Zero-Safe Nets: Modeling Transactions via Transition Synchronization

Roberto Bruni and Ugo Montanari
Dipartimento di Informatica, Università di Pisa, Italia

P/T Petri nets [Rei85] are unanimously considered as one of the most attractive concurrent models. When designing large systems, the more convenient approach is to start by outlining a very abstract net model and then refining each transition into a net that offers a more precise description of the activity. For example, communication protocols for passing and retrieving information cannot leave aside the fact that agent synchronization is built on top of finer actions (e.g., for sending message requests and acks), which must be executed according to certain operational strategies. This means that transitions are seen, at the refined level, as atomic computations, that we call transactions, which succeed when all the involved components conclude their tasks. In particular the commit of a transaction synchronizes all final operations of local tasks. Several approaches have appeared in the literature that present refinement techniques for top-down design of concurrent systems (e.g., the *Petri Box calculus* [BDE93]), see [vGG98] for an overview. Typically, at each refinement step a single transition t of the net N is refined into a subnet M, yielding the net $N[t \leftarrow M]$, but some constraints must be assumed on M for its behavior to be consistent with that of t [Vog87, Val79, SM83]. We adopt a different approach, where all transitions of N are refined by runs of the same *zero-safe net*.

Zero-safe nets (ZS nets) have been introduced in [BM97] to provide a basic synchronization mechanism for transitions, which is an essential feature for writing modular and expressive programs (P/T net transitions can only synchronize tokens). Besides transitions and ordinary places (here called *stable places*), ZS nets include a set of *zero places*. These are idealized resources that remain invisible to external observers, while stable markings, which just consist of tokens in stable places, define the observable states of the system. The idea is that any evolution step of a ZS net starts at some stable marking, evolves through hidden states (i.e., markings with some tokens in zero places) and terminates into a stable marking. Moreover, all the stable tokens produced during a certain step are released together at the end of the step, when the commit is executed.

The synchronization of transitions is achieved via zero tokens: consider a transition t that produces a token in the zero place z, then to complete the step there must fire a transition t' which consumes the token in z, but since at the abstract level the computation is atomic, t and t' are synchronized. Pursuing this view, a 'refined' ZS net and an 'abstract' P/T net are supposed to model the same given system. The latter, where only stable places are considered, tells how to abstract from the flow of zero tokens. The resulting computational strategy is expressive enough to model, e.g., a multicasting system, simple process algebras and non-deadlocking strategies for free-choice nets (cf. [Bru99, BM99a]). We remark that the notions of 'atomicity' and 'evaluation strategy,' required for designing systems via refinement, are built-in inside the ZS net model, and that a finite refined ZS net can represent an infinite abstract P/T net.

The operational and abstract semantics of ZS nets under the *collective token* and *individual token* philosophies have been presented in [BM97, BM98], where they are formulated as universal constructions in category theory. A comparison between the two approaches and several applications can be found in [BM99a, Bru99]. See [BM99b] for a tutorial presentation of the material.

References

[BDE93] E. Best, R. Devillers, and J. Esparza. General refinement and recursion for the Petri Box calculus. In *Proc. STACS'93*, volume 665 of *LNCS*, pages 130–140. Springer, 1993.

[BM97] R. Bruni and U. Montanari. Zero-safe nets, or transition synchronization made simple. In *Proc. EXPRESS'97*, volume 7 of *ENTCS*. Elsevier, 1997.

[BM98] R. Bruni and U. Montanari. Zero-safe nets: The individual token approach. In *Proc. WADT'97*, volume 1376 of *LNCS*, pages 122–140. Springer, 1998.

[BM99a] R. Bruni and U. Montanari. Zero-safe nets: Comparing the collective and individual token approaches. *Inform. and Comp.*, 1999. to appear.

[BM99b] R. Bruni and U. Montanari. Zero-safe nets: Composing nets via transition synchronization. In *Proc. Int. Colloquium on Petri Net Technologies*, pages 43–80. Fraunhofer Gesellschaft ISST, 1999.

[Bru99] R. Bruni. *Tile Logic for Synchronized Rewriting of Concurrent Systems*. PhD thesis, Computer Science Dept., University of Pisa, 1999.

[Rei85] W. Reisig. *Petri Nets: An Introduction*. EACTS Monographs on Theoretical Computer Science. Springer, 1985.

[SM83] I. Suzuki and T. Murata. A method for stepwise refinement and abstraction of Petri nets. *J. Comput. and System Sci.*, 27:51–76, 1983.

[Val79] R. Valette. Analysis of Petri nets by stepwise refinement. *J. Comput. and System Sci.*, 18:35–46, 1979.

[vGG98] R.J. van Glabbeek and U. Goltz. Refinement of actions and equivalence notions for concurrent systems. Hildesheimer Informatik Bericht 6/98, Institut fuer Informatik, Universitaet Hildesheim, 1998.

[Vog87] W. Vogler. Behaviour preserving refinements of Petri nets. In *Proc. WG'86*, volume 246 of *LNCS*, pages 82–93. Springer, 1987.