Project Abstract: Logic Atlas and Integrator (LATIN) *

Mihai Codescu[1], Fulya Horozal[2], Michael Kohlhase[2], Till Mossakowski[1], and Florian Rabe[2]

[1] Safe and Secure Cognitive Systems, German Research Centre for Artificial Intelligence (DFKI), Bremen, Germany
[2] Computer Science, Jacobs University Bremen, Germany
http://latin.omdoc.org

LATIN aims at developing methods, techniques, and tools for interfacing logics and related formal systems. These systems are at the core of mathematics and computer science and are implemented in systems like (semi-)automated theorem provers, model checkers, computer algebra systems, constraint solvers, or concept classifiers. Unfortunately, these systems have differing domains of applications, foundational assumptions, and input languages, which makes them non-interoperable and difficult to compare and evaluate in practice.

The LATIN project develops a foundationally unconstrained framework for the representation of logics and translations between them [9, 1]. The LATIN framework (i) subsumes existing proof theoretical frameworks such as LF and model theoretical frameworks such as institutions [3] and (ii) supplants them with a uniform knowledge representation language based on OMDoc. Special attention is paid to generality, modularity, scalability, extensibility, and interoperability.

$$\begin{tikzcd}
& & L^{Pf} \arrow[dd, "L^{sound}"] \\
Base \arrow[r, hook, "L^{truth}"] & L^{Syn} \arrow[ur, hook, "L^{pf}"] \arrow[dr, "L^{mod}"'] & \\
& & L^{Mod} \arrow[d, bend right, "M"'] \\
& & \mathcal{F} \arrow[u, hook, bend right]
\end{tikzcd}$$

Logics are represented as theories and translations as theory morphisms. Logic representations formalize the syntax, proof theory, and model theory of a logic within the LATIN framework. The representations of the model theory are parametric in the foundation of mathematics, which is represented as a theory itself; then individual models are represented as theory morphisms into the foundation. This can be represented in a diagram such as the one above, where the syntax of a logic L is represented as a theory L^{Syn}, which is then extended with the representation of proof rules to represent the proof theory as L^{pf}. Moreover, the model theory of the logic can be represented as a theory L^{Mod}, based on the representation of a foundation $\mathcal{F}$ which is included the model theory; the models are represented by the arrow M. The *Base* theory represents the type of formulas and the notion of truth for them. Moreover, we can represent soundness proofs as a morphism

* The LATIN project is supported by the Deutsche Forschungsgemeinschaft (DFG) within grant KO 2428/9-1.

L^{sound} from the proof theory to the model theory of L. Similarly, logic translations formalize the translations of syntax, proof theory, and model theory. This "logics-as-theories" approach makes system behaviors as well as their represented knowledge interoperable and thus comparable at multiple levels.

The LATIN framework has been implemented generically within the Heterogeneous Tool Set Hets [7] and instantiated with the logical frameworks LF, Isabelle, and Maude. Hets is a general institution-based framework for integration of formal methods and heterogeneous specification and proof management. While Hets implements a large number of logics and translations, their semantics and correctness had previously been determined only by model theoretic arguments. Within the LATIN project, Hets has been extended to support adding logics semi-automatically using a logic specification in one of the supported logical frameworks. This brings the advantage that the logics of Hets are represented fully formally and verified mechanically, and that new logics can be added dynamically.

To evaluate the developed framework and provide a service to the community, the project builds an atlas of logics used in automated reasoning, mathematics, and software engineering. The concrete logic representations span over 1000 theories and morphisms and can be found at the project web site, they include (i) *Type theory*, including a modular development of the lambda cube, Martin-Löf Type Theory, and Isabelle, (ii) *Logics*, including first-order, higher-order, modal, and description logics, (iii) *Set theory* including ZFC and the Mizar variant of Tarski-Grothendieck set theory. The atlas also includes a growing number of logic translations including, e.g., the relativization translations from modal, description, and sorted first-order logics to unsorted first-order logic, the interpretation of type theory in set theory, the negative translation from classical to intuitionistic logic, and the translation from first to higher-order logic. Elaborate case studies were documented in [4, 5, 10]. The LATIN atlas is extensible, and new logics can be added easily — including the reuse of already formalized logic features — and related to the existing logics via translations.

To make the logic atlas scalable, we base it on the knowledge representation language MMT [11]. MMT refines the markup language for structured theories that is part of OMDoc and provides a formal semantics for it. Moreover, MMT comes with a scalable infrastructure [6] centered around a flexible and foundation-independent API.

In the authoring work flow of LATIN, representations are written in Twelf [8] using our module system for it [12]. Twelf converts the content into OMDoc/MMT, which indexes and stores it in the SVN+XML database TNTBase [13]. In the application work flow, these OMDoc/MMT documents are imported into Hets. In the presentation work flow, the MMT web server uses XQueries to retrieve LATIN content and user-defined notations from TNTBase, which are used to render the content as JOBAD-enabled [2] XHTML+MathML. All stages of this pipeline are semantics-aware so that, for example, the web server can offer interactive dynamic services such as definition lookup or toggling the display of inferred types.

The browsable version of the atlas is available at the project web site. When browsing it, keep in mind that the logical framework, the formalizations in it, and the whole infrastructure processing them are ongoing work and thus subject to both constant improvement and temporary failures.

References

1. M. Codescu, F. Horozal, M. Kohlhase, T. Mossakowski, F. Rabe, and K. Sojakova. Towards Logical Frameworks in the Heterogeneous Tool Set Hets. In *Workshop on Abstract Development Techniques*, Lecture Notes in Computer Science. Springer, 2011. To appear.
2. J. Gičeva, C. Lange, and F. Rabe. Integrating Web Services into Active Mathematical Documents. In J. Carette and L. Dixon and C. Sacerdoti Coen and S. Watt, editor, *Intelligent Computer Mathematics*, volume 5625 of *Lecture Notes in Computer Science*, pages 279–293. Springer, 2009.
3. J. Goguen and R. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39(1):95–146, 1992.
4. F. Horozal and F. Rabe. Representing Model Theory in a Type-Theoretical Logical Framework. *Theoretical Computer Science*, 2011. To appear, see `http://kwarc.info/frabe/Research/HR_folsound_10.pdf`.
5. M. Iancu and F. Rabe. Formalizing Foundations of Mathematics. *Mathematical Structures in Computer Science*, 2011. To appear, see `http://kwarc.info/frabe/Research/IR_foundations_10.pdf`.
6. M. Kohlhase, F. Rabe, and V. Zholudev. Towards MKM in the Large: Modular Representation and Scalable Software Architecture. In S. Autexier, J. Calmet, D. Delahaye, P. Ion, L. Rideau, R. Rioboo, and A. Sexton, editors, *Intelligent Computer Mathematics*, volume 6167 of *Lecture Notes in Computer Science*, pages 370–384. Springer, 2010.
7. T. Mossakowski, C. Maeder, and K. Lüttich. The Heterogeneous Tool Set. In O. Grumberg and M. Huth, editor, *TACAS 2007*, volume 4424 of *Lecture Notes in Computer Science*, pages 519–522, 2007.
8. F. Pfenning and C. Schürmann. System description: Twelf - a meta-logical framework for deductive systems. *Lecture Notes in Computer Science*, 1632:202–206, 1999.
9. F. Rabe. A Logical Framework Combining Model and Proof Theory. Submitted to Mathematical Structures in Computer Science, see `http://kwarc.info/frabe/Research/rabe_combining_09.pdf`, 2010.
10. F. Rabe. Representing Isabelle in LF. In K. Crary and M. Miculan, editors, *Logical Frameworks and Meta-languages: Theory and Practice*, volume 34 of *EPTCS*, pages 85–100, 2010.
11. F. Rabe and M. Kohlhase. A Scalable Module System. Under review, see `http://arxiv.org/abs/1105.0548`, 2011.
12. F. Rabe and C. Schürmann. A Practical Module System for LF. In J. Cheney and A. Felty, editors, *Proceedings of the Workshop on Logical Frameworks: Meta-Theory and Practice (LFMTP)*, pages 40–48. ACM Press, 2009.
13. V. Zholudev and M. Kohlhase. TNTBase: a Versioned Storage for XML. In *Proceedings of Balisage: The Markup Conference 2009*, volume 3 of *Balisage Series on Markup Technologies*. Mulberry Technologies, Inc., 2009.