Towards Flexiformal Mathematics

by

Mihnea Iancu

PhD Research Proposal

Prof. Dr. Michael Kohlhase

Jacobs University, Germany

Prof. Dr. Herbert Jaeger

Jacobs University, Germany

Prof. Dr. William M. Farmer

McMaster University, Canada

Dr. Florian Rabe

Jacobs University, Germany

Date of submission: February 25, 2013

School of Engineering and Science

Executive summary

Even though computers and the internet have transformed many aspects of day-to-day life, their application to science in general and mathematics in particular, while substantial, has not yet reached its full potential. Scientific knowledge has the crucial feature of being based on rigorous principles and reasoning, much of which can be automated using computer-based methods. This is particularly true for mathematical knowledge, which is entirely built from precise definitions and logical inferences. Furthermore, because mathematics is the foundation and language of science, advanced computer support for it has the potential to revolutionize scientific development.

But computers can only "understand" mathematical documents that are formal (i.e. where the implicit definitions and reasoning rules are made explicit) and can only provide advanced support for documents that they understand. Currently, formalization requires completely rewriting existing knowledge in a logical system and, despite numerous efforts towards formalizing mathematics, only a small fraction of mathematics is formal. Furthermore, existing applications for such knowledge are mostly limited to proof checking and minimal proof inference so there is little practical incentive for formalization.

However, conceptually, the space of possible applications for formal content is actually quite large. Change management, definition lookup or applicable theorem search are only a few valuable services where automation can be immensely useful. Moreover, for many such services complete formalization is not necessary, as they can already be applied to partially formal documents.

Following this idea, I propose a novel, application-driven, approach to knowledge formalization. I use the notion of flexiformality which views documents as having flexible levels of formality ranging from completely formal to completely informal. The flexiformalization process can be viewed as a gradual process of progressive formalization by adding semantics to existing documents. Coupled with useful services that can make immediate use of any added semantics this provides a new improved way of creating, learning and discussing scientific knowledge and creates instant gratification for formalization work.

I feel this is crucial to catalyze the flexiformalization effort and is the key to achieving not only a significant amount of flexiformalized mathematical content but also practical applications for it. In the long term I believe this has the potential of automating much of mathematical and scientific development and, consequently, have a revolutionary effect on science similar to that which the industrial revolution had on manufacturing.

Therefore, I plan to create a system that can enable and stimulate incremental formalization of existing knowledge. Concretely, I will design a language for representing flexiformal knowledge and implement a system around it which provides valuable semantic services based on the semantics added through formalization.

Contents

1 Introduction **4**

2 State of the Art **6**

3 Preliminaries **8**

3.1 The MMT Language and System 8

3.2 Flexiformality 10

4 Motivation and Research Overview **11**

4.1 A Formal Language for Flexiformal Knowledge 12

4.2 A Semantic Kernel for Flexiformal Knowledge 15

5 Preliminary Work Plan and Schedule **16**

5.1 Work Plan 16

5.2 Timeline 21

6 Expected Results and Applications **21**

Abstract

The application of computer-based methods to mathematics, while meaningful, is constrained by the fact the most mathematical knowledge exists in forms that can only be understood by humans. In order for it to be also understood by machines it needs to be formalized, by making explicit the implicit definitions and inference rules that occur naturally in mathematical documents. But, despite numerous attempts, only a small fragment of mathematics is formalized as formalization work is very expensive. Furthermore, only a few applications exist that are build to take advantage of formal content.

Relying on the idea of flexiformality, I propose a flexible, application-driven approach to formalization. From the flexiformal perspective, documents have flexible levels of formality and, therefore, the flexiformalization process can be seen as a gradual process of progressive formalization by adding semantics to existing documents. I plan to design a formal language for representing flexiformal knowledge and implement a system around it which not only enables flexiformalization but also stimulates it by using the added semantics to provide valuable semantic services.

1 Introduction

Throughout history, knowledge has been a crucial building block for human civilization, and the creation, distribution and consumption of knowledge have been driving forces for human progress. This was never more clear than in our time when the advent of computers and the internet has revolutionized human society and has placed knowledge at its center. Nowadays, the world wide web provides a platform where a tremendous amount of knowledge is created, shared and consumed every day by billions of people.

However, despite the enormous amount of knowledge available on the internet in various forms, its practical utility is limited by the users' ability to find and understand knowledge relevant to them. This is because, currently, most of the worlds knowledge exists in formats designed for consumption by humans (text, images, pdf) and, as a result, their semantics is largely opaque to the computer and only limited computer support can be provided for it.

This limitation is highly relevant for scientific knowledge where meaningful computer support could be provided, since it is built on rigorous principles and reasoning. But it is particularly significant for mathematical knowledge, which is, by construction, based entirely on precise definitions and logical inferences. In order for computers to be able to make use of this, mathematical documents need to be *formal* by making explicit every definition and inference so that semantic reasoning can be represented as syntactic manipulation. *Informal* documents are documents which are not formal, and we distinguish here a particular kind of informality. *Rigorous* documents (the standard in day-to-day mathematics) are those where precise

definitions and formal inferences technically exist, but are implicit and left to the reader to infer.

Still, nowadays the vast majority of scientific knowledge in general and mathematical knowledge in particular exists in informal technical documents that can only be understood by experts. While for mathematics there are significant attempts at formalization [TB85, BC04, Pau94, Har96], only a relatively small part of mathematics is formal. Furthermore, most such attempts commit to a particular choice of foundation and, therefore, the resulting libraries of formalized mathematics are not interoperable. The QED project, whose goals were outlined in 1994 in the QED manifesto [Ano94], proposed creating a database of all mathematical knowledge, completely formalized and with all proofs formally checked but it never gained traction. In [Wie07] Freek Wiedijk identifies two major reasons for this failure, the small size of the formalized mathematics community and the lack of a "killer application" application for formalized content.

I propose an approach to formalizing mathematics that addresses these two issues. Firstly, I use a relaxed notion of formality based on the idea of flexiformality proposed in [KK11b]. This adopts an incremental view of the formalization process that does not require discarding all current informal mathematical content, but rather improving it by gradual formalization. Secondly, I use an application-driven approach where development is fueled by integration with semantic applications that can immediately make use of any additional semantics added during this incremental formalization process.

The result will be a *formal language for flexiformal knowledge* and a system implementing that language which provides semantic services and enables the gradual formalization workflow described above. Furthermore, the system will be integrated with existing mathematical libraries of various degrees of formality and with semantic applications.

Specifically, I will base my approach on the OMDoc [Koh06] document format and on the Mmt [RK13] language and system. OMDoc is an open markup format for mathematical documents which supports integration of formal and informal mathematical content. The Mmt language is designed as the minimal language that combines a module system with a foundationally uncommitted formal semantics. It is based on the formal core of OMDoc but omits OMDocs informal and narrative aspects. The OMDoc system is an open source Scala [OSV07] based implementation of the Mmt language, additionally providing some knowledge management services for (libraries of) Mmt documents.

Therefore, this proposal involves extending the Mmt language to be able to adequately represent structured knowledge in both its formal and informal incarnations, and extending the Mmt system so that it can leverage the flexiformal representation to provide meaningful semantic services. In particular, I will focus on mathematical documents and on formally capturing the natural modular structure implicitly used in informal mathematics.

This proposal is organized as follows. In section 2, I outline the state of the art and, in section 3, I briefly introduce MMT and flexiformality. Then, in section 4, I describe my central research goals and I present my concrete work plan and schedule in section 5. Finally, section 6 concludes the proposal.

2 State of the Art

Computer support for **conventional mathematics** has developed consistently over the years with many utilities reaching maturity and achieving widespread adoption.

A large part of mathematical knowledge is written, rendered and read using computer systems after being represented in *markup languages* such as LaTeX or, on the web, MathML.

Moreover, much this knowledge is now organized, archived and made available online in large *repositories of mathematical knowledge*. For instance, arXiv [Arx], euDML [BBNS11, EuD], Math Reviews [MRv] or ZentralBlatt Math [ZBM] contain hundreds of thousands of articles and provide services such as browsing, searching and subject based classification.

At the same time, *computer algebra systems*, such as Mathematica [Mat] , Maple [Map] or MuPAD [Fuc96] are routinely used for computation and verification of mathematical results. On the web, Wolfram Alpha [Wol] is a computational knowledge engine using Mathematica to perform deduction steps in search queries.

In the area of **formal mathematics** no system has achieved widespread adoption, although large formal libraries, as well as formalizations of significant individual theorems, have already been done in several such systems.

The Mizar project [TB85] attempts to reconstruct the mathematical vernacular in a computer-oriented environment. It is based on a variant of first order logic and uses Tarski-Grothendieck set theory. Its associated library, the Mizar Mathematical Library is one of the largest libraries of formalized mathematics with over 10000 definitions and over 50000 theorems including the fundamental theorem of algebra and the Jordan curve theorem.

The Coq system [BC04] is an interactive theorem prover based on the calculus of inductive constructions. Among the important formalization results formalized in Coq, are the the four color theorem, and the Feit-Thompson theorem.

The Isabelle prover [NPW02, Pau94] is generic, providing a meta-logic which can be used to declare object logics like FOL, HOL or ZFC. Isabelle/HOL has been used to formalize numerous mathematical results including Gödel's completeness theorem and the prime number theorem.

HOL Light [Har96] is a proof assistant in the HOL (Higher-Order Logic) family designed to have a small trustworthy kernel. Most notably, it is used in the Flyspeck [Hal06, HM$^+$] project which attempts to produce a formal proof of Kepler's

conjecture.

Applications for formal systems include semantic presentation [Urb06, ABR01] and IDE-style editors [Tea12, Wen12]. Similarly, MathWiki [CK07] is a (still in flux) wiki for formalized mathematics based on ProofWeb [Pro] and MediaWiki [MWk].

While no such formal system has achieved ubiquity in mainstream mathematics, there are many projects attempting **integrate formal methods into conventional mathematics**.

MathScheme [CFO10] is a long term project which aims to *combine formal deduction and symbolic computation* within one framework for mechanised mathematics. The MathScheme Library is grounded in a formal logic (Chiron [Far10]) and is based on the tiny theories approach.

PLATΩ[WAB06] is a *mediator between text-editors and and proof assistants*. It translates semantically annotated documents obtained from a WYSIWYG editor to the formal representation of an underlying proof system (ΩMEGA [SBA05]).

In order to make formalization more appealing to mathematicians some attempts exist create a compromise between the precision and rigor required by formal systems and the flexibility and expressivity desired by mathematicians by defining *semi-formal languages* that integrate formal and informal aspects.

WTT (Weak Type Theory) [Ned] is a refinement of de Bruijn's Mathematical Vernacular and is designed to act as an intermediary between common mathematical language and formal mathematics. Based on WTT, MathLang [KWZ08] aims at providing a compromise between expressivity and formality and thus provide an interface language between mathematicians and computers as well as a framework to make the links with existing formal proof systems.

The Naproche project [CFK+09] starts from the semi formal language of mathematics to develop a controlled natural language for mathematical texts and a proof checking software to formally check texts written in this language.

OMDoc [Koh06] is an open markup format and data model for mathematical documents designed to serve as a semantics-oriented representation format and ontology language for mathematical knowledge.

STEX [Koh08] is a semantic extension to LATEX that can produce not only a presentation oriented representation (pdf) but also a semantic content representation in XML/OMDoc which can be processed further and used on the web.

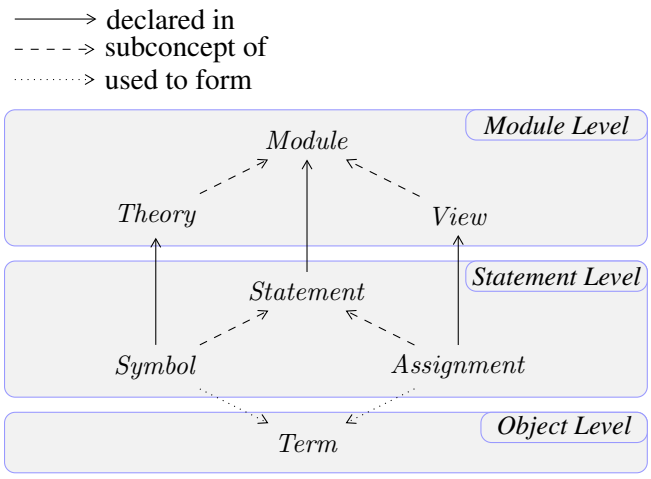

Figure 1: The MMT Ontology

3 Preliminaries

3.1 The MMT Language and System

MMT [RK13] is a generic, formal module system for mathematical knowledge and is a basis for foundation-independent knowledge representation.

3.1.1 MMT Language

The MMT language is designed to be applicable to a large collection of declarative formal base languages and all MMT notions are fully abstract in the choice of the base language. Therefore, MMT focuses on foundation-independence, scalability and modularity.

It is meant to be applicable to all base languages based on *theories*. Relations between theories are represented as MMT *theory morphisms* (or *views*). Theories and theory morphisms form the MMT *module* level. A *theory morphism* $\overline{\sigma} : S \to T$ is based on a *signature morphism* $\sigma : S \to T$ interpreting all symbols of S as terms in T but, additionally, $\overline{\sigma}$ translates all theorems of S to theorems of T. MMT uses the *Curry-Howard representation* to drop the distinction between symbols and axioms (and thus between signatures and theories). As a result, MMT needs only theories and theory morphisms.

I will only give a brief overview of MMT and refer to [RK13] for details (see also the MMT ontology in figure 1).

The central notion is that of a **theory graph**, a list of modules, which are theories T or theory morphisms v.

A **theory** declaration $T = \{Sym^*\}$ introduces a theory with name T containing a list of symbol declarations. A **symbol** declaration $c : \omega = \omega'$ introduces a symbol named c with **type** ω and **definiens** ω'.

Terms ω over a theory T are formed from symbols $T?c$ declared in T, bound variables x, applications $\omega\,\omega_1\,\ldots\,\omega_n$ of a function ω to a sequence of arguments, bindings $\omega X.\omega'$ using a binder ω, a bound variable context X, and a scope ω', and morphism application ω^v. Apart from morphism application, this is a fragment of the OPENMATH language [BCC+04].

Theory morphism declarations (or views) $v : T \to T' = \{Ass^*\}$ introduce a morphism with name v from T to T' containing a list of assignment declarations. Such a morphism must contain exactly one assignment $c := \omega'$ for every undefined symbol $c : \omega = \bot$ in T and some term ω' over T'. Theory morphisms extend homomorphically to a mapping of T-terms to T' terms.

Intuitively, a theory morphism formalizes a translation between two formal languages. For example, the inclusion from the theory of semigroups to the theory of monoids (which extends the former with two declarations for the unit element and the neutrality axiom) can be formalized as a theory morphism.

Every MMT declaration is identified by a canonical, globally unique URI.

MMT symbol declarations subsume most semantically relevant statements in declarative mathematical languages including function and predicate symbols, type and universe symbols, and — using the Curry-Howard correspondence — axioms, theorems, and inference rules. Their syntax and semantics is determined by the foundation, in which MMT is parametric.

3.1.2 MMT System

The MMT System [RK13] (MMT API) is a Scala-based [OSV07] open source implementation of the MMT language as described above. It also implements MMT-based knowledge management services and supports multiple backends for persistent storage as well as multiple frontends for machine and user access (see figure 2 for an overview of the architecture of the MMT system). Like the MMT language, the MMT system is implemented foundation-independently and all provided services carry over to the individual languages that are represented in MMT.

The supported backends are local storage (file system), SVN [Apa] repositories and TNTBase [ZKR10] databases. Knowledge items are retrieved on demand and once retrieved, they are cached in memory for efficiency.

The MMT frontends expose the functionality of the API and exist in two incarnations, an interactive shell (MMT Shell) and a RESTful inteface (MMT HTTP API). MMT Web is an web-based MMT application that uses the MMT HTTP API and offers interactive browsing using HTML + MathML and JOBAD [GLR09].

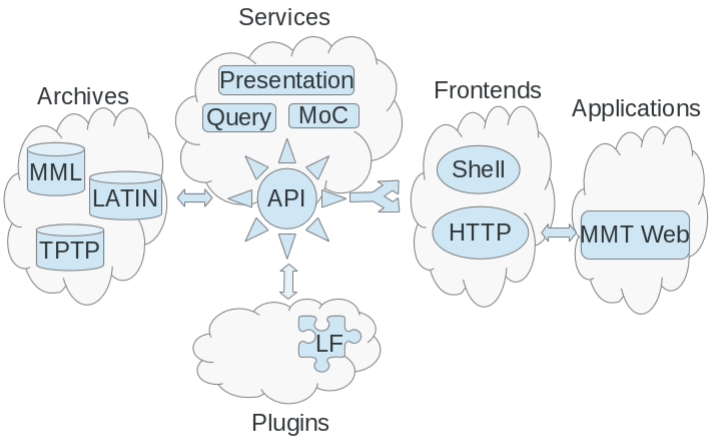

Figure 2: MMT System Overview

MMT knowledge management services include presentation [KMR08], querying [Rab12], change management [IR12a] and notation-based parsing [IR12b] (work in progress).

Therefore, the MMT system can provide integration between formal libraries and semantic applications by providing semantic services for formal libraries.

3.2 Flexiformality

I base my research on the idea of **flexiformality** [KK11b] meaning a generalization of the formal-informal dichotomy usually applied to mathematical content into a notion of flexible levels of formality. Intuitively, informal parts of a document are simply parts that have not been formalized (yet) and therefore correspond to a (potentially large) number of possible formalizations. Roughly speaking, if $I(D)$ is the set of formal interpretations of the document D then, if $I(D_1) \subset I(D_2)$, one can say D_1 is more formal than D_2.

This hints at a process of iterative formalization of scientific (in our case mathematical) documents through a posteriori formalization. In this process, mathematical documents can gradually evolve to be more formal (see figure 3, which shows the flexiformalization of a document yielding several *more formal* documents). This is particularly important because not all mathematical knowledge is meant to be formal, as mathematical documents also contains side-notes, comments or remarks

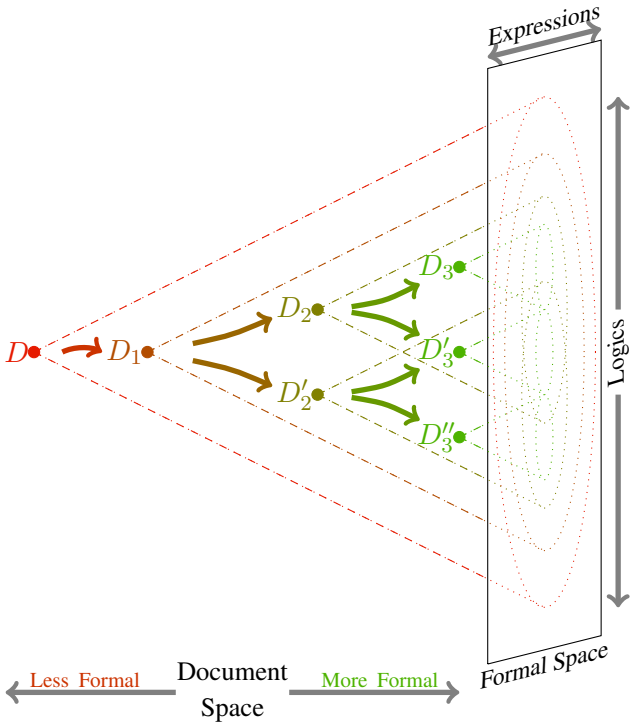

Figure 3: Flexiformal View of Document Formalization

directed at the human reader.

Active Documents Paradigm In order to enable optimal usage of flexiformalized knowledge by both humans and machines one needs to keep traditional documents as an interface and enhance them with semantic services that make use of the underlying content representation. In the active documents paradigm [KCD+11] (see figure 4) the knowledge base consists of semantically annotated documents together with a content commons that holds the corresponding background ontologies. Then, a *document player* relies on both the document and content commons to generate *active documents* for the user.

4 Motivation and Research Overview

As mentioned in section 2, a number of proof assistants exist and there are already relatively large libraries of formalized mathematics which include some fundamental mathematical results. However, only a small fraction of the total mathematical knowledge is formalized and, currently, formal systems are not very appealing to

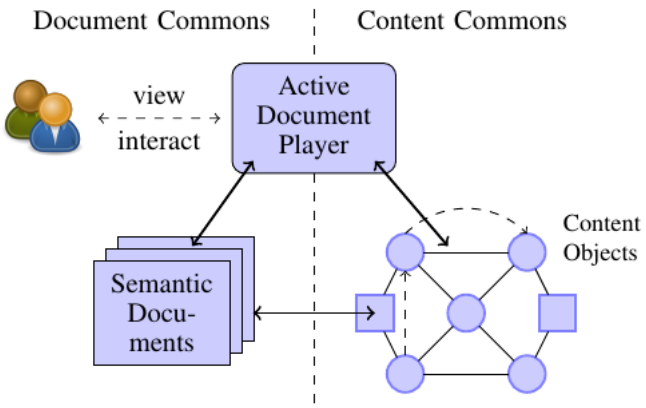

Figure 4: Active Documents Paradigm

mathematicians. This is because formalization is still extremely expensive and the few applications current formal systems provide do not appear to justify the effort of formalizing.

In contrast, flexiformalization is incremental, and, therefore, it needs only be done to the extent to which it is useful. This is crucial because, very often, useful services can already be provided for only partially formal content. However, I order for that to work in practice the fundamental pieces that enable flexiformalization must exist:

- A *formal language for flexiformal knowledge* that will permit the creation of flexiformal libraries which can contain knowledge of different levels of formality.
- A *semantic kernel for flexiformal knowledge* that will act a bridge between semantic applications and flexiformal libraries. The kernel will provide an abstraction layer for the concrete documents and make their content and semantics available to applications through semantic services.

4.1 A Formal Language for Flexiformal Knowledge

Some efforts to make formalized mathematics look more natural to mathematicians have already been made. For instance, Mizar uses a variety of operators and notations inspired by the mathematical vernacular and the result is that many theorems in the Mizar Mathematical Library can be recognized or understood by mathematicians without specific Mizar training (see figure 5 for an example). However, even though it can be understood, the Mizar specific syntax is still significantly different from mathematical notation.

```
theorem Th1:
  X c= Y & Y c= Z implies X c= Z
proof
  assume that
A1: X c= Y and
A2: Y c= Z;
  let x;
  assume x in X;
  then x in Y by A1,TARSKI:def 3;
  hence thesis by A2,TARSKI:def 3;
end;
```

Figure 5: Mizar Representation of Modus Barbara

Definition 1. *The number* e *is an important mathematical constant, approximately equal to* 2.71828*, that is the base of the natural logarithm. It is the limit of* $(1 + \frac{1}{n})^n$ *as* n *approaches infinity, an expression that arises in the study of compound interest, and can also be calculated as the sum of the infinite series* $e = 1 + \frac{1}{1} + \frac{1}{1\cdot 2} + \frac{1}{1\cdot 2\cdot 3} + \frac{1}{1\cdot 2\cdot 3\cdot 4} + \cdots$

Figure 6: Sample Informal Mathematical Definition

Projects like WTT, MathLang or Naproche attempt to act as an intermediary between the expressive yet ambiguous common mathematical language and the precise yet awkward formal languages. However, they mostly focus on the syntactic properties of mathematical notations rather than the high level modular structure of mathematical knowledge.

I feel that a knowledge base with interconnections as deep and complex as mathematics cannot be represented in any language that does not provide the kind of flexible module level notions that can capture mathematical intuitions. Furthermore, I consider that in many cases informality or ambiguity in day-to-day mathematics is actually a strength as it allows for more interpretations. In fact, informal mathematics leverages the ability of mathematicians to find valid interpretations easily in order to make concise statements that are both informative and mathematically sound (for those interpretations).

Consider for example definition 1 in figure 6 taken from Wikipedia [Wik]. It exemplifies several common practices of informal mathematics that are not currently adequately handled in formal languages.

Firstly, definition 1 has *ambiguous context references* and the *foundation is unspecified* as it avoids explicitly fixing many notions that occur in the definition. For instance, the types of e and n or the meaning of $+$. It is left to the reader to infer these details and to reach a valid, grounded interpretation of the definition. I

see (possibly generalized) theory morphisms as the right theoretical abstraction for representing this kind of interpretations by instantiating (mapping) undefined primitives in a theory to symbols in another theory (e.g. the $+$ from our definition to the symbol $+$ from the theory of natural numbers). In fact, I see theory morphisms as a fundamental part of informal mathematics that accounts for much of its ambiguity and flexibility. Therefore, a key theoretical goal of the research proposed here is to detect as many as possible of the numerous ways in which theories and theory morphisms can be used to capture and represent complex intuitions from informal mathematical and scientific documents.

Secondly, definition 1 serves a *dual role*, to certify to formal mathematical properties of an object (in this case, e) and to express the idea of the result in an informative and easily understandable way (see [Asp12] for an in-depth discussion of the dual role of mathematical proofs). These two roles have often diverging requirements. On the one hand, certification requires a precise, in-depth presentation of the properties or reasoning steps which implies a long and involved text. On the other hand, the role of a message involves presenting the key ideas in a human readable way, which typically implies a short text that omits technicalities. In our case, the technical statement "limit of $(1 + \frac{1}{n})^n$ as n approaches infinity" already defines e, but the rest of the definition ("is the base of the natural logarithm", "arises in the study of compound interest", etc..) provides additional information that is subjectively considered to be important or relevant. The last statement (" and can also be calculated as the sum of the infinite series $\cdots$ ") is particularly interesting as it is both a side comment and a semantically relevant statement expressing the equivalence of the two definitions[1].

Taking into account the problems and difficulties discussed above, I identify two key issues that a flexiformal language for mathematics must resolve.

The first key issue is the representation of the module level structure of mathematical knowledge. For instance, mathematical texts where the foundation is unspecified and which contain ambiguous context reference are omnipresent in mathematics and we already hinted above at theory morphisms as a potential solution to formally represent such ambiguities. Moreover, even though theories themselves appear simple, mathematicians also use a very flexible notion there: theories are rarely explicitly declared, imports are omitted, and interface-like theories (e.g. combinatorics, group theory) grow or change implicitly as new results are discovered.

The second issue is enabling representation of knowledge with a dual role (message and certification). From a markup perspective, certification corresponds to content elements that can represent the formal parts. Conversely, the role of a message corresponds to presentation elements that can represent the narrative structure and the informal parts of a document. Therefore, it is necessary to allow the mix of

[1] Note that this equivalence can also be represented using theory morphisms and used in practice by computing pushouts in the theory graph.

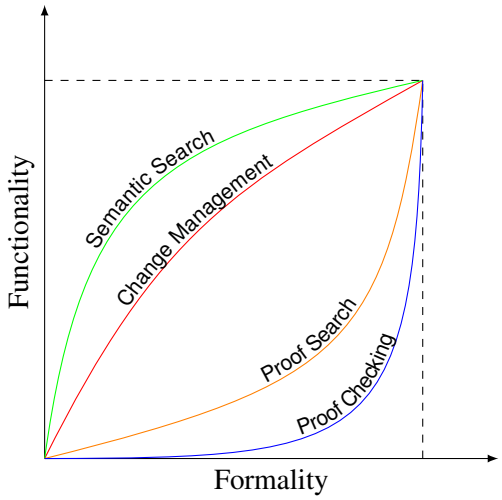

Figure 7: Functionality of Services in the Flexiformal Space

presentation and content elements. Both representations must be able to co-exist and, where necessary, refer to each-other. I believe the ability to combine presentation and content markup is necessary at all document levels (object, symbol, module) so that objects, symbols and modules can have narrative parts which are either not formalized yet or where the intuition of the semantics is presented in a human understandable way.

4.2 A Semantic Kernel for Flexiformal Knowledge

A fundamental advantage of formal content is that it can be understood and manipulated by computer-based systems. This results into a wide array of useful semantic services which can, potentially, be automated.

However, actually implementing such services can be expensive, especially for systems that were not built with this in mind. Therefore, given the relatively small size of formal mathematics corpora and the complexity of the existing systems, there has been little work in this direction.

Nevertheless, from a flexiformal perspective, such services, once implemented, are available for all flexiformal documents and gradually gain potency while moving from fully informal to fully formal in the flexiformal space. This has two important consequences.

Firstly, such services can still be useful for content that is only partially formalized. Figure 7 shows the amount of functionality of various semantic services at all levels of formality emphasizing that, in some cases, significant functionality can be provided with relatively little formalization work. Note that the graph only

takes into account the level of formality and not the individual difficulties inherent to each service. For instance, proof checking requires a fully formal theorem and proof while proof search, although conceptually more complex, only requires a fully formal theorem from so is less demanding in terms of formality.

Secondly, such services (being increasingly useful for more formal content – see fig 7) offer an important incentive for further formalization. I use formalization here to refer to *in-place formalization* involving a gradual increase in formality of a knowledge corpus by a systematic editing of individual knowledge items. This is opposite to the current dogma in formalized mathematics where formalization means a complete, bottom-up rewrite of documents in a certain fixed formal system.

5 Preliminary Work Plan and Schedule

The proposed research will be finished within three years. It will be conducted based on the goals outlined in section 4 and follow an application-driven course.

5.1 Work Plan

The concrete work plan consists of three distinct yet interrelated work areas. The first and second work areas involve designing the language and, respectively, system for flexiformal knowledge proposed in section 4. The third area involves evaluating the language and system by integration with libraries and applications. To take advantage of the interrelation between them, the three work areas will be developed in parallel and feed of each other.

5.1.1 Work Area I - *A Formal Language for Flexiformal Knowledge*

Clearly, designing a formal language suitable for conventional mathematics is an ambitious task. Therefore, I will not design the language from scratch but rather build upon two existing languages, namely MMT and OMDOC.

MMT is valuable for its focus on foundation independence and modularity but it is designed for formal knowledge. Therefore, it has little support for informal content and does not address many of the difficulties that occur in conventional mathematics. OMDOC, however, is interesting precisely because it already tackles these difficulties such as the narration/content distinction or the different levels of formality. Still, OMDOC lacks the strong formal basis and precise semantics of MMT.

Consequently, the resulting language can be seen either as ιMMT (an extension of the MMT language with informal aspects) or as OMDOC2 (an evolution of the

OMDoc language). For consistency, I will refer to it as the ιMMT language in the rest of this document.

I will organize the language design process in three work packages. First, I will examine other existing (semi)formal languages from a flexiformal perspective and, then, I will individually address the two key issues identified in section 4.1.

Work Package 1 *Investigating other (semi)formal languages*

I will begin by further studying other languages with similar design goals, with the purpose of finding useful features or insights. Specifically, I will focus on the following languages:

Mizar The design of the Mizar language is motivated by staying as close as possible to the mathematical vernacular. I have already extensively interacted with the Mizar language while working on translating the Mizar library to OMDoc format [IKRU12] and will make use of this experience.

WTT Similarly, Weak Type Theory (WTT) is a formal language for expressing mathematics that is close to the common mathematical language. WTT is also interesting because its type system is intentionally weak to lower the language restrictions so that it represents an intermediate step between informal and formal mathematical knowledge. Therefore, it relates to the idea of flexiformalization and I will further investigate this relation.

STEX Semantically annotated LaTeX is particularly interesting because a large part of the scientific community uses LaTeX for content authoring. Therefore, a successful flexiformal language would add to the usability of LaTeX the precision of formal languages. But this is precisely a design goal of STEX, which makes it a highly relevant case study.

Work Package 2 *Combining narrative and content oriented representations*

The markup language MathML already provides two ways to combine presentation and content markup, but only at the object level. The first is *mixed markup* where content and presentation elements are interspersed in one MathML (XML) tree. The second is *parallel markup* where both the presentation and content representations are explicitly provided resulting in two separate trees. I will generalize this to combine presentation and content markup, not only at the object level, but also at the module and statement level so that statements and modules can have narrative parts where the intuition of the semantics is presented in a human understandable way.

Work Package 3 *Capturing the modular structure of informal mathematics*

Theory morphisms are already useful for foundational ambiguity by permitting translations between foundations [IR11]. Furthermore, I discussed in section 4.1 how they might be useful as a solution for underspecification of symbol meaning by using them as interpretations. I will further investigate these two applications, search for other potential use cases and, if necessary, generalize the notion of a theory morphisms to capture more advanced mathematical intuitions. With respect to theories, I will look into, and possibly extend, the notion of realms introduced in [CFK13] as a solution for the complex theories occurring in informal mathematics (e.g. interface theories as discussed in 4.1).

5.1.2 Work Area II - *A Semantic Kernel for Flexiformal Knowledge*

Based on the MMT system and the ιMMT language, I will implement the ιMMT system. It will provide semantic services for flexiformal content and enable in-place formalization by supporting document evolution (through versioned content and authoring workflows).

The ιMMT system will act as a bridge between flexiformal corpora and semantic applications. From the perspective of the Active Documents Paradigm it can be seen as a basis for active document players.

I will organize the system implementation in three work packages. First, I will work on the system APIs to prepare for integration with libraries and applications. Then, I will implement the ιMMT language inside the system. Finally I will focus on developing semantic services, by extending the ones already implemented in MMT to support flexiformal content but also implementing new ones.

Work Package 1 *Enhancing the backend and frontend APIs*

Currently, MMT does not provide functionality to push back changes to the document stores. The temporary, in-memory storage can be modified, but the frontends expose little of that functionality. Therefore, permitting authoring workflows requires both extending the functionality provided by the backends and the functionality exposed by the frontends. Since authoring often involves many versions and several authors, I will look into the versioned links approach proposed in [KK11a] as a solution that will be implemented in the ιMMT system. Furthermore, this will be integrated with change management and the frontends will expose change functionality so that applications can request or push back changes when necessary.

Work Package 2 *Implementing the ιMMT language*

The ιMMT system will provide APIs and services for content represented in the ιMMT language and, therefore, the implementation needs to be synchronized with the language development. At each development iteration, the implementation will represent a first evaluation of the language in terms of adequacy and expressivity.

Work Package 3 *Developing semantic services*

The knowledge management services that already exist in MMT need to be updated or re-implemented to cover the added cases in the language grammar. For instance, the applicability of type checking to semi-formal content is conceptually difficult since the semantics of formal parts could depend on the context provided in the informal parts. Furthermore new services need to be developed tailored to the flexiformal aspect of the language. What exactly those services should be is still an open question but I expect that the process of integration with semantic applications will generate new ideas for useful services. Currently, candidates for useful new services are:

- *theory flattening* – projecting theories over views to generate new knowledge (e.g. carry theorems from group to integers by just viewing integers as a group)
- *theory clustering* – automatically organize sets of symbols in theories (using dependency information to optimally generate the theory graph).
- *view finder* – (semi)-automatically detect or create views between theories.

5.1.3 Work Area III - *Evaluation : Libraries and Applications*

I will evaluate the research described above on existing mathematical libraries and semantic applications. I will use the evaluation, not only as a final estimation of the thesis result, but as a continuous test of the designed methods that will fuel and guide development throughout the allotted time period. Therefore, I will place significant focus on the evaluation part as I plan to use an application-driven approach to achieve a result that is not only theoretically sound but also practically useful.

Work Package 1 *Applying to libraries*

As discussed in section 4.1, due to its flexibile and informal nature, the implicit structural properties of mathematical knowledge are difficult to detect or represent in an abstract way. Therefore, application to actual libraries is necessary to ensure the adequacy of the formalization and the usability of the implementation. I will translate several mathematical libraries of different levels of formality into the ιMMT language. Specifically, I will (where necessary) translate the corpora into markup compatible with ιMMT language and then import the result into the ιMMT system.

This will evaluate the flexibility and coverage of the language, the accuracy of the implementation and the backend infrastructure of the system. The corpora I plan to work with are the following :

LATIN The LATIN library is an LF-based formal logic atlas. It is interesting as a formal library because it is foundationally unconstrained, highly modular and strongly interconnected thus providing a fully formal test-bed of the kind of knowledge space that informal mathematics represents. Furthermore, the library can already be exported into MMT compatible OMDOC and already exists as an MMT library. Therefore it will help to ensure that the development of the flexiformal aspect does not reduce the expressivity of the language and the features of the system in the fully formal case.

MML The Mizar Mathematical Library is interesting because it is based on the Mizar language which, as mentioned before, is designed to closely mimic mathematical vernacular. This indicates that the Mizar system and library offer a unique perspective on the structure of informal mathematical knowledge. As mentioned before, there already is an export of the MML into MMT-compatible OMDOC format only without proofs [IKRU12]. This is because, while Mizar is fully formal, in the XML-based content export of the MML some proof steps that are automatically inferred by the Mizar system are not explicitly represented. Therefore I see the representation of the MML as a first step towards flexiformality offering the first real challenges in representing specific mathematical concepts and handling informality.

STC The STEX corpus contains semantically annotated LATEX and can be exported into OMDOC. It is especially interesting as a test case because the documents it contains have significant amounts of both semantic markup and informal text. Therefore it lies somewhere in the middle of the formal-informal dichotomy and provides a valuable test for the flexiformal approach.

ArXiv Finally, I plan to cover the ArXiv corpus as a large corpus of informal mathematical knowledge. I will make use of the work done within the ArXMLiv [SK08] project, which exports the ArXiv corpus to XML using LaTeXML [Mil]. In this case the purpose is not only to drive and evaluate the design of the language but also to use the services of the system to enable a posteriori formalization of the corpus.

Note that the complete formalization of the ArXiv corpus (or further formalization of the others) is not part of this proposal. The libraries will simply provide test cases of different degrees of formality and will be used "as is", (except potentially using results from other projects focused on semantization, e.g. [Gin12]). The ability to formalize "in-place" will be tested in a proof-of-concept manner on very small subsets of the libraries, but the full effort of formalization, while possibly integrated with this work, is external to this proposal.

Work Package 2 *Integrating with applications*

The functionality that the semantic services can provide will only be valuable in so far as it is useful to semantic applications. Therefore, I will integrate the ιMMT system discussed in section 4.2 with semantic applications that can make use of the services provided by the system. This will evaluate the overall system design and implementation as well as the individual services that are used.

As semantic applications are constantly being developed (e.g. the work discussed in [IR12b]), I will continue to actively search for additional integration opportunities but, currently, I plan to focus on integration with the following applications:

Planetary The Planetary [Koh12] system is a web-based active document player for semantically annotated document collections in Science, Technology, Engineering and Mathematics (STEM). It leverages semantic information to provide user assistance through browser-based user interaction. This provides a natural integration for the ιMMT system which can be a semantic service provider for Planetary.

SAlly The Semantic Alliance Framework [DJKK12] is a framework for semantic applications that complement existing software systems with semantic services and interactions based on a background ontology. Its focus on semantic services makes it another good opportunity for integration. ιMMT can function as a backend for Sally providing storage and semantic services while SAlly can focus on integration with the host system, user interface and user interaction.

5.2 Timeline

Figure 8 shows the timeline for the thesis. The allocated times for the three work areas and the various work packages are heavily interlaced to take advantage of the natural interconnections between them. However, the overall focus will slowly evolve during the three years from exploration and preparation to language design and service implementation and finally to integration and evaluation.

6 Expected Results and Applications

At the end of the three years I expect to have formal language capable of representing knowledge of all degrees of formality and which will permit complex combinations of narrative and content elements. Furthermore, it will include an advanced module level that captures at least those mathematical practices discussed in this proposal: ambiguous context references, unspecified foundation and interface theories. Moreover, I expect to have a system that implements this language, provides semantic services (e.g change management, presentation, definition expansion and lookup) and enables a process of gradual formalization of flexiformal knowledge.

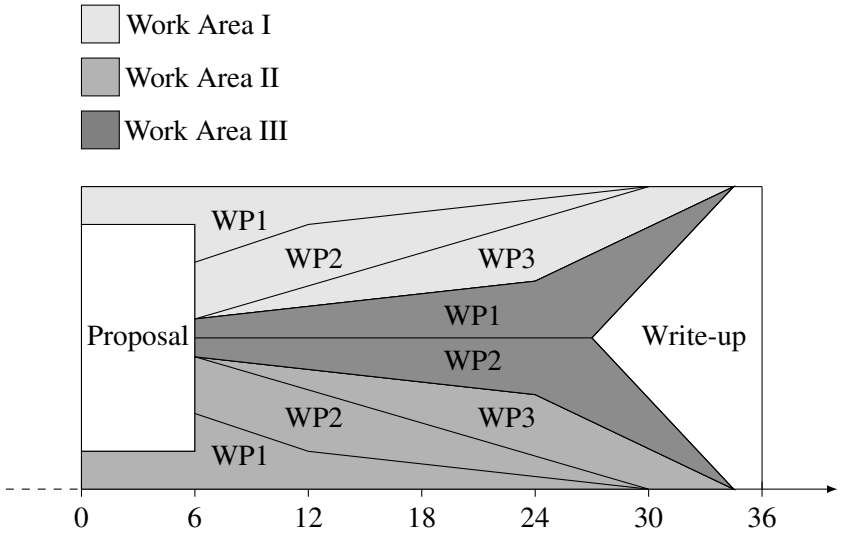

Figure 8: Thesis Timeline

Finally, I will integrate this system with existing mathematical libraries and with semantic applications.

This will permit library developers to progressively formalize their corpora using this system without having to rewrite entire libraries. At the same time it will allow application developers to easily develop semantic applications for a wide range of libraries by integrating with this system and using its services. I believe this will create a positive feedback loop that will incentivize both the formalization of libraries and the development of semantic applications. In the longer term it could help promote a much deeper integration of computers into mathematics in particular but also into science in general.

References

[ABR01] Ahmed Amerkad, Yves Bertot, and Laurence Rideau. Mathematics and proof presentation in Pcoq. In Uwe Egly, Armin Fiedler, Helmut Horacek, and Stephan Schmitt, editors, *Proceedings of the Workshop on Proof Transformation, Proof Presentations and Complexity of Proofs (PTP-01)*, pages 51–60. Universitá degli studi di Siena, 2001.

[Ano94] Anonymous. The QED Manifesto. In A. Bundy, editor, *Automated Deduction*, pages 238–251. Springer, 1994.

[Apa] Apache Software Foundation. Apache Subversion. `http://subversion.apache.org/`. seen February 2013.

[Arx] `arxiv.org` e-Print archive. seen February 2013.

[Asp12] Andrea Asperti. Proof, message and certificate. In Johan Jeuring, John A. Campbell, Jacques Carette, Gabriel Dos Reis, Petr Sojka, Makarius Wenzel, and Volker Sorge, editors, *AISC/MKM/Calculemus*, volume 7362 of *Lecture Notes in Computer Science*, pages 17–31. Springer, 2012.

[BBNS11] José Borbinha, Thierry Bouche, Aleksander Nowiński, and Petr Sojka. Project EuDML – a first year demonstration. In James Davenport, William Farmer, Florian Rabe, and Josef Urban, editors, *Intelligent Computer Mathematics*, number 6824 in LNAI, pages 281–284. Springer Verlag, 2011.

[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.

[BCC+04] Stephen Buswell, Olga Caprotti, David P. Carlisle, Michael C. Dewar, Marc Gaëtano, and Michael Kohlhase. The Open Math standard, version 2.0. Technical report, The OpenMath Society, 2004.

[CFK+09] Marcos Cramer, Bernhard Fisseni, Peter Koepke, Daniel Kühlwein, Bernhard Schröder, and Jip Veldman. The naproche project controlled natural language proof checking of mathematical texts. In Norbert E. Fuchs, editor, *CNL*, volume 5972 of *Lecture Notes in Computer Science*, pages 170–186. Springer, 2009.

[CFK13] Jacques Carette, William M. Farmer, and Michael Kohlhase. TetraPod Realms. TetraPod Blue Note, 2013.

[CFO10] Jacques Carette, William M. Farmer, and Russell O'Connor. The MathScheme Language. unpublished, 2010.

[CK07] Pierre Corbineau and Cezary Kaliszyk. Cooperative repositories for formal proofs. In Manuel Kauers, Manfred Kerber, Robert Miner, and Wolfgang Windsteiger, editors, *Towards Mechanized Mathematical Assistants. MKM/Calculemus*, number 4573 in LNAI, pages 221–234. Springer Verlag, 2007.

[DJKK12] Catalin David, Constantin Jucovschi, Andrea Kohlhase, and Michael Kohlhase. `Semantic Alliance`: A framework for semantic allies. In Jeuring et al. [JCC+12], pages 49–64.

[EuD] EuDML – the European digital mathematics library. `http://eudml.eu`. seen February 2013.

[Far10] William M. Farmer. Chiron: A set theory with types, undefinedness, quotation, and evaluation. SQRL Report 38, McMaster University, 2010.

[Fuc96] Benno Fuchssteiner et al. (The MuPAD Group). *MuPAD User's Manual*. John Wiley and sons, Chichester, New York, erste edition, March 1996.

[Gin12] Deyan Ginev. Designing Definition Discovery — Read, Recognize, Reflect, Repeat, 2012. Phd Research Proposal.

[GLR09] Jana Giceva, Christoph Lange, and Florian Rabe. Integrating web services into active mathematical documents. In Jacques Carette, Lucas Dixon, Claudio Sacerdoti Coen, and Stephen M. Watt, editors, *MKM/Calculemus Proceedings*, number 5625 in LNAI, pages 279–293. Springer Verlag, July 2009.

[Hal06] Thomas C. Hales. Introduction to the Flyspeck project. In *Mathematics, Algorithms, Proofs*, volume 05021 of *Dagstuhl Seminar Proceedings*, Dagstuhl, Germany, 2006. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany.

[Har96] John Harrison. HOL Light: A Tutorial Introduction. In *Proceedings of the First International Conference on Formal Methods in Computer-Aided Design*, pages 265–269. Springer, 1996.

[HM$^+$] Thomas C. Hales, Sean McLaughlin, et al. The Flyspeck project. `http://flyspeck.googlecode.com`. seen February 2013.

[IKRU12] Mihnea Iancu, Michael Kohlhase, Florian Rabe, and Josef Urban. The Mizar Mathematical Library in OMDoc: Translation and Applications. *Journal of Automated Reasoning*, 50(2):191–202, 2012.

[IR11] Mihnea Iancu and Florian Rabe. Formalizing Foundations of Mathematics. *Mathematical Structures in Computer Science*, 21(4):883–911, 2011.

[IR12a] Mihnea Iancu and Florian Rabe. Management of Change in Declarative Languages. In John A. Campbell, Jacques Carette, Gabriel Dos Reis, Johan Jeuring, Petr Sojka, Volker Sorge, and Makarius Wenzel, editors, *Intelligent Computer Mathematics*, pages 325–340. Springer, 2012.

[IR12b] Mihnea Iancu and Florian Rabe. (Work-in-Progress) An MMT-Based User-Interface. In *Workshop on User Interfaces for Theorem Provers*, 2012.

[JCC+12] Johan Jeuring, John A. Campbell, Jacques Carette, Gabriel Dos Reis, Petr Sojka, Makarius Wenzel, and Volker Sorge, editors. *Intelligent Computer Mathematics*, number 7362 in LNAI. Springer Verlag, 2012.

[KCD+11] Michael Kohlhase, Joe Corneli, Catalin David, Deyan Ginev, Constantin Jucovschi, Andrea Kohlhase, Christoph Lange, Bogdan Matican, Stefan Mirea, and Vyacheslav Zholudev. The Planetary System: Web 3.0 & Active Documents for STEM. *Procedia Computer Science*, 4:598–607, 2011. Finalist at the Executable Paper Grand Challenge.

[KK11a] Andrea Kohlhase and Michael Kohlhase. Maintaining islands of consistency via versioned links. In *Proceedings of the 29th annual ACM international conference on Design of communication (SIGDOC)* [SIG11], pages 167–174.

[KK11b] 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)* [SIG11], pages 181–188.

[KMR08] Michael Kohlhase, Christine Müller, and Florian Rabe. Notations for Living Mathematical Documents. In Serge Autexier, John Campbell, Julio Rubio, Volker Sorge, Masakazu Suzuki, and Freek Wiedijk, editors, *Mathematical Knowledge Management*, pages 504–519. Springer, 2008.

[Koh06] Michael Kohlhase. OMDoc – *An open markup format for mathematical documents [Version 1.2]*. Number 4180 in LNAI. Springer Verlag, August 2006.

[Koh08] Michael Kohlhase. Using LaTeX as a semantic markup format. *Mathematics in Computer Science*, 2(2):279–304, 2008.

[Koh12] Michael Kohlhase. The Planetary project: Towards eMath3.0. In Jeuring et al. [JCC+12], pages 448–452.

[KWZ08] Fairouz Kamareddine, J. B. Wells, and Christoph Zengler. Computerising mathematical text with mathlang. *Electron. Notes Theor. Comput. Sci.*, 205:5–30, 2008.

[Map] Maplesoft. `http://www.maplesoft.com/`. seen February 2013.

[Mat] Mathematica. `http://www.wolfram.com/products/mathematica/`. seen February 2013.

[Mil] Bruce Miller. `LaTeXML`: A LaTeX to XML converter. Web Manual at `http://dlmf.nist.gov/LaTeXML/`. seen February 2013.

[MRv] Mathematical reviews. http://www.ams.org/mr-database. seen February 2013.

[MWk] MediaWiki. http://www.mediawiki.org. seen February 2013.

[Ned] Rob Nederpelt. Weak Type Theory: a formal language for mathematics.

[NPW02] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*. Number 2283 in LNCS. Springer, 2002.

[OSV07] Martin Odersky, Lex Spoon, and Bill Venners. *Programming in Scala*. artima, 2007.

[Pau94] Lawrence C. Paulson. *Isabelle: A Generic Theorem Prover*, volume 828 of *Lecture Notes in Computer Science*. Springer, 1994.

[Pro] Wiki for formalized mathematics based on ProofWeb. http://prover.cs.ru.nl/wiki.php. seen February 2013.

[Rab12] Florian Rabe. A Query Language for Formal Mathematical Libraries. In John A. Campbell, Jacques Carette, Gabriel Dos Reis, Johan Jeuring, Petr Sojka, Volker Sorge, and Makarius Wenzel, editors, *Intelligent Computer Mathematics*, pages 142–157. Springer, 2012.

[RK13] Florian Rabe and Michael Kohlhase. A Scalable Module System. Manuscript, submitted to Information & Computation, 2013.

[SBA05] Jörg Siekmann, Christoph Benzmüller, and Serge Autexier. Computer supported mathematics with OMEGA. *Journal of Applied Logic, special issue on Mathematics Assistance Systems*, December 2005.

[SIG11] *Proceedings of the 29th annual ACM international conference on Design of communication (SIGDOC)*, 2011.

[SK08] Heinrich Stamerjohanns and Michael Kohlhase. Transforming the arxiv to XML. In Serge Autexier, John Campbell, Julio Rubio, Volker Sorge, Masakazu Suzuki, and Freek Wiedijk, editors, *Intelligent Computer Mathematics*, number 5144 in LNAI, pages 574–582. Springer Verlag, 2008.

[TB85] Andrzej Trybulec and Howard Blair. Computer Assisted Reasoning with MIZAR. In Aravind Joshi, editor, *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 26–28, 1985.

[Tea12] The Coq Development Team. The Coq Proof Assistant (Version 8.4) — Reference Manual – Chapter 16. http://coq.inria.fr/refman/Reference-Manual019.html, 2012.

[Urb06] Josef Urban. XML-izing Mizar: making semantic processing and presentation of MML easy. In Michael Kohlhase, editor, *Mathematical Knowledge Management, MKM'05*, number 3863 in LNAI, pages 346 – 360. Springer Verlag, 2006.

[WAB06] Marc Wagner, Serge Autexier, and Christoph Benzmüller. PLATO: A Mediator between Text-Editors and Proof Assistance Systems. *7th Workshop on User Interfaces for Theorem Provers (UITP)*, 174(2):87–107, 2006.

[Wen12] Makarius Wenzel. Isabelle/jEdit – a prover IDE within the PIDE framework. In Jeuring et al. [JCC+12], pages 468–471.

[Wie07] Freek Wiedijk. The QED Manifesto Revisited. In *From Insight to Proof, Festschrift in Honour of Andrzej Trybulec*, pages 121–133, 2007.

[Wik] Wikipedia, the free encyclopedia. http://www.wikipedia.org. seen February 2013.

[Wol] Wolfram—Alpha. http://www.wolframalpha.com. seen February 2013.

[ZBM] Zentralblatt MATH. http://www.zentralblatt-math.org. seen February 2013.

[ZKR10] Vyacheslav Zholudev, Michael Kohlhase, and Florian Rabe. A [insert XML Format] Database for [insert cool application]. In *XMLPrague 2010*. XMLPrague.cz, 2010.