Final Project Report
LATIN: Logic Atlas & Integrator
Acronym: LATIN

Reference number(s): KO 2428/9-1, MO 971/2-1

January 23, 2012

Michael Kohlhase
Jacobs University Bremen

Till Mossakowski
Universität Bremen

Contents

1 General Information (Allgemeine Angaben)

1.1 Reference Numbers (DFG Geschäftszeichen)

KO 2428/9-1, MO 971/2-1

1.2 Joint Proposal; Applicants (Antragsteller)

	Prof. Dr. Michael Kohlhase Professor of Computer Science 13. September 1964	Prof. Dr. Till Mossakowski Extraordinary Professor of Computer Science 25. October 1967
	Private Address (Privatanschrift):	
	Mühlental 5, 28717 Bremen Tel: +49 421 6396849	Bauernstr. 2, 28203 Bremen Tel: +49 421 9596505

1.3 Institute/Chair (Institut/Lehrstuhl)

	Jacobs University Campus Ring 1, 28759 Bremen Tel: +49 421 200 3140 Fax: +49 421 200 493140 m.kohlhase@jacobs-university.de	Universität Bremen Enrique-Schmidt-Str. 5, 28359 Bremen Tel: +49 421 214 64 226 Fax: +49 421 214 98 64 226 till@informatik.uni-bremen.de

1.4 Topic (Thema)

Developing methods and tools for interfacing logics and proof systems used in automated reasoning, mathematics, and software engineering

1.5 Report and Funding Period (Berichts- und Förderzeitraum)

1. Nov. 2009 - 31. Oct. 2011, cost neutral prolongment until 31. Oct. 2012

1.6 Research area and field of work (Fachgebiet und Arbeitsrichtung)

Scientific discipline: Computer Science
Fields of work: Automated Reasoning, Formal Methods, Knowledge Management, Artificial Intelligence

1.7 Application Areas (Verwertungsfelder)

Automated Reasoning, Formal Methods, Knowledge Management, Artificial Intelligence

1.8 List of Project-Related Publications

The publications whose references do not include a download URI have been added to the CD.

[1] M. Codescu, F. Horozal, M. Kohlhase, T. Mossakowski, and F. Rabe. „A Proof Theoretic Interpretation of Model Theoretic Hiding." In: *Recent Trends in Algebraic Development Techniques*. Ed. by H. Kreowski and T. Mossakowski. Vol. 7137. Lecture Notes in Computer Science. Springer, 2011.

[2] M. Codescu, F. Horozal, M. Kohlhase, T. Mossakowski, F. Rabe, and K. Sojakova. „Towards Logical Frameworks in the Heterogeneous Tool Set Hets." In: *Recent Trends in Algebraic Development Techniques*. Ed. by H. Kreowski and T. Mossakowski. Vol. 7137. Lecture Notes in Computer Science. Springer, 2011.

[3] Mihai Codescu and Till Mossakowski. „Refinement trees: calculi, tools and applications." In: *Algebra and Coalgebra in Computer Science, CALCO'11*. Ed. by Bartek Klin Andrea Corradini. Vol. 6859. Lecture Notes in Computer Science. Springer, 2011, pp. 145–160.

[4] Mihai Codescu, Bruno Langenstein, Christian Maeder, and Till Mossakowski. „The VSE Refinement Method in Hets." In: *Electronic Communications of the EASST* (2012). Accepted for publication.

[5] F. Horozal and F. Rabe. „Representing Model Theory in a Type-Theoretical Logical Framework.“ In: *Theoretical Computer Science* 412.37 (2011), pp. 4919–4945.

[6] M. Iancu and F. Rabe. „Formalizing Foundations of Mathematics.“ In: *Mathematical Structures in Computer Science* 21.4 (2011), pp. 883–911.

2 Progress Summary (Zusammenfassung)

LATIN aims at developing methods, techniques, and tools for interfacing logics and proof systems. Logics allow for making the mathematical knowledge at the core of science, engineering, and economics accessible to computational systems like (semi-)automated theorem provers, model checkers, computer algebra systems, constraint solvers, or concept classifiers. Unfortunately, these systems have differing foundational assumptions and input languages, which makes them non-interoperable and difficult to compare and evaluate in practice.

The LATIN project focuses on developing a foundationally unconstrained framework for knowledge representation that allows to represent the meta-theoretic foundations of the mathematical knowledge in the same format and to interlink the foundations at the meta-logical level. This approach of logics as theories leads to interoperability of both system behavior and represented knowledge.

LATIN has formulated a theoretical concept of *logical meta-framework* that combines the proof theoretic and the model theoretic approaches without being biased in any of the two directions. Within this framework, we have developed an atlas of logics and logic translations in a modular way, ranging from propositional logic, first-order, modal, description and higher-order logics to foundational set and type theories. The LATIN meta-framework is implemented in the MMT API. Therefore, the LATIN logic atlas can be browsed online using the MMT web server. This server provides semantics-aware interactive functionality, such as expression folding, hiding reconstructed subexpressions, type inference, and definition lookup.

The LATIN meta-framework has also been integrated into the Heterogeneous Tool Set Hets. As a result, Hets can now be extended with new logics in a declarative fashion, from their representation in a logical framework. Two such frameworks have also been implemented as new logics in Hets [11]. This means that the LATIN logic atlas has been combined with current theorem proving technology, and Hets can be used to perform truly heterogeneous proofs about heterogeneous theories formulated in logics from the atlas, as well as for finding models of logical theories in a modular way.

3 Progress Report (Arbeits- & Ergebnisbericht)

3.1 Initial Situation and Objectives (Ausgangslage)

Formal Methods is the field of formal specification and verification of software and hardware systems. It is without alternative in safety-critical or security areas where one cannot take the risk of failure. Formal method success stories include the verification of the Pentium IV arithmetic, the verification of the Traffic Collision Avoidance System TCAS, and various security protocols (e.g., [23, 15]). In many cases, only the use of logic-based techniques has been able to reveal serious bugs in software and hardware systems; in other cases, spectacular and costly failures such as the loss of the Mars Climate Orbiter could have been avoided by formal techniques.

Often, a single formal method cannot cover all aspects of a complex system. The need for the integration of different formal methods has led to several series of international conferences devoted to this topic such as IFM (e.g., [45]) and FroCoS (e.g., [57]). However, these integrations are always bi- or trilateral *ad-hoc* combinations, and not guided by a systematic integration approach. Likewise, other approaches like UML or the integration of external proof procedures into interactive provers either provide a semantic background or have the flexibility to support multiple formalisms, but not both.

Logical frameworks have been introduced in order to specify logics and related languages in a fixed meta-language. In particular, a logical framework is required for the semantically verifiable integration of formalisms because reasoning in the meta-logic is necessary to prove the correctness of integration. Logical frameworks have been used successfully for logic-independent investigations both on the theoretical (e.g., in the textbook [17]) and on the practical level (e.g., in the generic proof assistant [47]).

Formal logical frameworks like LF [21] and Isabelle [47] have focused mainly on proof theory and have only rarely taken model theory or logic translations into account. Model theoretic approaches like institutions [19], on the other hand, have developed various notions of translations, but these have not been formalized. Despite the overlap in motivation and applications, these two research communities are largely disjoint (meeting, e.g., in the LFMTP[1] and WADT[2] workshops, respectively) and often divided by conflicting philosophical perspectives. But without a logical framework that combines these features, attempts at building a formal library of logics (as attempted in Logosphere [48]), let alone their systematic integration cannot take off.

Moreover, research on logical frameworks has mainly focused on designing and implementing the meta-logic. Little attention has been paid to the development of large scale knowledge management services for them. While such services have been investigated in the CICM[3] community in principle, these solutions have not been integrated and systematically applied to a specific logical framework. Thus, logical frameworks lack the tool support needed for large scale success.

3.2 Project Developments (Durchgeführte Arbeiten)

The LATIN project has built on previous work of the proposers in the areas of (1) institutions and heterogeneous specification, (2) meta-logics and module systems, and (3) logical knowledge management.

LATIN's aim has been the development of an inventory of applied logics and their integration into a formal framework. This took both the form of an atlas for reference and documentation purposes and of a formal concept of heterogeneous specifications and proofs as a framework for formal development across multiple logics. The integration of logics includes not only the theoretical level but also extends to logics as they are parsed and presented by automated theorem proving tools.

Although the funds have been cut down to less than half, we have reached a good portion of this goal: We made progress in all of the work packages and developed a substantial logic atlas and corresponding tools. The remaining part that have not been tackled mainly concern scaling the atlas to logics of current interactive theorem proving systems with all their syntactic and semantic idiosyncrasies, especially integration of libraries of these systems, complex logic translations needed for translating among these systems, and flexible trust levels.

[1] `http://lfmtp.org/`
[2] `http://maude.sip.ucm.es/wadt2012/index.html`
[3] `http://trac.mathweb.org/CICM`

LATIN has continued and intensified the successful collaboration between Jacobs University Bremen and DFKI Bremen. Monthly project meetings serve organizational and technical discussions, which are often continued in ad-hoc meetings.

Most of the researchers and students employed in LATIN have participated in logic lectures held by the LATIN PIs about LATIN-related topics. The attendance has been cross-site in both directions, which fostered the collaboration between Jacobs and DFKI. This cross-site exchange also concerns students that have been employed as research assistants at the respective other site.

Fulya Horozal, who has been employed at Jacobs as a researcher in LATIN, works on her PhD thesis under co-supervision of the LATIN PIs, Michael Kohlhase and Till Mossakowski. Mihai Codescu has been employed as a LATIN researcher at DFKI, and works on his PhD thesis under co-supervision of Till Mossakowski and Andrzej Tarlecki. After his LATIN employment at Jacobs, Christoph Lange has changed to a position at University of Bremen under the direction of Till Mossakowski. This position is in the project OntoIOp, which applies LATIN results to an ISO standard about heterogeneous ontologies.

Results of LATIN have been published both in papers at international workshops and conferences and in journal papers, one of them [37] receiving the Best Paper Award at the MKM 2010 conference. Till Mossakowski and Mihai Codescu have presented the LATIN results in a lecture and a guided research seminar at Şcoala Normală Superioară Bucharest, invited by Dr. Răzvan Diaconescu.

Till Mossakowski has been the PC chair and (jointly with Mihai Codescu) organizer of the 20th Workshop on Algebraic Development Techniques (WADT 2010), a small international conference that has served as a place to present and exchange ideas and results of the LATIN project. Similarly, Florian Rabe (who contributed to the project at Jacobs University) organized the 3rd Workshop on Module Systems and Libraries for Proof Assistants (MLPA 2011) in conjunction with the 6th Workshop on Logical Frameworks and Meta-languages: Theory and Practice (LFMTP 2011).

3.2.1 Work Area I: Logic Atlas

An overview of the LATIN logic atlas was given in [7]. All encodings are available from the project website [36].

Work package I.1: First-order Logics First-order logics constitute a central part of the LATIN logic atlas. Most importantly, we gave a comprehensive and systematically modular representation of standard first-order logic in [27]. This representation also serves as a blueprint for further encodings. In particular, we extended it to sorted first-order logic and dependently-sorted first-order logic.

Going beyond plain first-order logic, the CASL family of first-order logics is a central part in the graph of logics and their translations of Hets. Often, the way to add proof support for a logic in Hets is via an appropriate encoding in CASL. Using the LATIN framework, new logics can be added to the logic graph of Hets from their representation in a logical framework.

We have represented CASL in LATIN using the logical framework LF. In doing so, we have followed the conventions and the presentation in [14] closely – in particular, this means that subsorting is added at a second stage and its semantics and proof calculus are obtained along the encoding of subsorting in multi-sorted first-order logic (without subsorting) using explicit embedding functions of the subsorts in their supersorts and membership predicates. This translation from theories with subsorting to theories without subsorting can be stated in LF as well by using a pattern-based functor (see WP II.2).

The formalization of Mizar is given in [29]. It consists of two parts. Firstly, Mizar constitutes a (first-order) logic in the LATIN logic atlas; this includes in particular the soft type system of Mizar. And secondly, we fix a theory in it – namely Tarski Grothendieck set theory – which serves as the root of the Mizar library.

To represent the model theory of logics, it was necessary to explicit formalize foundations of mathematics. This was also done in [29], where our representations of Zermelo-Fraenkel set theory, higher-order logic (see also WP I.2), and Mizar are given. For the exact use of foundations, see WP I.3 and II.2.

Work package I.2: Higher-order Logics In this work package we have concentrated on giving a modular representation of higher-order logics. Our representations distinguish between the underlying type theory and the logics. That way it is possible to combine different type theories with different logics.

Among the type theories, we represented the λ-cube [3]. This representation is fully modular so that the different corners of the cube can be combined in any way. We also represented a number of further orthogonal features of type theories such as union types, product types, etc. Similarly, the encoding of logics is fully modular. The features we support include in particular external propositions (where propositions are not terms of the type theory) and the more common internal propositions; intuitionistic and classical logic; description and choice operators. Moreover, we support different choices of primitive connectives, in particular Pravitz-primitives (implication and universal quantification) and Andrews primitives (equality).

From these primitives, we can combine a large family of logics that already includes HOL-Light [22] (which was initially not included in the LATIN proposal) and the Pure logic of Isabelle [47]. In addition, in [29] we extended the latter to a representation of Isabelle/HOL [46] and gave a translation to ZFC.

A representation of the logic underlying PVS and a systematic study of the translations between different variants of higher-order logics have been postponed.

Moreover, we have integrated the logic of HOL-Light in Hets. HOL-Light includes a well-developed formalization of real analysis, providing thus an important basis towards library integration; indeed, the challenge here is to study the relationship between the representation of real numbers in HOL-Light and another higher-order logic like Isabelle and conditions under which one can transfer results from one representation to another.

We have implemented an export utility for HOL-Light (in OCAML) that produces XML representations for HOL-Light theories. This representation is then read in by Hets and transformed in the internal Haskell data structures. Moreover, we have defined and implemented in Hets a logic translation from HOL-Light to Isabelle.

Work package I.3: Logical Frameworks in Hets The main objective here is to make adding logics and their translations to the logic graph of Hets more declarative, using the representation of logics in a logical framework. Hets already permits adding new logics by instantiation of a Haskell class, but this has the drawback that logics are not themselves represented as formal objects and cannot be automatically verified for correctness and moreover, they can only be added by Hets' developers.

Therefore, we developed the LATIN meta-framework – which abstracts from specific logical frameworks like LF [21] or Isabelle [47] (see WP II.2). We have implemented this meta-framework in Hets, which thus becomes flexible in the choice of the logical framework in which logics are represented.

Besides implementing the meta-framework in Hets, we must also provide a number of logical frameworks that serve as instances for the LATIN meta-framework. We have chosen the most important logical frameworks, the Edinburgh Logical Framework [21] and Maude [5]. (Isabelle [47] was already supported by Hets.)

Maude is a formal tool environment that can be used as a declarative programming language, as an executable formal specification language and as a formal verification system. The logic underlying the Maude system roughly is rewriting logic, a logic with good representation capabilities which make it a good candidate for a logical framework. Indeed, a number of logics, including equational, Horn or linear logic have been represented in rewriting logic [44]; moreover, rewriting logic is reflexive in the sense that one can represent rewriting logic within itself.

Our examples did not require translations between logical frameworks, and thus we have deferred studying them.

3.2.2 Work Area II: LATIN Framework

Work package II.1: Module Systems While meta-logical frameworks permit representation of model-theoretic logics, we want to extend this to the level of structured specifications in such logics. Here we use the MMT module system [55], which is of proof-theoretic nature and was already designed to be a comprehensive modular interchange language, and the CASL language for structured specifications [14], which is of model-theoretic nature.

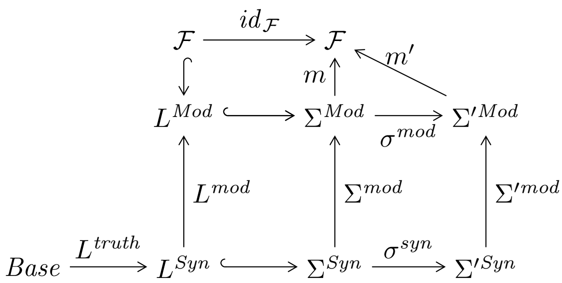
The logical framework LF [21] has been equipped with the MMT structuring language [53]. Moreover, a translation from the Isabelle module system to the LF module system was given in [52].

A particular challenge in integration the MMT and the CASL module systems was hiding, which restricts a specification to some export interface. MMT's proof theoretic nature leads to a theory-level semantics without support for hiding, as it is well-known that in a proof-theoretical setting, a structured specification with hiding can not be replaced equivalently with a flat theory.

In [6], we present an extension of MMT with support for hiding, by changing semantics to an inclusion of theories from the visible theory to the theory with hidden symbols. This idea, originating in [20], permits us to encode full CASL-style specifications in a proof-theoretic logical framework.

We have deferred the study of PVS. Moreover, we have not yet related the libraries of different theorem provers to each other; this will be a central issue of LATIN 2.

Work package II.2: Representing Logics The LATIN meta-framework was developed in [51] for the special case of using LF as a concrete language in which to declare logics. In [8], we generalized this by introducing a generic notion of logical framework. This notion is based on a declarative language given as a category C of theories. Our definition is general enough to cover the formal languages usually used as logical frameworks such as LF [21], Isabelle [47], and Maude [5].



The LATIN meta-framework follows a "logics as theories/translations as morphisms" approach such that a theory graph (i.e., a diagram in C) leads to a graph of institutions and comorphisms via a general construction.

More precisely, a logical framework provides a distinguished theory $Base$ that declares the primitive logical notions (in particular truth and consequence). Then a logic L is represented as a span $(L^{Syn}, L^{mod} : L^{Syn} \to L^{Mod}, L^{Pf}, L^{pf} : L^{Syn} \to L^{Pf})$ where L^{Syn}, L^{Pf}, and L^{Mod} represent syntax, proof theory, and model theory, respectively. Note that in the diagram, the proof theory is omitted for simplicity. Additionally, a morphism $L^{truth} : Base \to L^{Syn}$ defines how the primitive logical notions are realized in this specific logic.

Signatures or theories Σ of L are represented as certain extensions Σ^{Syn} of L^{Syn}. The LATIN meta-framework also provides a language of declaration patterns. By adding some pattern declarations to L^{Syn}, authors can specify which extensions Σ^{Syn} represent legal signatures. For example, for first-order logic, we use three patterns for functions symbols, predicate symbols, and axioms. The implementation is ongoing.

Σ^{Syn} induces theories Σ^{Mod} and Σ^{Pf} by pushout, which define the Σ-sentences, proofs, and models. While sentences and proofs are inherited from the framework via $Base$, the representation of models requires fixing an additional theory $\mathcal{F}$ that represents a foundation of mathematics (e.g., ZFC set theory). Then individual models of a signature Σ are represented as morphisms from Σ^{Mod} to $\mathcal{F}$.

Logic translations are represented accordingly using morphisms between the two diagrams representing the two logics. Thus, sentence and proof translation are represented as morphism application, and model reduction as morphism composition.

Currently the LATIN meta-framework still errs on the side of simplicity in the choice morphisms that represent logic translations. This was complicated by the fact that formal notions of logic translations are much more varied than of logics, which makes it harder to classify the existing translations. We have identified three classes of advanced logic translations (partial translations, type-coding translations and enumeration encodings) for which declarative support is desirable and feasible, but we have not yet been able to tackle these in detail. This is a central work package in the successor proposal L^{2}ATIN.

Work package II.3: Proof and Model Objects We have restricted our work here to languages for structuring models, with a focus on their use in algebraic specifications.

Architectural specifications [4] are a novel feature of CASL and have been introduced as means for describing the structure of the implementation of software systems. However, leaving out this perspective, architectural specifications can be regarded as a simple language for providing structure for models, instead of viewing them in a monolithic way. Recently [12], architectural specifications were complemented with a language for refinements, allowing thus to formally represent the entire stepwise development process; this language was however not supported by Hets. Let us stress that both the architectural and the refinement languages are independent of the formalism used at the level of structured specifications; in particular, they work over the Grothendieck institution built from the graph of logics of Hets, thus becoming heterogeneous languages.

In [10, 12], we introduce a sound calculus for checking correctness of refinements, together with an explicit notion of refinement trees, used for visualization of developments and for providing access points to their nodes. This has lead us to identifying a shortcoming in the semantics of one of the syntactical constructs of the architectural specification language - generic unit expressions, used to build generic units - and we provide a way to remedy it in [9]. Moreover, also in [10, 12] we derive from this a consistency check calculus for refinements, allowing in particular to reduce the task of checking consistency of large theories to organizing them in a correct architectural specification and checking consistency of the specification of each unit. Also, finding a model of the large theory reduces to finding models of the components of its architectural decomposition and then combining these models as prescribed by the architectural specification. This methodology has already been successfully applied (within a different project) for checking consistency of DOLCE [42]. Finally, in [13] we have investigated how the refinement notion of VSE [1] can be added in a non-disruptive manner to Hets, providing thus a first approach towards observational refinement, when an implementation is required to satisfy the requirements specification only up to observable behavior.

Future work includes a systematic study of the interplay between the architectural language for models and models-as-morphisms paradigm.

3.2.3 Work Area III: Tools

Two parallel lines of development have been integrated here: (1) the logical framework approach, with the Twelf tool for type checking, and (2) the institution approach, with the Heterogeneous Tool Set. Twelf has been extended with the MMT module system and thus grown into a LATIN validator that can check the logic definitions of the logic atlas. LF has also been added as a logic to Hets, and Hets calls the LATIN validator to check such specifications. Thus new logics can be added declaratively to Hets, via their LF specifications, and Hets provides bridges to existing theorem proving tools for these logics.

Work package III.1: LATIN Validator The MMT system [54] includes a parser and validator for LATIN developments represented in OMDoc. Validating a LATIN document consists of three stages of increasing strictness. Firstly, OMDoc-validity against the OMDoc RelaxNG schema [35] provides a simple sanity check. Secondly, structural validity provides a comprehensive validation of references to remote (i.e., anywhere on the web) knowledge items including those arising from imports, as well as type-checking all module level constructs. The implementation of structural validation is part of the MMT API. Since flexible trust levels have been deferred, they have been omitted here as well.

Thirdly, the MMT API includes a plugin interface through which the full validation of mathematical correctness can be implemented relative to the specific logical framework used. The user-provided plugins will only have to implement the non-modular aspects of the language because the MMT API can make it transparent to the plugin.

In particular, we implemented such a plugin for LF [21]. Moreover, Twelf has been extended with an MMT-compatible module system so that Twelf can export MMT documents directly. Therefore, for all content authored using Twelf, validation can be skipped entirely. However, in practice, only the third stage is skipped because the structural validation also yields valuable meta-data; these can be used, e.g., for the creation of indices and the building of archives. This has been employed in [28].

Since MMT uses OMDoc as its interface language, all OMDoc-level services can be applied directly to LATIN content. In particular, this is true for the services developed with LATIN itself in WP III.3/4. An overview of the MMT system and several case studies are presented in [37].

Work package III.2: Hets Extensions Here we describe how the LATIN meta-framework has been implemented in Hets.

At the syntactic level, we must enrich the underlying language of Hets with a way to write down new logic definitions. Since definitions of new logics have a different status than usual algebraic specifications, we extend the language at the level of libraries.

```
newlogic L =
  meta F
  syntax L^truth
  models L^mod
  foundation F
  proofs L^pf
```

We add therefore the following concrete syntax (on the right) in order to define new logics. Here L is the name of the newly defined logic and $\mathbb{F}$ is an identifier pointing to the logical framework used (e.g., LF, Isabelle or Maude). The identifiers $L^{truth}, L^{mod}, L^{pf}, \mathcal{F}$ are the components of the new logic L (cf. section II.2). They refer to previously declared signature morphisms of $\mathbb{F}$ and the signatures representing L^{Syn}, L^{Mod}, L^{Pf} can be inferred from them. $\mathcal{F}$ is a signature which gives the foundation. The syntax for logic translations is added in a similar way.

Hets analyzes such a logic declaration and automatically generates source files for the new logic, which is also added in the graph of known logics. After recompiling, the new object logic can be used to write specification like any other logic of Hets.

While we made it possible to add logics to Hets in a purely declarative way, further work is needed to turn this into a scalable tool. Firstly, the logic translations-as-theory morphisms approach needs to be generalized in order to cover more practically useful examples. Secondly, the new LF generated logics present in Hets need to be connected (via institution comorphisms) to the existing hard-coded logics in order to share the connection of the latter to theorem provers and other tools. Thirdly, it will be desirable to have a declarative interface for specifying the syntax of new logics, such that one is not forced to use the syntax of the logical framework. We are currently examining whether Eclipse and Xtext are helpful here. Finally, also the various tool interfaces of Hets should be made more declarative, such that Hets logics specified in a logical framework can be directly connected to theorem provers and other tools, instead of using a comorphism into a hard-coded logic. Then, in the long run, it will be possible to entirely replace the hard-coded logics with declarative logic specifications in the LATIN meta-framework — and only the latter needs to be hard-coded into Hets.

Both the LATIN meta-framework and the Hets implementation are parametric in the concrete logical framework $\mathbb{F}$. To provide such concrete instances, we proceed in two steps: Firstly, a dedicated implementation for $\mathbb{F}$ reads a human-friendly logic definition and exports it in a machine-friendly format. Secondly, Hets reads the latter. This is necessary to reuse mature and sophisticated implementations for $\mathbb{F}$, in particular, parsing and type reconstruction.

In the case of LF, we use Twelf as the dedicated implementation, which exports logic definitions as OMDoc. Similarly, an export of Maude modules has been implemented directly in the Maude system, making use of its reflective capabilities.

Moreover, we have also improved proof support for Maude by implementing a translation to CASL and developing a proof methodology for Maude modules (see [11]). Finally, our work in architectural specifications and refinements is fully supported by Hets: the analysis of architectural specifications has been corrected according to the changes proposed in [9], the static analysis of the refinement language has been implemented in Hets together with the proof calculus and the conservativity calculus for refinements. Moreover, the integration of VSE into Hets is completed.

While the integration of Twelf in Hets relies on the OMDoc format, we still have to extend Hets with capabilities to export from its built-in logics to OMDoc, for a better tool interoperability. An integration of SMT solvers into Hets is also currently missing.

Work package III.3: Database & Portal For this work package, we have concentrated on concepts and solutions of storing and managing OMDoc/MMT represented collections of files. We took the versioned XML database TNTBase [60] as a basis and extended it with an OMDoc/MMT Web API [61] that integrates fragment access (dereferencing MMT URIs), aggregation (e.g. generating OMDoc documents from fragments [59, 24]), and caching (e.g. of type judgments). In particular, the MMT validator from WP III.1 was integrated into the TNTBase to run it as a web service near the data to eliminate web latency incurred by multiple database lookups. Special care was taken to achieve incrementality of all computation in order to ensure scalability; see [37] (which won the "best paper award" at MKM 2010).

The MMT web server [54] can thus serve as a first incarnation of the LATIN web portal, it provides access to the OMDoc sources and presentations of the LATIN logic atlas. The presentations are generated on the fly from the OMDoc sources in TNTBase by the MMT API as dynamic XHTML+MathML, which supports interactions like user-adaptive elision (and reconstruction) of brackets and inferable arguments. The full types and terms in LF involve many large arguments that cutter the formalization. Experts do not want to write or read them (therefore Twelf allows reconstructing them), but non-experts need them to understand the formalization. Making their presentation adaptive solves a long-standing problem in LF-based logic representations.

We have experimented extensively with online editing facilities for LF-based logic formalizations in the form of integrated (logic) development environments (IDEs) that support the semantics of LF and OMDoc, but we have not arrived at a fully satisfactory solution yet. The same holds for navigation in the LATIN logic atlas; some form of machine support (the atlas comprises more than 1000 theories and views) will be crucial for the manageability in the future.

A final shortcoming of the current LATIN portal is that it concentrates completely on the formal aspects of logic representations and translations, even though they are almost unintelligible without some form of (integrated) documentation. This documentation currently only exists in the form of scientific papers "about the formalization of logics", such as [46, 29], which could in turn profit from a direct interaction with the fragments of the LATIN logic atlas they describe. Moreover, an integration of documentation and formalization would be extremely beneficial in the development process of logic formalization, which is often collaborative and usually proceeds in formalization stages over an extended period of time.

To describe this process, we have developed the notion of *flexiformal* (i.e. at flexible levels of formality) representations [38, 34], which conceptualizes the possibility to mix formal and informal content in OMDoc at every level. The MMT format used in LATIN (and developed as a nucleus for the OMDoc 2 format) does not have that facility yet. Therefore we have developed a portal for flexiformal content in parallel to the LATIN portal in the form of the Planetary system [49, 16, 43, 41, 2], which shares the storage level of the LATIN portal, but uses sTeX (a semantic version of LaTeX that can be converted into OMDoc; see [56, 39] for details) as a surface language. Planetary also integrates advanced editing features (for sTeX) and Web 3.0 features (user-supplied content and localized commenting). For the future we envision an integration of the LATIN portal with the planetary system in the form of an "Open Archive of Flexiformalizations" (OAFF), which combines the formal aspects of the LATIN portal already achieved in the LATIN project with the flexiformality in Planetary. But before we can achieve this integration we have to extend MMT with informal aspects (to progress on the way to OMDoc2), and reconcile the current surface languages LF and sTeX, as well as the transformation pipelines based on Twelf and LaTeXML.

Work package III.4: Search For this work package we have concentrated on the question of how to make the existing unification-based search technology in the MathWebSearch system [40] can be made useful in the context of the LATIN logic atlas. As we had a translation of the Mizar library into OMDoc, we developed an OMDoc crawler for MathWebSearch. In the process of this, we realized that we had to re-implement the system to utilize XML-based REST-full inter-process communication of the components. The new system [50] uses standard formats like MathML throughout and is much more resilient than the old system. We were able to fully index the Mizar library (without proofs) on a standard laptop. Based on this, we are working on extending the MathWebSearch querying facilities so that we can use the full power of unification to discover "applicable theorems", e.g. if a user looks for an upper bound of a formula $\mathbf{A}$, then he can query for any theorems of the form $\overline{\forall}\mathbf{B} \leq \mathbf{C}$. A solution would be a unifier σ of $\mathbf{A}$ and $\mathbf{B}$ and the upper bound of $\mathbf{A}$ would be $\sigma(\mathbf{C})$. We are implementing this querying scheme for OMDoc and have a first prototypical integration of the the Applicable Theorem Search into the MizarWiki system; see [30, 31] for details.

3.3 Results (Erzielte Ergebnisse)

The LATIN proposal formulated three objectives, and we will now summarize the main achievements of the project in terms of these. We should stress though that the objectives are long-term objectives spanning more than one specific research project. During LATIN, we have mainly covered objective O1, and partly O2 and O3.

3.3.1 O1: Logic Atlas

By a logic atlas, we mean a graph that describes logics and their interrelations. The novelty of our approach is that logics are treated as theories of a meta-logic, and theory morphisms of that meta-logic are used to relate them to each other. The main difficulty here is (i) to find the selection of logics that are important enough to be included in the initial logic graph, (ii) to identify their inheritance relationships, and (iii) to design and implement the translations between them.

- We have formulated a theoretical concept of a *meta-logical framework* that combines the proof theoretic and the model theoretic approaches without being biased in any of the two directions. This LATIN meta-framework is parametric in the specific formal language that is used as the meta-logic. We have implemented the meta-framework in Hets along with instantiations for specific meta-logics [8].

- Within this framework, we have developed an atlas of logics and logics translations based on the meta-logic LF. Currently, the logic atlas contains highly modularized formalizations of various logics, type theories, foundations of mathematics, algebra, and category theory. Among the formalizations of logics are propositional (PL), first-order (FOL) and higher-order logic (HOL), sorted ($SFOL$) and dependent first-order logic ($DFOL$), description logics (DL), modal (ML) and common logic (CL). Among the foundations are encodings of Zermelo-Fraenkel set theory, Isabelle's higher-order logic, and Mizar's set theory. Notable special cases were published as [27] and [29]. A high-level overview of a fragment of the logic atlas is given in the left part of Figure 1. The whole graph is significantly more complex as we use the LF module system to obtain a maximally modular design of logics. For example, propositional, higher-order, modal, and description logics are formed from orthogonal modules for the individual connectives, quantifiers, and axioms. For example, the $\wedge$-connective is only declared once in the whole logic atlas and imported into the various logics and foundations and related to the type theoretic product via the Curry-Howard correspondence. Moreover, we use individual modules for syntax, proof theory and model theory so that the same syntax can be combined with different interpretations. For example, the formalization of $\wedge$ consists of the signatures $\wedge^{Syn}$ for syntax containing the connective itself, $\wedge^{Pf}$ for proof theory containing natural deduction style inference rules for $\wedge$, and $\wedge^{Mod}$ for model theory containing a meta-language (in this case ZFC) to axiomatize the properties of models of $\wedge$, an interpretation of $\wedge$ in ZFC and axioms specifying its truth values. The whole modularized logic atlas comprises over 1000 LF theories and morphisms.

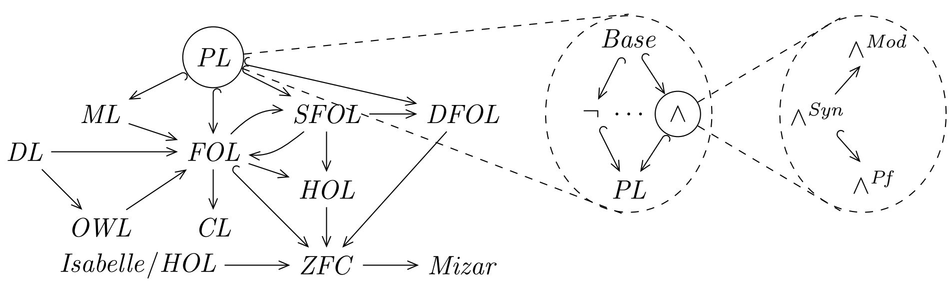


Figure 1: Logics and their Interrelations in the LATIN Logic Atlas

- The LATIN meta-framework is implemented in the MMT API [54]. Therefore, the LATIN logic atlas can be browsed online using the MMT web server. This server provides semantics-aware interactive functionality such as expression folding, hiding reconstructed subexpressions, type inference, and definition lookup. An example is shown in Figure 2. The LATIN logic atlas is browsable at the project homepage.

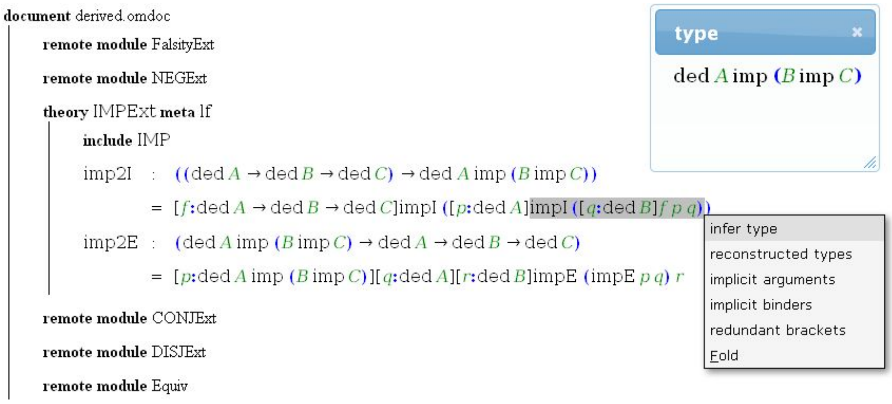


Figure 2: The MMT Webserver with Type Inference Functionality

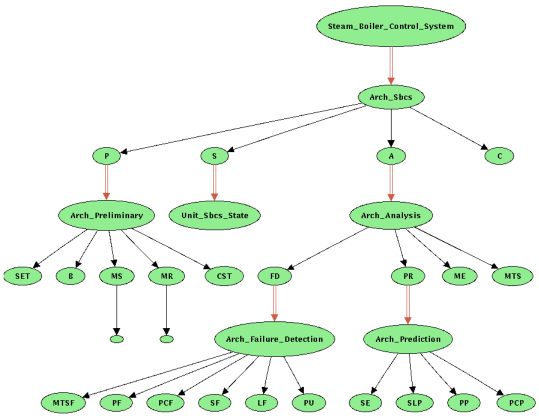


Figure 3: A Refinement Tree in Hets

3.3.2 O2: Heterogeneous Proofs, Models, and Libraries

A heterogeneous proof will be a proof object that contains homogeneous proofs as sub-objects connected by special proof steps. Heterogeneous models will be the counterpart of heterogeneous proofs, used as examples and consistency proofs of heterogeneous theories, and as disproofs of heterogeneous conjectures. Heterogeneous libraries will permit to strengthen the links between logics beyond logic translations.

The CASL architectural language has been complemented with a refinement language and with a proof calculus for checking correctness of refinements [9, 12]. CASL refinements are now fully supported by Hets. The tool also provides a visual representation of the modular construction of models for a logical theory, under the form of refinement trees (see Fig. 3, where red links denote refinements between specifications and black links denote decompositions of models). Architectural refinements also have been used for an application in a different project, namely for modular model finding for an upper ontology that is to large to be tackled in a non-modular way [42]. The architectural refinement language provides refinement of theories and structuring for models independently of the logic, and can be also used heterogeneously for different logics of the logic atlas.

3.3.3 O3: Heterogeneous Verification and Flexible Trust Levels

The correctness of a heterogeneous development involving theories, proofs, and models can be checked in several ways: truly heterogeneously or homogeneously after translation to a uniform logic; fully formally or with semi-formal and informal parts. In the long run, we intend to support all of these.

The LATIN meta-framework has been integrated into Hets [8]. As a result, Hets can now be extended with new logics in a declarative fashion, from their representation in a logical framework like LF or Maude. The latter two have also been implemented as new logics in Hets [11]. This means that the LATIN logic atlas has been combined with current theorem proving technology, and Hets can be used to perform truly heterogeneous proofs about heterogeneous theories formulated in logics from the atlas.

3.4 Future Work (Ausblick auf zukünftige Arbeiten)

We have now developed the expertise that allows us to represent large-scale logics, providing advanced proof-support, a stable community of users and a large library of formalizations. We aim to fulfill the long-term objectives of LATIN by expanding the LATIN meta-framework to a framework for library integration in the proposed successor project LATIN 2. Moreover, LATIN 2 will build an an archive system OAFF that offers cross-system suport of e.g. management of change, search, documentation, and dissemination.

Till Mossakowski leads the ISO Standard Development Initiative 17347 *Ontology Integration and Interoperability (OntoIOp)* (ISO/TC37/SC3/WG3, current status is Working draft), and Christoph Lange is employed to work on this full-time. In OntoIOop, a *distributed ontology language* (DOL) for heterogeneous ontologies is being developed. DOL will allow for the verbatim integration of existing ontologies, translate ontologies across different formalisms, match ontologies, extract modules etc. The results of LATIN will provide an important foundation for DOL.

The work on LATIN has substantially improved our understanding of structured knowledge representation languages in general. We will employ this understanding in the continuous evolution of the OMDoc language [35]. In particular, OMDoc 2 will integrate the LATIN meta-framework for the representation of logics, logic translations, and models as developed in WP II.2.

3.5 Interdisciplinary Development (Interdisziplinäre Weiterentwicklung)

In the SFB/TR 8 “Spatial cognition” (project I1-[OntoSpace]), in cooperation with John Bateman, professor of linguistics at the University of Bremen, we will use different ontology languages for specifying spatial ontologies. Their heterogeneous integration is based on the foundational results of LATIN.

The concepts and structures of the MMT logic/theory graph approach developed in the LATIN project have been used to describe the mathematical practice of “framing” (viewing a mathematical object o as another that we already understand, thus gleaning insights on the nature of o that way) in terms of theory morphisms. This has been used to generate user-adaptive help messages for users with differing prerequisite knowledge in spreadsheet-based financial controlling systems [33].

We are currently working on employing MMT-based framing in the context of serious games for STEM. There, MMT theories will be employed as knowledge representation for game objects, game situations, and the mathematical theory employed in problem solving. Here MMT inclusions structure the knowledge, and views are used to model framing and the analogical transfer of mathematical solutions to (semantically annotated) game scenarios. This work is still very much at the beginning, but it looks as if many insights gained in the LATIN project (e.g., theory pushouts as solution situations) can be transferred to this setting.

3.6 Potential for Exploitation (Verwertungspotenzial)

LATIN is a foundational project. Commercial exploitations may exist in fields where logics play a crucial role. For example, in the field of ontological modeling, OntoIOp (see above) is expected to cover also interoperability of services and devices (which often use different interfaces and languages). These applications are medium to long term.

3.7 Contributors (Beteiligte Wissenschaftler)

3.7.1 Employed researchers

Mihai Codescu MSc. (DFKI), 01. Feb 2010 - 31. Oct 2012
Mihai Codescu conducted research in all work areas, primarily on work packages I.1, II.3 and III.2.

He was in particular responsible for extending the Heterogeneous Tool Set Hets in such a way that logics from the LATIN logic atlas can be read in and used for Hets.

Feryal Fulya Horozal MSc. (Jacobs University), 16. Jan 2010 - 30. Apr 2012
Feryal Fulya Horozal conducted research in all work areas, primarily on work packages I.1, I.2, and II.2. In work area I, she was mainly responsible for the parts of the LATIN logic atlas pertaining to first-order logic and the type theories of the λ-cube. In work area II, she was mainly responsible for the part of the LATIN framework pertaining to declaration patterns.

Dr. Christoph Lange (Jacobs University), 01. Nov 2009 - 31. May 2011
Christoph Lange conducted research on work package III.3, helping to investigate flexiformality, in particular, the connection of the LATIN framework to Semantic Web Technologies (linked open data).

3.7.2 Employed research assistants

Catalin David, B.Sc. (Jacobs) Catalin David worked on the LATIN portal, WP III.3. where he was responsible for prototyping the integrating with the Planetary system and developing novel approaches for interactive graph views.

Ştefania Dumbravă (DFKI*) Ştefania Dumbravă has been improving the OMDoc export from Hets.

Deyan Ginev, B.Sc. (Jacobs) Deyan Ginev worked on the LATIN portal, WP III.3. concentrating on flexiformalization, and metadata support in the LATIN portal.

Mihnea Iancu, B.Sc. (Jacobs) Mihnea Iancu represented specific foundations in the LATIN logic atlas, WA I (in particular ZFC set theory and Mizar and HOL-related languages) and implemented an import of the Mizar library into the LATIN framework.

Iulia Ignatov (DFKI*) Iulia Ignatov helped specifying first-order logics in the LATIN logic atlas (WP I.1), as well as programmed tool support for interfacing the LATIN logic atlas with Hets (WP III.2).

Dimitar Misev, B.Sc. (Jacobs) Dimitar Misev implemented tool support in WA III, in particular, prototypical translations from the TPTP and SUMO to MMT and OMDoc.

Kristina Sojakova, M.Sc. (DFKI*) Kristina Sojakova worked on the Hets-LATIN integration (WP III.2).

Jonathan von Schröder (DFKI) Jonathan von Schröder has been working on the integration of HOL-Light and on developing a parser for Isabelle theories.

Silviu Oprea (DFKI*) Silviu Oprea worked on the LATIN portal, WP III.3. in particular on the translation and interactive presentation of diagrams.

Stefan Mirea (Jacobs) Stefan Mirea worked on the LATIN portal, WP III.3. in particular on a JavaScript-based framework for the integration of semantic services into formula (and document) presentations.

*: students enrolled at Jacobs University that were employed by DFKI

3.7.3 Contributing researchers at the project sites

Florian Rabe, Ph.D. (Jacobs University)
Florian Rabe has contributed substantially to all work packages except II.3 both through his own research and through the supervision of students. He was primarily concerned with the design of the LATIN framework, its LF incarnation, and the LATIN logic atlas as well as the MMT language, API, and web server.

Dr. Christian Maeder (DFKI)
Christian Maeder is the main programmer of the Heterogeneous Tool Set Hets and provided various kinds of help and supervision to researchers and students working on Hets.

3.7.4 National and international collaborators

Prof. Alan Bundy, Dr. Bogdan Grechuk, University of Edinburgh, UK Alan Bundy's research group is working on evolution of higher-order ontologies for physics. They need a way to integrate the libraries HOL light, which are the best existing ones for real analysis, with Isabelle/HOL, which provides better user interface and integration with external solvers than HOL Light. Dr. Bogdan Grechuk from Bundy's group (now in Leceister) visited DFKI in 2010.

Dr. Răzvan Diaconescu, Şcoala Normală Superioară Bucharest, Romania Till Mossakowski has a long-term collaboration with Răzvan Diaconescu about institutions and institution morphisms. He sent Codruţa Gîrlea to Bremen to complete her master's thesis under the supervision of Till Mossakowski.

Prof. William Farmer, Prof. Jacques Carette, and Russell O'Connor, Ph.D. McMaster University, Hamilton, Ontario, Canada Prof. Kohlhase and Dr. Rabe initiated a long-term collaboration with these contributors to investigate theory and theory morphism-based knowledge representation and system integration. So far, two meetings have taken place in the form of non-public workshops: a 2-week workshop (January 2011) hosted by the Canadian contributors and a 1-day workshop (July 2011) during the Conference on Intelligent Computer Mathematics (CICM 2011) in Bertinoro, Italy. A third meeting is planned during CICM 2012 in Bremen.

Bruno Langenstein, Dipl. Inform., DFKI Saarbrücken, Germany Bruno Langenstein is the main programmer of the Verification Support Environment (VSE), relevant for WP II-3, which he helped us to link to the Heterogeneous Tool Set Hets.

Sebastian Reichelt, Dipl. Inform., Karlsruhe Mr. Reichelt is the developer of the HLM system, a young proof assistant with an innovative user interface. We initiated a collaboration in order to integrate it with the LATIN tools.

Dr. Adrián Riesco, Universidad Complutense de Madrid, Spain Adrián Riesco has done a PhD thesis about the integration of Maude and Hets.

Prof. Carsten Schürmann, IT University, Copenhagen, Denmark Prof. Kohlhase, Dr. Rabe, and Prof. Schürmann have a long history of collaboration including research visits and joint projects. The LATIN project is in part a continuation of the LogoSphere project [48]. Specifically, for LATIN, Prof. Schürmann contributed the design of the LF module system (see [53]) that was a part of the LATIN framework designed in WA II. During the reporting period, this was mediated through a 1-month research visit of Dr. Rabe in Copenhagen (June 2010) and a 1-week research visit of Prof. Schürmann in Bremen (January 2012).

Josef Urban, Ph.D., Radboud University, Nijmegen, The Netherlands Dr. Urban is one of the experts on the Mizar language and visited Jacobs University for 2-week period (August 2010). This visit resolved many of the theoretical and technical issues regarding the import of the Mizar library into LATIN (WP I.1). A preliminary version of this import led to one joint paper [32].

3.8 Qualification of Junior Researchers (Qualifikation des wissenschaftlichen Nachwuchs)

In general, we have established a group of competent researchers, a task that is particularly difficult in the theoretical subjects of this area.

3.8.1 Jacobs University

In particular, at Jacobs University, there are now 1 post-doctoral researcher, 2 Ph.D. students, 3 M.Sc. students, and 2 B.Sc. students, who are primarily involved with foundational research that is directly relevant for LATIN. Most of them have been involved for several years. Their work has been integrated by forming the theory subgroup within Prof. Kohlhase's research group. This group is led Dr. Florian Rabe. Moreover, LATIN has generated use cases for several M.Sc. and Ph.D. students in related fields. In multiple cases, this has led to student-co-authored papers at international journals [29, 27], conferences [7, 8, 6, 28, 58, 37], and workshops [25, 18, 26].

LATIN-related theses at Jacobs University started or completed during the reporting duration:

Maria-Alexandra Alecu B.Sc. 2009-2012: Plugin-based Type-Checking in Logical Frameworks (working title)

Catalin David B.Sc., 2007-2010: Interactive Documents and Computer Algebra Systems: JOBAD and Wolfram—Alpha M.Sc., 2010-2012: Integrating Semantic Services in Desktop Applications (working title)

Deyan Ginev M.Sc., 2009-2011: The Structure of Mathematical Expressions; Ph.D., 2011-: TBD

Ştefania Dumbravă B.Sc. (math), 2007 - 2010: Structured Specifications with Hiding in the Edinburgh Logical Framework LF M.Sc., 2010-2012: Reflecting Theories (working title)

Mihnea Iancu B.Sc., 2007-2010: Formalizing Foundations of Mathematics M.Sc., 2010-2012: Management of Change in Declarative Languages

Iulia Ignatov B.Sc. 2009-2012: Modular Representation of Type Theories (working title)

Feryal Fulya Horozal Ph.D., 2008-: Representing Declarative Languages and Their Translations

Figen Füsun Horozal M.Sc., 2009-2012: Management of Change in OWL Ontologies

Alin Iacob M.Sc., 2009-2011: Towards Project-Based Workflows in Twelf Ph.D., 2011-: TBD

Christoph Lange (now University Bremen) Ph.D., 2006-2011: Enabling Collaboration on Semiformal Mathematical Knowledge by Semantic Web Integration

Dimitar Misev B.Sc. 2007-2010: Integrating SUMO and OMDoc

Mihaela Rusu B.Sc. 2008-2011: Interactive Semantical Graphs

Kristina Sojakova (now Carnegie Mellon University) M.Sc., 2008-2010: Mechanically Verifying Logic Translations

Vladimir Zamdzhiev (now University of Oxford) B.Sc. (math) 2008-2011: Formalizing Syntactical Objects within Formalized Set Theory B.Sc. (CS) 2008-2011: Universal OpenMath Machine

All students are still members of the group unless mentioned otherwise.

3.8.2 DFKI

At DFKI, Till Mossakowski has formed a small subgroup of Prof. Bernd Krieg-Brückner's research group. Partly in collaboration with international partners, the following LATIN-related theses have been started or completed:

Mihai Codescu Ph.D., 2006-2012: Architectural Refinement in Hets

Martin Kühl Diploma, 2009-2010: Integrating Maude into Hets

Codruţa Gîrlea M.Sc., 2010-2011: An Extended Modal Logic Institution

Tzu-Keng Fu Ph.D., 2011-: Universal Logic and the Geography of Thought

Moreover, Adrían Riesco from the Universidad Complutense de Madrid has completed his PhD thesis *Depuración Declarativa y Verificación heterogenénea en Maude* while collaborating with Till Mossakowski and Mihai Codescu on LATIN-related topics.

3.9 List of Project-Related Publications

The publications whose references do not include a download URI have been added to the CD.

Articles

[1] Mihai Codescu, Till Mossakowski, Don Sannella, and Andrzej Tarlecki. „Refinement trees: calculi, tools and applications.“ In: (). Submitted.

[2] Mihai Codescu, Bruno Langenstein, Christian Maeder, and Till Mossakowski. „The VSE Refinement Method in Hets.“ In: *Electronic Communications of the EASST* (2012). Accepted for publication.

[3] F. Horozal and F. Rabe. „Representing Model Theory in a Type-Theoretical Logical Framework.“ In: *Theoretical Computer Science* 412.37 (2011), pp. 4919–4945.

[4] M. Iancu and F. Rabe. „Formalizing Foundations of Mathematics.“ In: *Mathematical Structures in Computer Science* 21.4 (2011), pp. 883–911.

[5] 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.“ In: *Procedia Computer Science* 4 (2011): *Special issue: Proceedings of the International Conference on Computational Science (ICCS).* Ed. by Mitsuhisa Sato, Satoshi Matsuoka, Peter M. Sloot, G. Dick van Albada, and Jack Dongarra. Finalist at the Executable Papers Challenge, pp. 598–607. DOI: 10.1016/j.procs.2011.04.063. URL: https://svn.mathweb.org/repos/planetary/doc/epc11/paper.pdf.

Conference Papers

[6] M. Codescu, F. Horozal, M. Kohlhase, T. Mossakowski, and F. Rabe. „A Proof Theoretic Interpretation of Model Theoretic Hiding.“ In: *Recent Trends in Algebraic Development Techniques*. Ed. by H. Kreowski and T. Mossakowski. Vol. 7137. Lecture Notes in Computer Science. Springer, 2011.

[7] M. Codescu, F. Horozal, M. Kohlhase, T. Mossakowski, and F. Rabe. „Project Abstract: Logic Atlas and Integrator (LATIN).“ In: *Intelligent Computer Mathematics*. Ed. by J. Davenport, W. Farmer, F. Rabe, and J. Urban. Vol. 6824. Lecture Notes in Computer Science. Springer, 2011, pp. 287–289.

[8] M. Codescu, F. Horozal, M. Kohlhase, T. Mossakowski, F. Rabe, and K. Sojakova. „Towards Logical Frameworks in the Heterogeneous Tool Set Hets.“ In: *Recent Trends in Algebraic Development Techniques*. Ed. by H. Kreowski and T. Mossakowski. Vol. 7137. Lecture Notes in Computer Science. Springer, 2011.

[9] Mihai Codescu. „Lambda Expressions in CASL Architectural Specifications.“ In: *Recent Trends in Algebraic Development Techniques*. Ed. by Hans-Jörg Kreowski Till Mossakowski. Lecture Notes in Computer Science. Springer, 2011.

[10] Mihai Codescu and Till Mossakowski. „Refinement trees: calculi, tools and applications.“ In: *Algebra and Coalgebra in Computer Science, CALCO'11*. Ed. by Bartek Klin Andrea Corradini. Vol. 6859. Lecture Notes in Computer Science. Springer, 2011, pp. 145–160.

[11] Mihai Codescu, Till Mossakowski, Adrían Riesco, and Christian Maeder. „Integrating Maude into Hets.“ In: *AMAST 2010*. Ed. by Mike Johnson and Dusko Pavlovic. Vol. 6486. Lecture Notes in Computer Science. Springer, 2010, pp. 60–75. URL: http://www.springerlink.com/content/978-3-642-17795-8#section=842927.

[12] F. Horozal, A. Iacob, C. Jucovschi, M. Kohlhase, and F. Rabe. „Combining Source, Content, Presentation, Narration, and Relational Representation.“ In: *Intelligent Computer Mathematics*. Ed. by J. Davenport, W. Farmer, F. Rabe, and J. Urban. Vol. 6824. Lecture Notes in Computer Science. Springer, 2011, pp. 211–226.

[13] M. Kohlhase, F. Rabe, and V. Zholudev. „Towards MKM in the Large: Modular Representation and Scalable Software Architecture.“ In: *Intelligent Computer Mathematics*. Ed. by S. Autexier, J. Calmet, D. Delahaye, P. Ion, L. Rideau, R. Rioboo, and A. Sexton. Vol. 6167. Lecture Notes in Computer Science. Springer, 2010, pp. 370–384.

[14] Christoph Lange, Michael Kohlhase, Catalin David, Deyan Ginev, Andrea Kohlhase, Bogdan Matican, Stefan Mirea, and Vyacheslav Zholudev. „The Planetary System: Executable Science, Technology, Engineering and Math Papers.“ In: *ESWC (Part II)*. 8th Extended Semantic Web Conference (ESWC) (Hersonissos, Crete, Greece, May 29–June 2, 2011). Ed. by Grigoris Antoniou, Marko Grobelnik, Elena Paslaru Bontas Simperl, Bijan Parsia, Dimitris Plexousakis, Pieter De Leenheer, and Jeff Z. Pan. Lecture Notes in Computer Science 6644. Heidelberg: Springer Verlag, 2011, pp. 471–475. ISBN: 978-3-642-21033-4. arXiv:1103.1482 [cs.DL].

[15] F. Rabe, M. Kohlhase, and C. Sacerdoti Coen. „A Foundational View on Integration Problems.“ In: *Intelligent Computer Mathematics*. Ed. by J. Davenport, W. Farmer, F. Rabe, and J. Urban. Vol. 6824. Lecture Notes in Computer Science. Springer, 2011, pp. 106–121.

[16] K. Sojakova and F. Rabe. „Translating Dependently-Typed Logic to First-Order Logic.“ In: *Recent Trends in Algebraic Development Techniques*. Ed. by A. Corradini and U. Montanari. Vol. 5486. Lecture Notes in Computer Science. Springer, 2009, pp. 326–341.

[17] V. Zholudev, M. Kohlhase, and F. Rabe. „A [insert XML Format] Database for [insert cool application].“ In: *XMLPrague 2010*. Proceedings of XMLPrague. XMLPrague.cz, 2010.

Workshop Papers

[18] S. Dumbrava, F. Horozal, and K. Sojakova. „A Case Study on Formalizing Algebra in a Module System.“ In: *Workshop on Modules and Libraries for Proof Assistants*. Ed. by F. Rabe and C. Schürmann. Vol. 429. ACM International Conference Proceeding Series. ACM, 2009, pp. 11–18.

[19] F. Horozal, M. Kohlhase, and F. Rabe. „Extending OpenMath with Sequences.“ In: *Intelligent Computer Mathematics, Work-in-Progress Proceedings*. Ed. by A. Asperti, J. Davenport, W. Farmer, F. Rabe, and J. Urban. Vol. UBLCS-2011-04. Technical Report, University of Bologna. University of Bologna, 2011, pp. 58–72.

[20] F. Horozal and F. Rabe. „Representing Model Theory in a Type-Theoretical Logical Framework.“ In: *Fourth Workshop on Logical and Semantic Frameworks, with Applications*. Ed. by M. Ayala-Rincón and F. Kamareddine. Vol. 256. Electronic Notes in Theoretical Computer Science. Elsevier, 2009, pp. 49–65.

[21] Corneliu C. Prodescu and Michael Kohlhase. „MathWebSearch 0.5 - Open Formula Search Engine.“ In: *Wissens- und Erfahrungsmanagement LWA (Lernen, Wissensentdeckung und Adaptivität) Conference Proceedings*. 2011. URL: https://svn.mathweb.org/repos/mws/doc/2011/newmws/main.pdf.

[22] F. Rabe. „Representing Isabelle in LF.“ In: *Logical Frameworks and Meta-languages: Theory and Practice*. Ed. by K. Crary and M. Miculan. Vol. 34. Electronic Proceedings in Theoretical Computer Science. Electronic Proceedings in Theoretical Computer Science, 2010, pp. 85–100.

[23] F. Rabe and C. Schürmann. „A Practical Module System for LF.“ In: *Proceedings of the Workshop on Logical Frameworks: Meta-Theory and Practice (LFMTP)*. Ed. by J. Cheney and A. Felty. Vol. LFMTP'09. ACM International Conference Proceeding Series. ACM Press, 2009, pp. 40–48.

Submitted

[24] Mihnea Iancu, Michael Kohlhase, Florian Rabe, and Josef Urban. „The Mizar Mathematical Library in OMDoc: Translation and Applications.“ submitted to JAR. 2011. URL: https://svn.omdoc.org/repos/latin/public/MizarOMDocAppl.pdf.

[25] Florian Rabe and Michael Kohlhase. „A Scalable Module System.“ Manuscript, submitted to Information & Computation. 2011. URL: http://kwarc.info/frabe/Research/mmt.pdf.

4 Signatures (Unterschriften)

24. 1. 2012

Date	Prof. Dr. Michael Kohlhase	Prof. Dr. Till Mossakowski

References

[1] S. Autexier, D. Hutter, B. Langenstein, H. Mantel, G. Rock, A. Schairer, W. Stephan, R. Vogt und A. Wolpers. „VSE: Formal methods meet industrial needs“. In: *International Journal on Software Tools for Technology Transfer, Special issue on Mechanized Theorem Proving for Technology* 3.1 (2000).

[2] Serge Autexier, Catalin David, Dominik Dietrich, Michael Kohlhase und Vyacheslav Zholudev. „Workflows for the Management of Change in Science, Technologies, Engineering and Mathematics“. In: *Intelligent Computer Mathematics*. Hrsg. von James Davenport, William Farmer, Florian Rabe und Josef Urban. LNAI 6824. Springer Verlag, 2011, S. 164–179. ISBN: 978-3-642-22672-4. URL: http://kwarc.info/kohlhase/papers/planetary-moc.pdf.

[3] H. Barendregt. „Lambda Calculi with Types“. In: *Handbook of Logic in Computer Science*. Hrsg. von S. Abramsky, D. Gabbay und T. Maibaum. Bd. 2. Oxford University Press, 1992.

[4] Michel Bidoit, Donald Sannella und Andrzej Tarlecki. „Architectural Specifications in CASL“. In: *Formal Aspects of Computing* 13 (2002), S. 252–273.

[5] M. Clavel, S. Eker, P. Lincoln und J. Meseguer. „Principles of Maude“. In: *Proceedings of the First International Workshop on Rewriting Logic*. Hrsg. von J. Meseguer. Bd. 4. 1996, S. 65–89.

[6] M. Codescu, F. Horozal, M. Kohlhase, T. Mossakowski und F. Rabe. „A Proof Theoretic Interpretation of Model Theoretic Hiding“. In: *Recent Trends in Algebraic Development Techniques*. Hrsg. von H. Kreowski und T. Mossakowski. Bd. 7137. Lecture Notes in Computer Science. Springer, 2011.

[7] M. Codescu, F. Horozal, M. Kohlhase, T. Mossakowski und F. Rabe. „Project Abstract: Logic Atlas and Integrator (LATIN)“. In: *Intelligent Computer Mathematics*. Hrsg. von J. Davenport, W. Farmer, F. Rabe und J. Urban. Bd. 6824. Lecture Notes in Computer Science. Springer, 2011, S. 287–289.

[8] M. Codescu, F. Horozal, M. Kohlhase, T. Mossakowski, F. Rabe und K. Sojakova. „Towards Logical Frameworks in the Heterogeneous Tool Set Hets“. In: *Recent Trends in Algebraic Development Techniques*. Hrsg. von H. Kreowski und T. Mossakowski. Bd. 7137. Lecture Notes in Computer Science. Springer, 2011.

[9] Mihai Codescu. „Lambda Expressions in CASL Architectural Specifications“. In: *Recent Trends in Algebraic Development Techniques*. Hrsg. von Hans-Jörg Kreowski Till Mossakowski. Lecture Notes in Computer Science. Springer, 2011.

[10] Mihai Codescu und Till Mossakowski. „Refinement trees: calculi, tools and applications“. In: *Algebra and Coalgebra in Computer Science, CALCO'11*. Hrsg. von Bartek Klin Andrea Corradini. Bd. 6859. Lecture Notes in Computer Science. Springer, 2011, S. 145–160.

[11] Mihai Codescu, Till Mossakowski, Adrían Riesco und Christian Maeder. „Integrating Maude into Hets“. In: *AMAST 2010*. Hrsg. von Mike Johnson und Dusko Pavlovic. Bd. 6486. Lecture Notes in Computer Science. Springer, 2010, S. 60–75. URL: http://www.springerlink.com/content/978-3-642-17795-8#section=842927.

[12] Mihai Codescu, Till Mossakowski, Don Sannella und Andrzej Tarlecki. „Refinement trees: calculi, tools and applications“. In: (). Submitted.

[13] Mihai Codescu, Bruno Langenstein, Christian Maeder und Till Mossakowski. „The VSE Refinement Method in Hets“. In: *Electronic Communications of the EASST* (2012). Accepted for publication.

[14] CoFI (The Common Framework Initiative). *CASL Reference Manual*. Bd. 2960. LNCS. Springer, 2004.

[15] W. Damm, H. Hungar und E.-R. Olderog. „Verification of cooperating traffic agents“. In: *International Journal of Control* 79.5 (Mai 2006), S. 395–421.

[16] Catalin David, Deyan Ginev, Michael Kohlhase, Bogdan Matican und Stefan Mirea. „A Framework for Modular Semantic Publishing with Separate Compilation and Dynamic Linking“. In: *1st Workshop on Semantic Publication (SePublica)*. (Hersonissos, Crete, Greece, 30. Mai 2011). Hrsg. von Alexander García Castro, Christoph Lange, Evan Sandhaus und Anita de Waard. CEUR Workshop Proceedings 721. Aachen, 2011. URL: https://svn.mathweb.org/repos/planetary/doc/sepublica11/paper.pdf.

[17] R. Diaconescu. *Institution-independent Model Theory*. Birkhäuser, 2008.

[18] S. Dumbrava, F. Horozal und K. Sojakova. „A Case Study on Formalizing Algebra in a Module System". In: *Workshop on Modules and Libraries for Proof Assistants*. Hrsg. von F. Rabe und C. Schürmann. Bd. 429. ACM International Conference Proceeding Series. ACM, 2009, S. 11–18.

[19] J. Goguen und R. Burstall. „Institutions: Abstract Model Theory for Specification and Programming". In: *Journal of the Association for Computing Machinery* 39(1) (1992), S. 95–146.

[20] J. Goguen und G. Rosu. „Composing Hidden Information Modules over Inclusive Institutions". In: *From Object-Orientation to Formal Methods, Essays in Memory of Ole-Johan Dahl*. Hrsg. von O. Owe, S. Krogdahl und T. Lyche. Springer, 2004, S. 96–123.

[21] R. Harper, F. Honsell und G. Plotkin. „A framework for defining logics". In: *Journal of the Association for Computing Machinery* 40.1 (1993), S. 143–184.

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

[23] John Harrison. „Floating-Point Verification". In: *J. UCS* 13.5 (2007), S. 629–638.

[24] Erik Hennum und Vyacheslav Zholudev. „Tag Libraries for XSLT and XQuery". In: *Proceedings of Balisage: The Markup Conference*. Bd. 7. Balisage Series on Markup Technologies. Mulberry Technologies, Inc., 2011. DOI: doi:10.4242/BalisageVol7.Hennum01. URL: http://www.balisage.net/Proceedings/vol7/cover.html.

[25] F. Horozal, M. Kohlhase und F. Rabe. „Extending OpenMath with Sequences". In: *Intelligent Computer Mathematics, Work-in-Progress Proceedings*. Hrsg. von A. Asperti, J. Davenport, W. Farmer, F. Rabe und J. Urban. Bd. UBLCS-2011-04. Technical Report, University of Bologna. University of Bologna, 2011, S. 58–72.

[26] F. Horozal und F. Rabe. „Representing Model Theory in a Type-Theoretical Logical Framework". In: *Fourth Workshop on Logical and Semantic Frameworks, with Applications*. Hrsg. von M. Ayala-Rincón und F. Kamareddine. Bd. 256. Electronic Notes in Theoretical Computer Science. Elsevier, 2009, S. 49–65.

[27] F. Horozal und F. Rabe. „Representing Model Theory in a Type-Theoretical Logical Framework". In: *Theoretical Computer Science* 412.37 (2011), S. 4919–4945.

[28] F. Horozal, A. Iacob, C. Jucovschi, M. Kohlhase und F. Rabe. „Combining Source, Content, Presentation, Narration, and Relational Representation". In: *Intelligent Computer Mathematics*. Hrsg. von J. Davenport, W. Farmer, F. Rabe und J. Urban. Bd. 6824. Lecture Notes in Computer Science. Springer, 2011, S. 211–226.

[29] M. Iancu und F. Rabe. „Formalizing Foundations of Mathematics". In: *Mathematical Structures in Computer Science* 21.4 (2011), S. 883–911.

[30] Mihnea Iancu, Michael Kohlhase und Florian Rabe. *Translating the Mizar Mathematical Library into OMDoc format*. KWARC Report. Jacobs University Bremen, 2011. URL: https://svn.omdoc.org/repos/latin/public/Mizar2OMDoc-Report.pdf.

[31] Mihnea Iancu, Michael Kohlhase, Florian Rabe und Josef Urban. „The Mizar Mathematical Library in OMDoc: Translation and Applications". submitted to JAR. 2011. URL: https://svn.omdoc.org/repos/latin/public/MizarOMDocAppl.pdf.

[32] Mihnea Iancu, Michael Kohlhase, Florian Rabe und Josef Urban. „The Mizar Mathematical Library in OMDoc: Translation and Applications". submitted to JAR. 2011. URL: https://svn.omdoc.org/repos/latin/public/MizarOMDocAppl.pdf.

[33] Andrea Kohlhase und Michael Kohlhase. „Spreadsheet Interaction with Frames: Exploring a Mathematical Practice". In: *MKM/Calculemus Proceedings*. Hrsg. von Jacques Carette, Lucas Dixon, Claudio Sacerdoti Coen und Stephen M. Watt. LNAI 5625. Springer Verlag, Juli 2009, S. 341–356. ISBN: 978-3-642-02613-3. URL: http://kwarc.info/kohlhase/papers/mkm09-framing.pdf.

[34] Andrea Kohlhase und Michael Kohlhase. „Towards a Flexible Notion of Document Context“. In: *Proceedings of the 29th annual ACM international conference on Design of communication (SIGDOC)*. 2011.

[35] M. Kohlhase. *OMDoc: An Open Markup Format for Mathematical Documents (Version 1.2)*. Lecture Notes in Artificial Intelligence 4180. Springer, 2006.

[36] M. Kohlhase, T. Mossakowski und F. Rabe. *The LATIN Project*. see https://trac.omdoc.org/LATIN/. 2009.

[37] M. Kohlhase, F. Rabe und V. Zholudev. „Towards MKM in the Large: Modular Representation and Scalable Software Architecture“. In: *Intelligent Computer Mathematics*. Hrsg. von S. Autexier, J. Calmet, D. Delahaye, P. Ion, L. Rideau, R. Rioboo und A. Sexton. Bd. 6167. Lecture Notes in Computer Science. Springer, 2010, S. 370–384.

[38] Michael Kohlhase. *OAF: FlexiForms*. 23. Juli 2010. URL: http://trac.kwarc.info/oaf/wiki/FlexiForms?version=5 (besucht am 11. 08. 2010).

[39] Michael Kohlhase. „Using LaTeX as a Semantic Markup Format“. In: *Mathematics in Computer Science* 2.2 (2008), S. 279–304. URL: https://svn.kwarc.info/repos/stex/doc/mcs08/stex.pdf.

[40] Michael Kohlhase und Ioan Şucan. „A Search Engine for Mathematical Formulae“. In: *Proceedings of Artificial Intelligence and Symbolic Computation, AISC'2006*. Hrsg. von Tetsuo Ida, Jacques Calmet und Dongming Wang. LNAI 4120. Springer Verlag, 2006, S. 241–253. URL: http://kwarc.info/kohlhase/papers/aisc06.pdf.

[41] Michael Kohlhase, Joe Corneli, Catalin David, Deyan Ginev, Constantin Jucovschi, Andrea Kohlhase, Christoph Lange, Bogdan Matican, Stefan Mirea und Vyacheslav Zholudev. „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)*. Hrsg. von Mitsuhisa Sato, Satoshi Matsuoka, Peter M. Sloot, G. Dick van Albada und Jack Dongarra. Finalist at the Executable Papers Challenge, S. 598–607. DOI: 10.1016/j.procs.2011.04.063. URL: https://svn.mathweb.org/repos/planetary/doc/epc11/paper.pdf.

[42] Oliver Kutz und Till Mossakowski. „A Modular Consistency Proof for Dolce“. In: *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence and the Twenty-Third Innovative Applications of Artificial Intelligence Conference*. AAAI Press; Menlo Park, CA, 2011, S. 227–234. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3754.

[43] Christoph Lange, Michael Kohlhase, Catalin David, Deyan Ginev, Andrea Kohlhase, Bogdan Matican, Stefan Mirea und Vyacheslav Zholudev. „The Planetary System: Executable Science, Technology, Engineering and Math Papers“. In: *ESWC (Part II)*. 8th Extended Semantic Web Conference (ESWC) (Hersonissos, Crete, Greece, 29. Mai–2. Juni 2011). Hrsg. von Grigoris Antoniou, Marko Grobelnik, Elena Paslaru Bontas Simperl, Bijan Parsia, Dimitris Plexousakis, Pieter De Leenheer und Jeff Z. Pan. Lecture Notes in Computer Science 6644. Heidelberg: Springer Verlag, 2011, S. 471–475. ISBN: 978-3-642-21033-4. arXiv:1103.1482 [cs.DL].

[44] Narciso Martí-Oliet und José Meseguer. „What is a logical system?“ In: Hrsg. von Dov M. Gabbay. New York, NY, USA: Oxford University Press, Inc., 1994. Kap. General logics and logical frameworks, S. 355–391. ISBN: 0-19-853859-6. URL: http://portal.acm.org/citation.cfm?id=208683.208700.

[45] Dominique Méry und Stephan Merz, Hrsg. *Integrated Formal Methods - 8th International Conference, IFM 2010, Nancy, France, October 11-14, 2010. Proceedings*. Bd. 6396. Lecture Notes in Computer Science. Springer, 2010. ISBN: 978-3-642-16264-0.

[46] T. Nipkow, L. Paulson und M. Wenzel. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*. Springer, 2002.

[47] L. Paulson. *Isabelle: A Generic Theorem Prover*. Bd. 828. Lecture Notes in Computer Science. Springer, 1994.

[48] F. Pfenning, C. Schürmann, M. Kohlhase, N. Shankar und S. Owre. *The Logosphere Project*. http://www.logosphere.org/. 2003.

[49] *Planetary Developer Forum*. URL: http://trac.mathweb.org/planetary/ (besucht am 08. 09. 2011).

[50] Corneliu C. Prodescu und Michael Kohlhase. „MathWebSearch 0.5 - Open Formula Search Engine". In: *Wissens- und Erfahrungsmanagement LWA (Lernen, Wissensentdeckung und Adaptivität) Conference Proceedings*. 2011. URL: https://svn.mathweb.org/repos/mws/doc/2011/newmws/main.pdf.

[51] F. Rabe. „A Logical Framework Combining Model and Proof Theory". see http://kwarc.info/frabe/Research/rabe_combining_09.pdf. 2010.

[52] F. Rabe. „Representing Isabelle in LF". In: *Logical Frameworks and Meta-languages: Theory and Practice*. Hrsg. von K. Crary und M. Miculan. Bd. 34. Electronic Proceedings in Theoretical Computer Science. Electronic Proceedings in Theoretical Computer Science, 2010, S. 85–100.

[53] F. Rabe und C. Schürmann. „A Practical Module System for LF". In: *Proceedings of the Workshop on Logical Frameworks: Meta-Theory and Practice (LFMTP)*. Hrsg. von J. Cheney und A. Felty. Bd. LFMTP'09. ACM International Conference Proceeding Series. ACM Press, 2009, S. 40–48.

[54] Florian Rabe. *The MMT Language and System*. 11. Okt. 2011. URL: https://svn.kwarc.info/repos/MMT.

[55] Florian Rabe und Michael Kohlhase. „A Scalable Module System". Manuscript, submitted to Information & Computation. 2011. URL: http://kwarc.info/frabe/Research/mmt.pdf.

[56] *Semantic Markup for LaTeX*. Project Homepage. URL: http://trac.kwarc.info/sTeX/ (besucht am 22. 02. 2011).

[57] Cesare Tinelli und Viorica Sofronie-Stokkermans, Hrsg. *Frontiers of Combining Systems, 8th International Symposium, FroCoS 2011, Saarbrücken, Germany, October 5-7, 2011. Proceedings*. Bd. 6989. Lecture Notes in Computer Science. Springer, 2011. ISBN: 978-3-642-24363-9.

[58] V. Zholudev, M. Kohlhase und F. Rabe. „A [insert XML Format] Database for [insert cool application]". In: *XMLPrague 2010*. Proceedings of XMLPrague. XMLPrague.cz, 2010.

[59] Vyacheslav Zholudev und Michael Kohlhase. „Scripting Documents with XQuery: Virtual Documents in TNTBase". In: *Proceedings of Balisage: The Markup Conference*. Bd. 5. Balisage Series on Markup Technologies. Mulberry Technologies, Inc., 2010. DOI: 10.4242/BalisageVol5.Zholudev01. URL: http://www.balisage.net/Proceedings/vol5/cover.html.

[60] Vyacheslav Zholudev und Michael Kohlhase. „TNTBase: a Versioned Storage for XML". In: *Proceedings of Balisage: The Markup Conference*. Bd. 3. Balisage Series on Markup Technologies. Mulberry Technologies, Inc., 2009. DOI: 10.4242/BalisageVol3.Zholudev01.

[61] Vyacheslav Zholudev, Michael Kohlhase und Florian Rabe. „A [insert XML Format] Database for [insert cool application]". In: *Proceedings of XML Prague 2010*. 2010, S. 317–339. URL: http://kwarc.info/vzholudev/pubs/XMLPrague.pdf.