The Flexiformalist Manifesto

Michael Kohlhase
Computer Science
Jacobs University Bremen, Germany
m.kohlhase@jacobs-university.de

***Abstract*—In this manifesto, we develop the "Flexiformalist Program". This may be thought of as a modern extension of Hilbert's "Formalist Program", which solved the foundational crisis of mathematics but remained purely theoretical: Even though formal representations are a prerequisite for computer support in mathematics, formalization is a possibility that is usually unconsummated and thus does not have any practical influence on day-to-day mathematics.**

The Flexiformalist Program aims to change this and calls for

1. The development of a regime of partially formalizing
   a. mathematical knowledge into a modular ontology of mathematical theories (*content commons*), and
   b. mathematical documents by semantic annotations and links into the content commons (*semantic documents*),

2. The establishment of a software infrastructure with
   a. a *distributed network of archives* that manage the content commons and collections of semantic documents,
   b. *semantic web services* that perform tasks to support current and future mathematic practices
   c. *active document players* that present semantic documents to readers and give access to respective

3. the re-development of a comprehensive part of mathematical knowledge and the mathematical documents that carries it into a *flexiformal digital library of mathematics* (FDLM).

We believe that such a flexiformal digital library will significantly empower mathematicians, scientists, and engineers in research, education, and application. Our experiences with early approximations of the FDLM show that the approach can be profitably transferred to other domains in science, technology, and engineering.

***Index Terms*—???????,??????**

I. Introduction

Mathematics plays a fundamental role in science, technology, and engineering (STEM). Mathematical knowledge is rich in content, sophisticated in structure, and technical in presentation. Its conservation, dissemination, and utilization constitutes a challenge for the community and an attractive line of inquiry. In this manifesto, we will take an MKM (Mathematical Knowledge Management) view. MKM is an interdisciplinary field at the intersection of mathematics, computer science, Semantic Web, library science, and scientific publishing that develops representation formats, methods, and tools to facilitate the creation of a "universal digital library of mathematics" and empower its users with added value services (see [Far05] for an introduction). It is a driving intuition for the MKM community that mathematical knowledge constitutes an attractive "test tube" for structure research and tool development and that results and tools can be generalized to all of STEM.

Most of mathematical knowledge is currently recorded in the form of informal (see below) documents – ranging from journal papers over preprints to sketches on blackboards. We can distinguish three levels of representation in electronic documents:

1) **digitized** (usually from printed material): such documents are essentially electronic images of their printed precursors and can be distributed electronically to be read by humans
2) **presentational:** i.e. electronically encoded text interspersed with presentation markup – meta-level commands that describe the visual (or auditory) appearance of layout of the document and its components (e.g. formulae and tables).
3) **semantic:** i.e. text interspersed with content markup that makes the meaning conveyed by the document explicit and thus machine-actionable: document components are classified by their function, and relations between them and other objects are marked up with standardized or machine-actionable vocabularies.

It is a crucial observation that documents at the higher levels can easily be transformed into documents at the lower level, while transformations up the hierarchy currently require human intervention and can only be automated heuristically, e.g. by optical character recognition (OCR) from 1. to 2. and computational linguistics from 2. to 3. Also note that computer support for access, aggregation, re-use, application, and verification is (largely) restricted to the semantic level, except for relatively shallow information retrieval methods.

Efforts are currently under way to digitize and possibly OCR large parts of the published mathematical literature and turn them into generally accessible "Digital Mathematical Libraries". We claim that *the digitization/OCR effort should be complemented by a flexiformalization effort that makes the DMLs semantically accessible and turns mathematical documents into active documents*. This manifesto makes a concrete proposal on how to deal with STEM documents at the semantic level.

II. Formality?, Informality?, Flexiformality!

As all machine support is based on syntactic manipulations (until we achieve artificial intelligence) we need some formalization if we want to enlist computers in mathematics. Machine support in mathematics and STEM is advantageous, since humans and machines have complementary strengths and weaknesses. Humans have unmatched abilities in exploring mathematical theories while developing deep insights into the key properties and inherent invariants, which allow them to conjecture key statements and drive proofs via accurate intuitions. Machines excel at systematic analysis of large structures – e.g. for verifying large and convoluted proofs or indexing large datasets for search. We claim that mathematics research and application will be strongest, if we employ a combination of human and machine strengths. In particular, machine support will allow us to break the *one-brain barrier of STEM research and application* – to progress and make connections, all relevant knowledge must be co-located in one brain.

To start off our discussion, we recap the notions of formal informal representations in mathematics.

A. Formality

Since the foundation debate in mathematics almost a century ago, "formal mathematics" has been defined as comprising reifications of mathematical knowledge expressed in a "formal system", i.e. in a well-defined logical language with a syntactic proof system, where grammaticality of expressions and the verification of proofs is decidable. Moreover, formal systems are usually expected to have a well-defined model theory, into which expressions interpreted compositionally.

Formal developments of mathematics fix a foundation: a logical system $\mathcal{L}$ and a foundational theory $\mathcal{F}$, e.g. first-order logic with descriptions and ZFC set theory. Based on this foundation, mathematical objects are specified via axioms and/or definitions (special $\mathcal{L}$-expressions), and their properties stated in form of "assertions" ($\mathcal{L}$-expressions again) which are justified by proofs (again $\mathcal{L}$ expressions; we assume $\mathcal{L}$ to contain a proof system).

These concepts were developed in Hilbert's "Formalist Program" which established that all of contemporary mathematics could in principle be formalized in first-order logic and axiomatic set theory (the commonly accepted foundation of mathematics). Contrary to the aim of Hilbert's program, it was also established that foundations cannot be complete (all valid statements can be proven in $\mathcal{L}$) and no foundation can be proven consistent – inconsistency would render all statements true and the foundation useless. These foundational constraints notwithstanding various foundations of mathematics[1] have remained empirically without contradictions for almost a century.

Note that we can also view programming languages as foundations for mathematical objects, they also provide base vocabularies for expressing mathematical objects, but favor operational aspects of meaning over set-theoretic models and apply these to the automation of simplification and computation rather than that of inference and entailment. The last decades have seen the development of foundations that integrate computation and inference in a single foundation; the calculus of inductive constructions [BC04] is a particularly influential example that integrates computation into constructive logic and the SPAD/Aldor [JS92] one that integrates axiomatics into computation.

B. Informality

In the sense above, almost all mathematical documents are informal in at least four ways:

I1. *The foundation is unspecified*: mathematical documents usually leave the foundation open.

I2. *The language is informal*: mathematical vernacular is a mixture of natural language with formulae and discourse-level cues on the epistemic status of text fragments (definition, theorem, proof, etc.). This is informal, as we do not have decision procedures for grammaticality or interpretation.

I3. Even *formulae are informal*: as they are in presentation markup that specifies the layout, and not the logic/functional structure of the mathematical object or property represented. For instance $f(a + b)$ could denote the application of a function f to the sum of a and b or the product of a scalar f with $a + b$.

I4. *Context references are underspecified*: mathematical objects and concepts are often identified by name without making the references to context explicit. This applies both to the natural language part, the formulae, and at the statement level (citations of definitions, theorems, and proofs).

In a world, where mathematical documents are exclusively addressed at human readers, all of these informalities are features, not bugs, since they avoid spurious over-specifications (e.g. most foundations are essentially equivalent, and most arguments can be formalized into most foundations). The mathematical community has developed the standard of "*rigorous developments*" for the subset of documents that could be formalized in some foundation given enough resources.

But this underlying assumption of formalizability is almost never consummated in practice, even though (as we have argued above) this would be the prerequisite of machine support. We claim that there are two main reasons for this:

P1. The services of symbolic software systems have largely been restricted to

- *symbolic computation*, which restricts itself to the manipulation of (representations of) mathematical objects and largely ignores the levels of statements and theories, and
- *proof verification*, which fixes a solved problem – peer review works for all relevant mathematical results. Verification of safety-critical systems, whose properties can be expressed as mathematical statements and whose proofs are non-intuitive and massive in size are a different matter. There, formalization and machine-support in proof-construction and -checking is already industrial practice.

In both cases formalization poses a problem as it becomes a difficult problem to decide whether the formalization corresponds to the initial, informally given problem.

[1]Note that foundations need not be first-order set theories, but can be type theories (often constructive) or based on category theory.

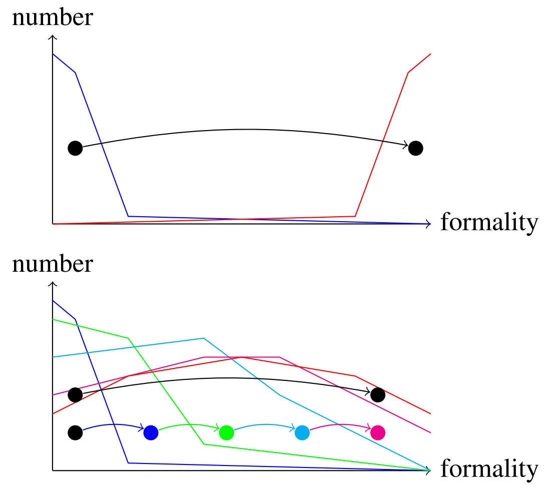

Fig. 1. Full Formalization vs. Flexiformalization

P2. In particular, the significance of proofs in mathematical documents is two-fold: they not only act as a *certificate for validity*, but first and foremost as form of communication about the key properties and inherent invariants of the mathematical objects in question (see [Asp12] for a discussion). In fact the proofs are the main triggers of determining these properties and invariants.

P3. The situation is compounded by the fact that formality is an all-or-nothing property, so formalization is a big investment since it has to overcome at least the four sources of informality shown above.

To solve **P3**, we need to develop a notion of graded formality, which allows multi-dimension, stepwise formalization, so that instead of having to fully formalize documents in one long jump documents can be incrementally made more formal to the desired level in a step-by-step process possibly involving multiple players (see Figure 1). But note that **P***i* are interrelated; we will discuss them in the next two sections starting with the last, since it is the most fundamental.

III. Flexiformality

In [KK11] we have introduced the concept of *flexiforms* for representations of mathematical knowledge of flexible formality, and the concept of *flexiformalization* for any act of disambiguation by explicit markup.

We take the 'meaning' of a document to be the set of all of its possible formal representations. But even the space of fully formal reified mathematical knowledge is large and difficult to grasp — it contains all well-formed expressions in all logics, so we conceptualize it as a two-dimensional space $\mathfrak{F}$: Let $\mathfrak{L}$ be the set of all logical systems, then the space $\mathfrak{F}$ of formalizations can be constructed as $\mathfrak{F} := \{\langle \mathcal{L}, e\rangle \mid e \in \mathcal{L} \in \mathfrak{L}\}$, and any formal representation as a point in $\mathfrak{F}$.

We consider documents as underspecified representations of formalizations, so for any document D, there is a set $\mathcal{I}(D) \subseteq \mathfrak{F}$ it could be formalized as. Note that $\mathcal{I}(D)$ is non-empty, since we postulate documents to be formalizable (in principle) and indeed $\mathcal{I}(D)$ is usually quite large, since even

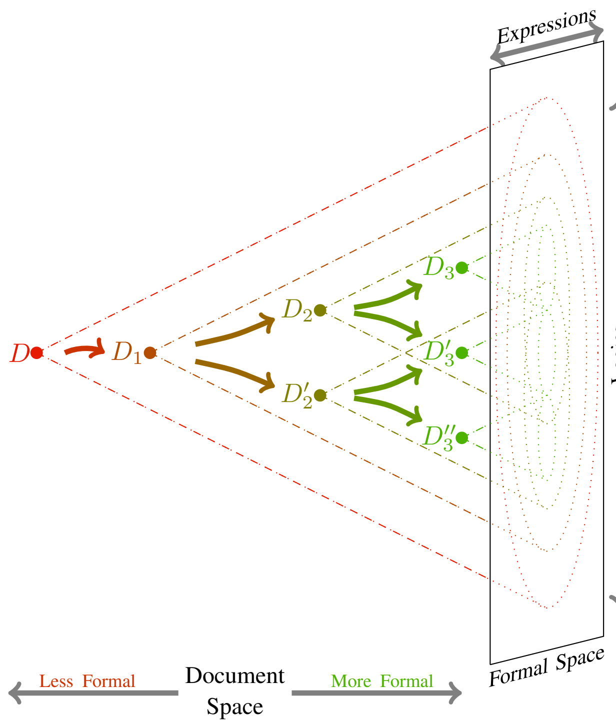

Fig. 2. Stepwise Flexiformalization of Documents

rigid mathematical documents omit many aspects and details of the formalizations. In particular, mathematical objects can be formalized in different logics, and in a given logic as different expressions — these include different concretizations of the concept as well as logically equivalent formulations of a concretization.

In Figure 2 we have depicted the space $\mathfrak{F}$ as a plane on the right hand side, and alternative sequences of documents with their interpretations depicted as cones based in $\mathfrak{F}$. We understand this sequence as a *stepwise flexiformalization* process, beginning with a document D. In our example, each successive formalization step will fix certain formal aspects, restricting the set of possible formalizations further and further. Following this intuition we can define that a document D is **more formal than** D' (write $D' \lll D$), iff $\mathcal{I}(D) \subseteq \mathcal{I}(D')$. This relation on documents and objects is a partial ordering relation (because the subset relation is) and provides an answer to the question of graded formality raised above. Fragments of a document D correspond to sub-formalizations of $\mathcal{I}(D)$, so we can extend the 'more formal than'-relation to document fragments and the objects of formalization.

IV. Semantic Services

To judge possible machine support for STEM research and application we need to realize that many services do not require full formalization (as computation and verification do in **P1**). For instance, screen readers for mathematical documents only require presentational markup of formulae, definition lookup (see below) or formula search only content

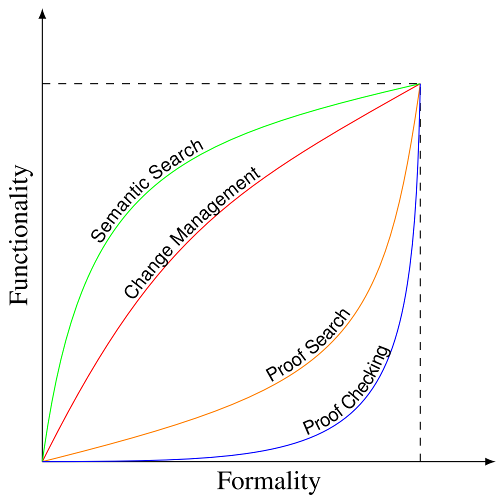

Fig. 3. Functionality of Flexiformal Services

markup[2], and management of change an explicitly represented dependency relation. The full situation is probably best understood in terms of the diagram in Figure 3, where we graph the functionality of services against the level of formality of the documents/knowledge representation they act on. Semantic search can give a high return even on relatively informal documents. For instance the DBPedia system [Aue+07] gets by with screen-scraping Wikipedia fact boxes – here the systematic arrangement in the fact boxes serves as a kind of formalization from which RDF triples can be harvested, which can be used for querying via description logic inference systems. At the far other end of the scale we find systems like Wolfram Alpha [Wol], which is based on a full formalization of various knowledge sources in the Mathematica language[3] and can give answers that combine complex computations with chaining inferences. For change management, the situation is similar, but less pronounced, since any non-trivial method needs an explicitly represented dependency relation, so methods range from industrial requirements tracing [Jar98] to automated change management techniques in formal methods, e.g. [Aut+00; IR12]. On the other side of the (imaginary) diagonal of Figure 3 we have proof search and proof checking, where – somewhat counter-intuitively – the latter has a more convex gradient since proof search only requires the formalization of the conjecture, whereas in proof checking the proof has to be fully formalized as well.

V. Active Documents, Semantic Libraries

To enable an optimal collaboration between man and machine, we need at the same time to keep close to established workflows of mathematicians and give algorithms the explicit representations needed for computation. For the first goal we want to keep traditional documents as "user interfaces" and augment them with embedded services that activate the content for interaction and adaption (we call such enhanced documents **active documents**). Whereas the documents themselves are essentially tree-structured, the knowledge reified in them is best structured as a hypergraph, where the nodes are mathematical objects, statements, and theories, whereas the edges are given by the content-structural relations among them – e.g. the "inheritance" and "views" relations between theories, the "justification" of theorems by proofs, and the "dependency" of new concepts on the concepts in the definienda. We call such a hyper-graph structure the **Content Commons**, since we envisage it as a shared, communal resource.

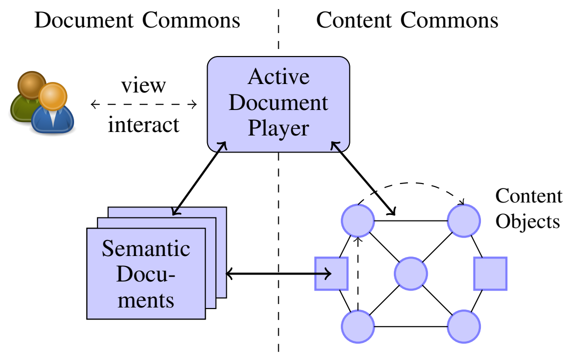

Fig. 4. Active Documents Paradigm

In the **Active Documents Paradigm** (ADP see Figure 4 for an overview and e.g. [Koh+11] for details) both structures are read by a **document player**: a software application that generates document presentations that are instrumented with controls for user interactions (the active documents). It is crucial for the ADP that the documents are semantically annotated (we call them **semantic documents**), usually by classifications of text fragments and references into the content commons that serves as an explicit semantic context. For instance, symbols in formulae or technical terms in a text could be linked to rigorous definitions; the link is an example of a formalization of the informal context references discussed in clause **I4** above. Having such formal links directly translates into an active-document service: "*definition lookup*", which displays the definition induced by a click (or hover) on the symbol or technical term and invites the reader to explore the context from the definition.

We call a collection of semantic documents together with a content commons a **semantic library** and remark that active documents only make sense as members of semantic libraries. Semantic libraries can arise in multiple ways and depths (of formalization). One example is the OSCAR3 system [CMR06], a tool for shallow, chemistry-specific parsing of scientific documents which attempts to identify chemical names, ontology terms, and chemical data and augments documents with links and elaborations. The Science-WISE [ScW13] project system is similar, but concentrates on semantic navigation via ontology relations and crowd-sources the semantic annotations. A math-oriented approach

[2]in the sense of content MathML [Aus+10, Chapter 4] and OpenMath [Bus+03], i.e., semantic annotations of the functional structure of formulae and disambiguation of symbols by reference to their definition

[3]A task worth hundreds of person years carried out over the better part of a decade.

is to analyze digital mathematical documents and recover semantic structures, e.g. by transforming LaTeX documents into HTML5, transforming the relations given in the functional LaTeX markup into RDFa annotations which can then be harvested into a content commons realized as a triple store (RDF database); see [Sta+10] for details and [Gin+09] for linguistics-based analysis methods. On the other side of the flexiformality spectrum are presentation workflows that start from completely formal representations and generate semantic documents from that: For instance, the Mizar Mathematical Library [MizLib] – which contains over 1000 "articles" with over 50000 theorems and over 10000 definitions – is published in the *Journal of formalized Mathematics* (JFM [JFM]). JFM articles are generated from formal "Mizar articles" in an automated presentation process;the JFM still misses out on the chance to make them active, but the Mizar Wiki [Urb+10] does not. Our group has developed the Planetary system [Koh+11; PDF], a generic active document player that can be instantiated to all levels of flexiformality.

VI. An Active, Semantic Layer for STEM Libraries

Mathematical documents are at the same time *i)* precision tools optimized for the efficient communication of mathematical knowledge among specialists who share a common knowledge context and *ii)* formidable obstacles to be overcome to build up this shared context which is a prerequisite for understanding. The understanding problem is aggravated by the fact that mathematical knowledge has been growing ever more diverse and intricate over time. The current digitization efforts (e.g. EuDML [Eud] and Google Books) go a first step towards wider adoption of mathematical knowledge by providing *universal access to the mathematical literature*. With the emerging technologies of flexiformal, semantic libraries and active documents, we have a way to make the mathematical literature more *accessible to non-specialists*[4], by giving access to crucial aspects of the context at the "points of pain" (i.e. in the documents) at the cost of partial flexiformalization of technical documents and the establishment of a content commons.

This already reveals the main non-technical problem involved in semantic mathematical libraries: unless there is an initial investment into a core content commons to link into, the cost of semantic annotation of documents outweighs the benefit from active documents. We claim that by an act of technology adoption by a major player (a professional society, charitable foundation, funding agency, or major publisher), we can achieve method standardization and a critical mass of content that kickstarts active mathematical documents and semantic libraries. There is precedent in this: a bold (at the time) move of the AMS of requiring TeX/LaTeX in its journals brought about the improvement in mathematical/scientific typesetting we still profit from by setting a standard and creating critical mass of know-how and tools. We conjecture that the induced network effect will lead to widespread flexiformalization[5], and that we will see additional synergy effects, such as the following one: As soon as a larger body of mathematical theories becomes available (by marking up concepts and axioms) we can automatically search for "views" (aka. representation theorems) that allow to import all the theorems of the source theory of the view into the target theory (after translation with the view's signature morphism). We conjecture that systematic automated search will reveal many long-distance views that could not have been found otherwise, as the chance that humans know source and target theories well enough to notice the structural similarities (a one-brain constraint) is slim in today's highly specialized sciences. Methods for this search exist [NK07], we only lack the semantic libraries.

In conclusion we can state that the relaxation of the black-and-white distinction of formality and informality to the flexiformality continuum opens up considerable opportunities for machine support in STEM research and applications. The flexiformalist manifesto suggests three concrete steps to reap these benefits: *i*) foundational research in the concept of flexiformality, *ii*) joint representation formats and tool stacks, and *iii*) a large case study to evaluate applicability and practical utility of flexiformalization. We have already seen first steps in all three directions, and propose flexiformalization as an exciting and integrating topic of meta-scientific research.

Acknowledgements The flexiformal manifesto presented here crystallizes ideas that have been floating around in the MKM and CALCULEMUS communities for almost two decades, the author gratefully acknowledges that his ideas have been influenced by discussions, publications, and system demos by Andrea Asperti, Bruno Buchberger, Jacques Carette, Claudio Sacerdoti Coen, Joe Corneli, James Davenport, William Farmer, Deyan Ginev, Mihnea Iancu, Fairouz Kamareddine, Manfred Kerber, Andrea Kohlhase, Christoph Lange, Bruce Miller, Till Mossakowski, Florian Rabe, Jörg Siekmann, Andrzej Trybulec, Josef Urban, and Stephen Watt. Work on the concepts presented here has been partially supported by the Leibniz association under grant SAW-2012-FIZ_KA-2 and the German Research Foundation (DFG) under grant KO 2428/10-1.

References

[Asp12] Andrea Asperti. "Proof, Message and Certificate". In: *Intelligent Computer Mathematics*. Ed. by Johan Jeuring et al. LNAI 7362. Berlin and Heidelberg: Springer Verlag, 2012, pp. 17–31.

[Aue+07] Sören Auer et al. "DBpedia: A Nucleus for a Web of Open Data". In: *ISWC/ASWC*. Ed. by Karl Aberer et al. Lecture Notes in Computer Science 4825. Springer Verlag, 2007.

[4]and we are all non-specialists for most of mathematics

[5]For instance the linked open data effort was jump-started by the emergence of DBPedia [Aue+07], which covered all conceivable subject of interest (albeit in a shallow way). This established a common data set which other datasets could relate to and thus interoperate.

[Aus+10] Ron Ausbrooks et al. *Mathematical Markup Language (MathML) Version 3.0*. W3C Recommendation. World Wide Web Consortium (W3C), 2010. URL: http://www.w3.org/TR/MathML3.

[Aut+00] Serge Autexier et al. "Towards an Evolutionary Formal Software-Development Using CASL". In: *Proceedings Workshop on Algebraic Development Techniques, WADT-99*. Ed. by C. Choppy and D. Bert. LNCS 1827. Springer Verlag, 2000, pp. 73–88.

[BC04] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development — Coq'Art: The Calculus of Inductive Constructions*. Texts in Theoretical Computer Science. An EATCS Series. Springer Verlag, 2004.

[Bus+03] Stephen Buswell et al. *The OpenMath Standard, Version 2.0 Public Draft 4*. Tech. rep. The OpenMath Society, 2003. URL: http://www.openmath.org/standard/om20-2003-11-24/.

[CMR06] Peter Corbett and Peter Murray-Rust. "High-Throughput Identification of Chemistry in Life Science Texts". In: *Computational Life Sciences II, Second International Symposium, CompLife 2006*. Ed. by Michael R. Berthold, Robert C. Glen, and Ingrid Fischer. Vol. 4216. LNCS. Springer Verlag, 2006, pp. 107–118.

[Eud] *EuDML – The European Digital Mathematics Library*. URL: http://eudml.eu (visited on 08/02/2011).

[Far05] William M. Farmer. "Mathematical Knowledge Management". In: *Encyclopedia of Knowledge Management*. Ed. by David G. Schwartz. Idea Group Reference, 2005, pp. 599–604.

[Gin+09] Deyan Ginev et al. "An Architecture for Linguistic and Semantic Analysis on the arXMLiv Corpus". In: *Applications of Semantic Technologies (AST) Workshop at Informatik 2009*. 2009. URL: http://www.kwarc.info/projects/lamapun/pubs/AST09_LaMaPUn+appendix.pdf.

[IR12] Mihnea Iancu and Florian Rabe. "Management of Change in Declarative Languages". In: *Intelligent Computer Mathematics*. Ed. by Johan Jeuring et al. LNAI 7362. Berlin and Heidelberg: Springer Verlag, 2012, pp. 325–340.

[Jar98] M. Jarke. "Requirements Tracing". In: *Communication of the ACM* 41.12 (1998).

[Jeu+12] Johan Jeuring et al., eds. *Intelligent Computer Mathematics*. LNAI 7362. Berlin and Heidelberg: Springer Verlag, 2012.

[JFM] *Journal of Formalized Mathematics*. URL: http://www.mizar.org/JFM (visited on 09/27/2012).

[JS92] Richard D. Jenks and Robert S. Sutor. *AXIOM: The Scientific Computation System*. Springer-Verlag, 1992.

[KK11] Andrea Kohlhase and Michael Kohlhase. "Towards a Flexible Notion of Document Context". In: *Proceedings of the 29th annual ACM international conference on Design of communication (SIGDOC)*. 2011, pp. 181–188. URL: http://kwarc.info/kohlhase/papers/sigdoc2011-flexiforms.pdf.

[Koh+11] Michael Kohlhase et al. "The Planetary System: Web 3.0 & Active Documents for STEM". In: *Procedia Computer Science* 4 (2011): *Special issue: Proceedings of the International Conference on Computational Science (ICCS)*. Ed. by Mitsuhisa Sato et al. Finalist at the Executable Paper Grand Challenge, pp. 598–607. DOI: 10.1016/j.procs.2011.04.063.

[MizLib] *Mizar Mathematical Library*. URL: http://www.mizar.org/library (visited on 09/27/2012).

[NK07] Immanuel Normann and Michael Kohlhase. "Extended Formula Normalization for ϵ-Retrieval and Sharing of Mathematical Knowledge". In: *Towards Mechanized Mathematical Assistants. MKM/Calculemus*. Ed. by Manuel Kauers et al. LNAI 4573. Springer Verlag, 2007, pp. 266–279.

[PDF] *Planetary Developer Forum*. URL: http://planetary.mathweb.org/ (visited on 09/15/2012).

[ScW13] *ScienceWise – Scientific Web-based Interactive Semantic Environemnt*. Jan. 16, 2013. URL: http://sciencewise.info (visited on 01/16/2013).

[Sta+10] Heinrich Stamerjohanns et al. "Transforming large collections of scientific publications to XML". In: *Mathematics in Computer Science* 3.3 (2010): *Special Issue on Authoring, Digitalization and Management of Mathematical Knowledge*. Ed. by Serge Autexier, Petr Sojka, and Masakazu Suzuki, pp. 299–307. URL: http://kwarc.info/kohlhase/papers/mcs10.pdf.

[Urb+10] Josef Urban et al. "A wiki for Mizar: Motivation, considerations, and initial prototype". In: *Intelligent Computer Mathematics*. Ed. by Serge Autexier et al. LNAI 6167. Springer Verlag, 2010, pp. 455–469. arXiv: 1005.4552v1 [cs.DL].

[Wol] *Wolfram—Alpha*. URL: http://www.wolframalpha.com (visited on 01/05/2013).