Knowledge Management across Formal Libraries

Dennis Müller
Jacobs University Bremen

May 23, 2016

Abstract

In my Ph.D., I want to contribute to developing an open archive for formalizations, a common and open infrastructure for managing and sharing formalized mathematical knowledge such as theories, definitions, and proofs, based on a uniform foundation-independent representation format for libraries, integrate existing formal libraries into this archive and develop methods to efficiently transfer and share information between them.

Research Problem: In the last 10 years, the formalization of mathematical knowledge (and subsequent verification/automation of formal proofs) has – especially in conjunction with problems such as as Kepler's conjecture, the classification theorem for finite simple groups etc. – become of growing interest to mathematicians. By now, there is a vast plurality of formal/symbolic systems and corresponding libraries; almost all of which are, however, non-interoperable because they are based on differing, mutually incompatible foundations (e.g., set theory, higher-order logic, constructive type theory, etc.), library formats, and library structures, and much work is spent developing basic libraries for mathematics in each system.

Also, formalizations in current systems are usually based on the *homogeneous method*, which fixes one foundation with all primitive notions (e.g., types, axioms, and rules) and uses only conservative extensions (e.g., definitions, theorems) to model domain knowledge. For this purpose, most systems support complex conservative extension principles, such as type definitions in the HOL systems [HOL], provably terminating functions in Coq [Tea] or Isabelle/HOL [NPW02], or provably well-defined indirect definitions in Mizar [Miz].

The combination of fixed foundation and homogeneous method means that a lot of – expensive – formalization work is needed just to build the setting of interest (e.g., the real numbers) as a conservative extension of the fixed foundation. However, the resulting formalizations are actually less valuable: It becomes virtually impossible to move them between foundations. Therefore, almost all current systems are mutually incompatible, with only a few ad hoc translations between them (e.g., [KW10; KS10]).

On the other hand, the *heterogeneous method*, going back to the works by Bourbaki [Bou64], focuses on defining theories that may introduce new primitive notions, and considers truth relative to a theory. The heterogeneous method optimizes reusability by stating every result in the weakest possible theory and using *theory morphisms* to move results between theories in a truth-preserving way. This is often called the little theories approach [FGT92]. Similarly, scientific practice prefers the heterogeneous method. For example, while all mathematics can be reduced to first principles (e.g., using the homogeneous method based on axiomatic set theory), it is usually carried out in highly abstracted settings that hide the foundation. For example, the category of categories is used routinely without focusing on its foundational subtleties. Correspondingly, there is an inconvenient discrepancy between the currently existing formal libraries and the way (informal) mathematics is usually done in practice.

Research Objectives: We want to tackle these interoperability and plurality problems by developing an open archive for formalizations (**OAF**), a common and open infrastructure for managing and sharing formalized mathematical knowledge such as theories, definitions, and proofs, designed to be scalable with respect to both the size of the knowledge base and the diversity of logical foundations.

Theoretically, the main prerequisite has been established in the LATIN project [KMR09]. The LATIN logical framework [Rab13; Cod+11] integrates institutional representations of model theory and type theoretical representations of proof theory and thus permits combining the benefits of both worlds. A paradigmatic example was published as [HR11]. However, whereas LATIN provides a logical framework, there still remains the problem of integrating the existing formal libraries.

There are two facets of library integration. Firstly, one can *refactor a single library* to increase reuse through modularity, sharing, and inheritance. Secondly, one can *connect or merge two libraries* from different systems. This requires translating the libraries into a common language (namely MMT) and then identifying and eliminating overlap between the two libraries.

My initial focus will be on integrating the specification language **PVS** [ORS92], with others to follow. PVS is a specification language integrated with support tools and a theorem prover under active development at the Computer Science Laboratory of SRI International.

On this basis, I then want to tackle the problem of refactoring and merging libraries.

Methodology: The aim is to integrate the syntax, underlying foundation and (ultimately) available libraries of different theorem provers into **MMT** [RK13; HKR12; KRSC11], a uniform foundation-independent representation format for libraries, which allows formalizing the logical foundations alongside the libraries and thus acts as framework for aligning libraries.

Integrating each such library entails four things:

1. Writing an MMT-Plugin, that allows for importing the existing archives into the MMT API,
2. writing an MMT theory, that provides the underlying foundational theory,
3. (potentially) implementing desirable features on the logical framework level (subtyping features, recursive definition principles etc.) to provide syntactical constructors for the specific peculiarities of the theorem prover under consideration and
4. (potentially) adapting and improving the MMT language and API in the process.

In the case of PVS, the particular work will be in translating the inductive/coinductive types, record types and the sophisticated subtyping mechanism that PVS provides into the MMT system and to match the conceptually different module systems of both languages (PVS uses theories and namespaces quite differently than MMT).

Furthermore, I investigate methods for refactoring and integrating/connecting the various theorem prover libraries. Useful notions in that regard are *alignments* [Kal+16] between semantically equivalent symbols across different libraries and foundations, *interface theories* [KRSC11] that abstract from the specifics of a given foundation and *theory intersections* [MK15] for generating interface theories.

Preliminary results: Apart from the preliminary results due to others (mentioned in the previous paragraphs), personal results include:

- A specification of the theorem prover TPS and a translation of its library into MMT (in progress),
- extensions of the logical framework LF to allow for specifying more complex foundational theories with (among others) judgmental and predicate subtypes, an infinite type hierarchy and record types, including a logical framework based on homotopy type theory as a case study [Mmta],
- an almost complete specification of PVS in MMT using the aforementioned LF extensions [Mmtc],
- a corresponding translation of PVS's Prelude and NASA libraries into the MMT language [Mmtb],
- a survey of different alignment types and a simple language to specify different kinds of alignments [Kal+16],
- a first implementation of an *expression translation* machinery in MMT, that uses alignments and various theory morphisms to translate arbitrary expressions between different formal libraries,

- an implementation of an algorithm for finding alignments between libraries,
- various improvements on the MMT API (parsing, type checking, new language features, interfaces to external databases and applications).

Future work: I am currently working on survey papers on subtyping principles and type hierarchies – these are intended to serve as a starting point for implementing the corresponding features in MMT as generic as possible.

The PVS specification and translation need to be improved with respect to record types, (co-)inductive definition principles and more obscure language features, such as update expressions, to faithfully capture the actual behaviour of the PVS system. Also, I want to additionally import PVS's proof files, to allow for e.g. exporting and translating proof sketches. The latter will require a generic specification of various proof tactics.

The expression translation and alignment finding algorithms are in an early stage and need to be improved, evaluated as to their usefulness and extended to allow for more complex translations. Furthermore, I want to investigate methods for generating interface theories from alignments and theory morphisms.

Last but not least, more formal systems will be imported into MMT, providing additional challenges for all the presented research objectives.

References

[Bou64] N. Bourbaki. "Univers". In: *Séminaire de Géométrie Algébrique du Bois Marie - Théorie des topos et cohomologie étale des schémas*. Springer, 1964, pp. 185–217.

[Cod+11] M. Codescu et al. "Towards Logical Frameworks in the Heterogeneous Tool Set Hets". In: *Recent Trends in Algebraic Development Techniques*. Ed. by H. Kreowski and T. Mossakowski. LNCS 7137. Springer, 2011.

[FGT92] W. Farmer, J. Guttman, and F. Thayer. "Little Theories". In: *Conference on Automated Deduction*. Ed. by D. Kapur. 1992, pp. 467–581.

[HKR12] Fulya Horozal, Michael Kohlhase, and Florian Rabe. "Extending MKM Formats at the Statement Level". In: *Intelligent Computer Mathematics*. Ed. by Johan Jeuring et al. LNAI 7362. Berlin and Heidelberg: Springer Verlag, 2012, pp. 65–80. URL: http://kwarc.info/kohlhase/papers/mkm12-p2s.pdf.

[HR11] F. Horozal and F. Rabe. "Representing Model Theory in a Type-Theoretical Logical Framework". In: *Theoretical Computer Science* 412.37 (2011), pp. 4919–4945.

[Kal+16] Cezary Kaliszyk et al. "A Standard for Aligning Mathematical Concepts". Submitted to CICM2016. 2016. URL: http://kwarc.info/kohlhase/submit/alignments16.pdf.

[KMR09] M. Kohlhase, T. Mossakowski, and F. Rabe. *The LATIN Project*. see https://trac.omdoc.org/LATIN/. 2009.

[KRSC11] Michael Kohlhase, Florian Rabe, and Claudio Sacerdoti Coen. "A Foundational View on Integration Problems". In: *Intelligent Computer Mathematics*. Ed. by James Davenport et al. LNAI 6824. Springer Verlag, 2011, pp. 107–122. URL: http://kwarc.info/kohlhase/papers/cicm11-integration.pdf.

[KS10] A. Krauss and A. Schropp. "A Mechanized Translation from Higher-Order Logic to Set Theory". In: *Interactive Theorem Proving*. Ed. by M. Kaufmann and L. Paulson. Springer, 2010, pp. 323–338.

[KW10] C. Keller and B. Werner. "Importing HOL Light into Coq". In: *Interactive Theorem Proving*. Ed. by M. Kaufmann and L. Paulson. Springer, 2010, pp. 307–322.

[Miz] *Mizar*. http://www.mizar.org. 1973–2006. URL: http://www.mizar.org.

[MK15] Dennis Müller and Michael Kohlhase. "Understanding Mathematical Theory Formation via Theory Intersections in MMT". 2015. URL: http://cicm-conference.org/2015/fm4m/FMM_2015_paper_2.pdf.

[Mmta] *MathHub MMT/LFX Git Repository*. URL: http://gl.mathhub.info/MMT/LFX (visited on 05/15/2015).

[Mmtb] *MathHub PVS/NASA Git Repository*. URL: http://gl.mathhub.info/PVS/NASA (visited on 05/15/2015).

[Mmtc] *MathHub PVS/Prelude Git Repository*. URL: http://gl.mathhub.info/PVS/Prelude (visited on 05/15/2015).

[NPW02] T. Nipkow, L. Paulson, and M. Wenzel. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*. Springer, 2002.

[ORS92] S. Owre, J. Rushby, and N. Shankar. "PVS: A Prototype Verification System". In: *11th International Conference on Automated Deduction (CADE)*. Ed. by D. Kapur. Springer, 1992, pp. 748–752.

[Rab13] F. Rabe. "A Logical Framework Combining Model and Proof Theory". In: *Mathematical Structures in Computer Science* 23.5 (2013), pp. 945–1001.

[RK13] Florian Rabe and Michael Kohlhase. "A Scalable Module System". In: *Information & Computation* 0.230 (2013), pp. 1–54. URL: http://kwarc.info/frabe/Research/mmt.pdf.

[Tea] Coq Development Team. *The Coq Proof Assistant: Reference Manual*. INRIA. URL: https://coq.inria.fr/refman/.

[HOL] HOL4 development team. http://hol.sourceforge.net/.