

Refinement notions for CSP-CASL

Temsgen Kahsai and Markus Roggenbach

Swansea University, United Kingdom,
{csteme, csmarkus}@swan.ac.uk

In this talk we give a status report of an ongoing PhD project which develops and studies various notions of refinement for the specification language CSP-CASL [6]. CSP-CASL combines the description of *processes* written in the process algebra CSP [5, 7] with the specification of *data types* formulated in the algebraic specification language CASL [1].

The starting points for the PhD project are the various notions of refinement for CSP and CASL alone. For CSP, each of its semantic models induces a notion of refinement, i.e., the model \mathcal{T} induces the notion of trace refinement which preserves safety properties, the model \mathcal{N} induces the notion of failures-divergence refinement which preserves livelock-freedom, and the model \mathcal{F} induces the notion of stable-failures refinement which preserves deadlock-freedom. In algebraic specification [3], on the other side, we have model class inclusion as the simplest form of refinement, while, for instance, observational refinement [2] captures a more ‘refined’ relation between model classes.

In our project, we combine refinement notions on CSP and CASL alone into refinement notions for CSP-CASL. Having CSP-CASL available as an institution, every such CSP-CASL refinement for basic specifications can also be seen as a refinement that allows one to change the signature. The case study on the electronic payment system EP2 [4] yields good practical insight into the question if such a newly designed refinement notion is useful. On the theoretical side, we study decomposition theorems and the question if properties such as deadlock-freedom are preserved under CSP-CASL refinement.

References

1. E. Astesiano, M. Bidoit, H. Kirchner, B. Krieg-Brückner, P. D. Mosses, D. Sannella, and A. Tarlecki. CASL: the common algebraic specification language. *Theoretical Computer Science*, 286(2):153–196, Sept. 2002.
2. M. Bidoit and R. Hennicker. Constructor-based observational logic. *Journal of Logic and Algebraic Programming*, 67(1-2):3–51, Apr.-May 2006.
3. B.-B. E. Astesiano, H.-J. Kreowski. *Algebraic Foundations of Systems Specifications*. Springer, 1999.
4. A. Gimblett, M. Roggenbach, and H. Schlingloff. Towards a formal specification of an electronic payment systems in CSP-CASL. In *Revised Selected Papers of WADT’04*, LNCS 3423. Springer, 2005.
5. C. A. R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.
6. M. Roggenbach. CSP-CASL – A new integration of process algebra and algebraic specification. *Theoretical Computer Science*, 354:42–71, 2006.
7. A. Roscoe. *The theory and practice of concurrency*. Prentice Hall, 1998.