Loose semantics in the verification of communicating systems

Holger Schlingloff
Fraunhofer FIRST
Kekuléstr.7
D-12489 Berlin
holger.schlingloff@first.fraunhofer.de

Satish Mishra
Humboldt-Universität zu Berlin
Rudower Chaussee 25
D-12489 Berlin
mishra@informatik.hu-berlin.de

The specification language CSP-CASL combines algebraic and process algebraic formalisms for the description of reactive systems with structured data [Rog 03]. We are using this formalism to formally specify and verify a Swiss banking system and its communication protocol. The EP2 banking system describes the actors involved in an electronic payment and the interfaces between them [EP2]. The so-called EP2 terminal is the main customer interface for initiating the transactions; it can connect to different authorisation servers with a customized security protocol. In [GRS 05] we describe some aspects of an EP2 terminal, where the dynamic behaviour is modelled in CSP and the data on the channels is described in the common algebraic specification language CASL [ABK+02]. One particularity of this formalism is that CASL has a loose semantics for the definition of data types [Mos 04]. Intuitively, this means that there are more admissible interpretations of a specification than in the initial semantics; there are no further constraints on the models of a specification. This facilitates the use of parameterised data types and refinement relations between specifications. However, it brings about some unexpected phenomena.

In our project, we used the formal specification for transformational verification. We structured the specification into different layers of abstraction, where each lower layer is a refinement of the one above. Thus, refinement supports the incremental development of a specification into an implementation. There are two different aspects to specification refinement in CSP-CASL: data refinement and process refinement. Data refinement amounts to the augmentation of the axioms by additional requirements and reduces the class of possible models. Process refinement eliminates nondeterministic choices and amounts to a reduction of model classes in the denotational semantics of CSP. There are different possibilities for the definition of this notion; usually each refinement step transforms the specification from a more abstract level to a more concrete one. Since CASL has loose semantics, these refinement steps can be verified interactively. Automated tool support is under development [IR 05].

Refinement between specifications is closely coupled to conformance between specification and implementation. If a specification S' refines S, and I conforms to S', then I must conform also to S. Similarly, supposed T' is a test sequence which passes at the specification level S', then T' must pass also on the next, more abstract level of specification S. Here, an implementation conforms to a specification, if all tests which can be performed on both yield the same result, pass or fail (various other testing preorders have been defined in the literature, see [BT 01]). This property allows us to re-use test cases which have been developed from a more abstract specification for a more concrete level. We manually developed test sequences for an actual implementation of an EP2 terminal, and compared them to tests which can be derived from the specification. We are developing an automatic test generation algorithm similar to the one for LOTOS [GJ 99]. Reaching a test verdict requires to check whether a communication is consistent with the set of axioms in the algebraic data type. Here, loose semantics makes a significant difference since only those equalities are valid which can be derived from the axioms.

[ABK+02] E. Astesiano, M. Bidoit, H. Kirchner, B. Krieg-Brückner, P. Mosses, D. Sannella, and A. Tarlecki: CASL: the common algebraic specification language. TCS 286(2):153-196, 2002.
[BM 03] M. Bidoit, P. Mosses: CASL User Manual. Springer LNCS 2900, 2003.
[BT 01] E. Brinksma, J. Tretmans: Testing transition systems: an annotated bibliography. In: Proc. Modeling and verification of parallel processes, Springer LNCS 2067, pp. 187-195, 2001.
[EP2] EP2 Konsortium: eft/pos 2000, Project Overview available at http://www.eftpos2000.ch, 2002.
[GJ 99] M.-C. Gaudel and P. James: Testing Algebraic Data Types and Processes: A Unifying Theory. Formal Aspects of Computing, 10(5-6), pp 436-451, 1999.
[GRS 05] A. Gimblett, M. Roggenbach, and H. Schlingloff: Towards a formal specification of electronic payment systems in CSP-CASL; Selected papers from "WADT 2004. 17th International Workshop on Algebraic Development Techniques" Barcelona, Spain. (March 2004). Springer LNCS 3423, pp. 61-78 (2005)
[IR 05] Y. Isobe, M. Roggenbach: A Generic Theorem Prover of CSP Refinement. In: N. Halbwachs and L. Zuck (eds.): TACAS 2005. LNCS 3440, pp. 108-123, 2005.
[Mos 04] P. Mosses (ed.): CASL Reference Manual. Springer LNCS 2960, 2004.
[Rog 03] M. Roggenbach: CSP-Casl – A new integration of process algebra and algebraic specification. In: F. Spoto and A. Nijholt, AMiLP – 3rd AMAST Workshop on Algebraic Methods in Language Processing, pp 229-243. 2003.