Using Domain Specific Languages to Support Verification in the Railway Domain

Phillip James, Arnold Beckmann, and Markus Roggenbach

Swansea University, UK

Abstract. We explore the support of automatic verification via careful design of a domain specific language (DSL) in the context of algebraic specification. Formally a DSL is a loose specification the logical closure of which we regard as implicitly encoded "domain knowledge". We systematically exploit this "domain knowledge" for automatic verification. We illustrate these ideas within the Railway Domain using the algebraic specification language Casl and an existing DSL, designed by Bjørner, for modelling railways. Empirical evidence to the benefit of our approach is given in the form of the successful automatic verification of four railway track plans of real world complexity.

1 Introduction

For many years, verification based on techniques such as model checking or interactive theorem proving has been successful in various industrial case studies, e.g., see [10,4,6]. However, the use of formal methods within industry is still limited as it often requires verification experts. Domain specific languages [3] aim to abstract away technical details from the user. Classically, DSLs allow non-experts to create programs or specifications. In the context of *programming*, additional motivation for using DSLs includes *improved tool support*, improved ease of use, and increased productivity. Here we demonstrate, for algebraic *specification*, an approach where DSLs within the railway domain aid *verification*.

We suggest the following approach: Given a DSL for a particular class of systems and a set of decidable properties one is interested in, the DSL can be systematically extended to allow for automatic verification. We claim that the principles underlying this extension are universal, i.e., can be applied whenever one designs or adapts a DSL for verification. The overall aim of our approach is to develop a "push button" verification process for critical systems.

To illustrate this approach, we take an established DSL from Bjørner [1] and formalize it in the algebraic specification language Casl [9]. This allows connections with modern theorem proving technology via the Heterogeneous Toolset (HeTS) [8]. We then extend the DSL for automatic proof support. Finally, we give strong empirical evidence that our approach works by modelling and verifying four track plans provided by our industrial partner Invensys Rail.

Concretely, we demonstrate that we can exploit features of Bjørner's DSL to allow automatic verification of safety properties, e.g., routes that share railway components can not be open at the same time. To gain these results, we

A. Biere, A. Nahir, and T. Vos (Eds.): HVC 2012, LNCS 7857, pp. 274–275, 2013.

show that Bjørner's DSL (1) contains inherent structure allowing property specific abstraction and lifting of domain models and (2) is rich enough to prove suitable domain specific lemmas over such property specific abstractions. This demonstrates that domain specific languages can be designed to support automatic verification. To the best of our knowledge, we are the first to consider and formulate such a methodology for designing DSLs aimed at verification [7]. The underlying general principles we present include domain specific abstraction, domain specific property lifting and systematic property support. The result of this work is a step towards a platform for creating domain specific languages with effective automatic verification support for domain engineers.

1.1 Related Work

Various formal methods have been applied to railway verification. These include approaches using process algebraic modelling and verification in CSP [10], algebraic specification with ASF+SDF [4] and model-oriented specification using the B method, where, for example in [2] several lines of the Paris Metro system were verified. Finally, of close relevance to this work is the development environment for verification of railway control systems created by Haxthausen and Peleska [5]. This environment includes a DSL allowing modelling of control systems, and an automatic translation from models described in this DSL to executable control programs. At each level of production, various safety checking steps are taken.

References

1. Bjørner, D.: Dynamics of Railway Nets: On an Interface between Automatic Control and Software Engineering. In: CTS 2003 (2003)
2. Boulanger, J., Gallardo, M.: Validation and verification of METEOR safety software. In: Allen, J., Hill, R.J., Brebbia, C.A., Sciutto, G., Sone, S. (eds.) Computers in Railways VII, vol. 7, pp. 189–200. WIT Press (2000)
3. Fowler, M.: Domain Specific Languages. Addison-Wesley (2010)
4. Groote, J.F., van Vlijmen, S., Koorn, J.: The Safety Guaranteeing System at Station Hoorn-Kersenboogerd. Technical report. Utrecht University (1995)
5. Haxthausen, A., Peleska, J.: A domain-oriented, model-based approach for construction and verification of railway control systems. In: Jones, C.B., Liu, Z., Woodcock, J. (eds.) Formal Methods and Hybrid Real-Time Systems. LNCS, vol. 4700, pp. 320–348. Springer, Heidelberg (2007)
6. James, P., Roggenbach, M.: Automatically verifying railway interlockings using SAT-based model checking. In: Bendisposto, J., Leuschel, M., Roggenbach, M. (eds.) AVoCS 2010, vol. 35. ECEASST (2010)
7. James, P., Roggenbach, M.: Designing domain specific languages for verification: First steps. In: Hofner, P., McIver, A., Struth, G. (eds.) ATE 2011, vol. 760. CEUR (2011)
8. Mossakowski, T., Maeder, C., Lüttich, K.: The Heterogeneous Tool Set, Hets. In: Grumberg, O., Huth, M. (eds.) TACAS 2007. LNCS, vol. 4424, pp. 519–522. Springer, Heidelberg (2007)
9. Mosses, P.D. (ed.): Casl Reference Manual. LNCS, vol. 2960. Springer, Heidelberg (2004)
10. Winter, K.: Model checking railway interlocking systems. Australian Computer Science Communications 24, 303–310 (2002)