Planet Earth Simulator [5]) have in common that they want to model day-to-day problems and make the partially interactive models available on the Web, the OKSIMO project depends on a propriety Fcl input language, i.e., a visual programming language, to submit new models. In contrast to *WoM*, OKSIMO lacks a structured model repository.

In summary, the described approaches have different focuses, 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.

4 Conclusion

We introduced the Web of Models, a platform for storing, searching, exploring, and sharing mathematical models. In this paper, we focused on the structured representation of a model as an instance of the XModel Schema. It defines the scope of *WoM* and, besides others, allows to classify models as well as relate them to each other. It standardizes the model's structure and thus improves model consistency. It allows to integrate simulations whether as references to an already existing program or directly inside the model's description via model embedded simulations. With the help of such simulations the user can explore and experiment with the models. Created from a LaTeX document an XModel instance can be transformed into different representations like XHTML. Finally, XModel is the cornerstone of *WoM*, of the model platform built around, and for all available transformation and processing options. Moreover, it supports any future evolution steps by automated processing capabilities.

References

[1] American Mathematical Society. `http://www.ams.org/`.

[2] BibTeXML. `http://www.arakhne.org/bib2ml/`.

[3] O. Caprotti and D. Carlisle. OpenMath and MathML: semantic markup for mathematics. *Crossroads*, 6:11–14, November 1999.

[4] Statistisches Bundesamt Deutschland. `http://www.destatis.de/`.

[5] Gerd Doeben-Henisch. The planet earth simulator project - a case study in computational semiotics. In *AFRICON, 2004. 7th AFRICON Conference in Africa*, volume 1, pages 417–422, September 2004. `http://oksimo.inm.de/`.

[6] Pieter Eykhoff. *System Identification: Parameter and State Estimation*. Wiley, 1974.

[7] Jean-Marie Gaillourdet, Thomas Grundmann, Martin Memmel, Karsten Schmidt, Arnd Poetzsch-Heffter, and Stefan Deßloch. WoM: An Open Interactive Platform for Describing, Exploring, and Sharing Mathematical Models. In A. König et al., editor, *Knowledge-Based and Intelligent Information and Engineering Systems, 15th International Conference, KES2011*, Lecture Notes in Computer Science, 2011. to appear.

[8] Jana Giceva, Christoph Lange, and Florian Rabe. Integrating web services into active mathematical documents. In Jacques Carette, Lucas Dixon, Claudio Coen, and Stephen Watt, editors, *Intelligent Computer Mathematics*, volume 5625 of *LNCS*, pages 279–293. Springer Berlin / Heidelberg, 2009.

[9] Michael Kohlhase. OMDoc: Open mathematical documents. In Fred Vries, Graham Attwell, Raymond Elferink, and Alexandra Tödt, editors, *Open Source for Education*, pages 137–143. Open Universiteit Nederland, November 2005.

[10] MathWeb. `http://www.mathweb.org`.

[11] Erica Melis, Giorgi Goguadze, Paul Libbrecht, and Carsten Ullrich. ActiveMath – a learning platform with semantic web features, 2009. `http://www.activemath.org`.

[12] Bruce Miller. LaTeXML. `http://dlmf.nist.gov/LaTeXML`.

[13] A. Naumowicz and A. Kornilowicz. A brief overview of Mizar. In S. Berghofer and et al., editors, *TPHOLs 2009*, 2009. LNCS 5674.

[14] PlanetMath. `http://planetmath.org`.

[15] StatProb. The encyclopedia sponsored by statistics and probability societies. `http://statprob.com`.

[16] W. A. Stein et al. *Sage Mathematics Software*. The Sage Development Team, 2011. `http://www.sagemath.org`.

[17] Wolfram. Wolfram demonstrations project. `http://demonstrations.wolfram.com`.