Verification of Business Rules Programs

Bruno Berstel – Da Silva

Dissertation
zur Erlangung des Doktorgrades
der Technischen Fakultät
der Albert-Ludwigs-Universität Freiburg

July 11, 2012

Bruno Berstel – Da Silva
Verification of Business Rules Programs

Version 1.0.1
Built on July 12, 2012

Dekan:	Prof. Dr. Bernd Becker
Erstgutachter:	Prof. Dr. Andreas Podelski
Zweitgutachter:	Prof. Dr. Georg Lausen
Vorsitz:	Prof. Dr. Bernhard Nebel
Beisitz:	Prof. Dr. Peter Thiemann

Summary

The technical contribution of the thesis is to present—to the best of our knowledge, for the first time—an approach to the formal verification of business rules programs. We propose a verification method for proving correctness properties for a business rules program in a compositional way. The approach enables rule authors and tool developers to understand, express formally, and prove, properties of the execution behavior of business rules programs. The conceptual contribution of this thesis is to present the enabling framework for treating *business rules* as a topic of scientific investigation in semantics and program verification.

Multitier architectures increasingly use business rules to encode the application tier (the so-called business logic). The authoring and the execution of business rules programs is supported by a Business Rules Management System (BRMS). A business rules program consists in a set of mutually independent rules, that is, conditional update statements authored in a modular, case-by-case approach. A business rules program is declarative in that it does not specify the control flow; the set of rules are executed on a set of objects, exhaustively for each rule and each object, in any order.

Until now, the emphasis in research has been on optimizing the efficient execution of business rules programs by a BRMS. A variety of compilation and execution schemes have been developed, including the well-known Rete algorithm. The verification of business rules programs has been neglected as a topic of scientific research. The need for correctness is, however, no less obvious for business rules programs than it is for safety-critical systems, even though the risks at stake are economic and usually not life-threatening.

The thesis is structured in three main parts.

In the first part, we present a *formal definition of the execution behavior* of business rules programs. Previous descriptions of business rules execution depended on the intrinsics of a specific compilation scheme. The very first issue in the formalization task is the diversity of compilation and execution schemes used in existing BRMS. We have designed a general, yet formally simple, framework that enables us to describe the execution behavior of business rules programs and to single out the main differences between the various execution schemes. The formalization of the execution behavior of business rules programs allowed us to observe that the apparent simplicity of business rules is only superficial. Indeed, the interplay between executions of one or several rules on one or several, possibly shared, objects (selected nondeterministically from a finite, but unbounded set) can become extremely complex, even for small examples.

An execution is formally a sequence of states. To account for the unboundedness of the set of objects in each state, we model a state as a first-order logic structure. We account for the diversity of execution schemes, including recent alternatives to the Rete algorithm, by introducing concepts that allow us to distinguish between the *applicability* and the *eligibility* of a rule.

In the second part of the thesis, we introduce *correctness specifications* for business rules programs. Previously, the only way to assess the correctness of a business rules program was to examine each of its rules and its possible behaviors when applied to various objects. The expected global effect of a program had to be expressed in natural language, which could result in misunderstandings

between authors of programs and their users. The difficulty in formally defining the correctness of business rules programs stems from the gap between the local behavior of the single application of a rule and the global effect of a whole business rules program on a set of objects. The local behavior involves only the object on which the rule is applied, whereas the global behavior involves the whole, finite but unbounded, set of objects on which the program is run. We define the meaning of a Hoare triple for a program and global assertions as a conservative extension of the meaning of a Hoare triple for a single rule application and local assertions. We thus obtain correctness specifications that follow the modular and declarative nature of a business rules program.

In the third and final part of the thesis, we propose a *compositional verification method* for business rules programs. The challenge of compositionality stems from the possibility of interferences between rule applications during the execution of a program. Proving a correctness property of a business rules program cannot simply follow the decomposition of the program into its syntactic constituents, i.e., the rules. Borrowing the intuition of the Owicki-Gries method for parallel programs, we present a proof system that features an extended notion of compositionality, suitable for business rules programs. By our proof system, a global Hoare triple for a program can be derived from local Hoare triples for its individual rules. We show that the proof system is sound and relatively complete (we use relative completeness in the same sense as for Hoare logic). We derive proof rules for important classes of business rules programs and assertions, as special cases of the general proof rule. We use several examples to illustrate the practical application of the general proof rule and its specializations.

In a non-technical appendix to the thesis, we demonstrate the practical potential of our formal approach in the context of an existing commercial Business Rules Management System. This BRMS comes with a lightweight analysis engine (named "Rule Static Analysis"). We show that its various analysis and verification features can be given a solid foundation thanks to the approach to the formal verification of business rules programs presented in this thesis.

Acknowledgments

I carried out this thesis as a remote-working student, first of the Max-Planck Institut für Informatik of Saarbrücken, then of the Institut für Informatik of the Albert-Ludwigs-Universität in Freiburg. At the same time I was a software engineer at ILOG, and then at IBM after this company acquired ILOG. My thanks go to Nitsan Seniak for proposing me the subject of rule program verification within ILOG; to Hassan Aït-Kaci for establishing the connection with Andreas Podelski; and to Andreas Podelski, my supervisor, for offering me to reify our thoughts as a PhD thesis, and for his support and guidance throughout the years. It is also an honor for me to count Prof. Lausen as the Zweitgutachter, as well as Prof. Nebel and Prof. Thiemann as the members of my defense committee.

Achieving the present work in an industrial environment would not have been possible without the initial agreement of Jean-François Abramatic, and overall the creativity in management demonstrated by Nitsan Seniak and Antoine Melki. More generally, I would like to thank the members of the "BEAR R&D Governance" team within ILOG for their patience.

Substantial advances in this work have been achieved during a six months period spent in the Modélisation et Vérification team of the LIAFA laboratory in Paris Diderot university. I am most grateful to Ahmed Bouajjani for welcoming me. This stay was a major step for me, thanks to the friendly atmosphere and stimulating discussions. My thoughts go to Mihaela Sighireanu, Tayssir Touili, Cezara Drăgoi, Constantin Enea, Mathilde Bouvel, and Ahmed Rezine, as well as the whole "Vérif" team.

Within ILOG I have also benefited from enriching conversations and exchanges. I was nurtured by the outcomes of the RVS project, which included Hassan Aït-Kaci, Ulrich Junker, Michel Leconte, and Andreas Podelski; by the proofreading and most relevant comments of Hassan, Michel, Philippe Bonnard, and Hélène Kencker; and by the fruitful discussions with Hugues Citeau, Amina Chniti, Aurélie Baton, Marie Girard, and Philippe Laborie. Very special thanks are deserved by Michel for his continuous support, sustained stimulation, and for passing down the demand for accuracy inherited from my father.

During the long process that this work has represented for me, the right words were found on several occasions by Ulrich Junker and Claire David to help me find the required energy.

Last, but far from least, I am specially grateful to my wife and children for their support and their care.

Contents