

# Cyclic Communicating Processes: Hierarchy and Verification

P.S. Thiagarajan  
School of Computing  
National University of Singapore  
thiagu@comp.nus.edu.sg

## ABSTRACT

We advocate *Cyclic Communicating Processes* (CCPs) as a viable and tractable model of computation for reactive systems. As the name suggests, the model consists of a network of sequential agents that communicate with each other. The crucial restriction is that the control flow of each agent is *cyclic*. The communication mechanism consists of the agents performing common actions together.

At the level of pure control flow, this is a very restricted model and is well-understood. For instance, in domain of Petri nets, this class would correspond to marked graphs or more generally, *T*-systems [3]. However, this skeletal control flow model can be augmented along a number of dimensions to arrive at a fairly expressive and attractive system model that we call Communicating Cyclic Processes.

The first extension consists of endowing the control states with finite number of colors which can serve as abstractions of the values of the variables owned by the agents. This leads to a restricted class of colored Petri nets [5] that one could term as colored marked graphs or colored *T*-systems. This class of CCPs has a natural semantics in terms of finite 1-safe Petri nets. Though the underlying structure of 1-safe net systems that arise from CCPs can be fairly complex, our current research indicates that one can use powerful and yet decidable non-interleaved branching time logics to reason about their behaviors.

The second extension consists of adding the notion of hierarchy to the actions where each action is itself a CCP. In this setting, a CCP is best viewed as having a start event which sets the initial state and a finish event which traps one of a pre-determined set of final states. Suppose  $\gamma$  is a high level action which has a CCP denoted  $S_\gamma$  associated with it and  $P$  is the set of agents that participate in  $\gamma$  (at the top level). Then the start event of  $S_\gamma$  can be guarded by an assertion involving the values of the variables owned by  $P$  whereas the finish event can be guarded by an assertion involving the values of the variables owned by  $S_\gamma$ . This extension adds considerable expressive power, the resulting model has many of the appealing features of statecharts [4]

and it has a clean and simple operational semantics.

In the talk, we will concentrate on specification and verification issues related to branching temporal logics for CCPs. We will also highlight aspects of the following lines of current and future research:

- Identify applications where the control flow of the threads consist of nested loops involving sensing, actuating, computing and communication phases. A related topic is schedulability analysis for both uniprocessor and multi-processor environments. Adding timing constraints would also be a natural extension here.
- *Directly* verify behavioral properties of hierarchical CCPs along the lines of techniques developed in sequential settings [1, 2].
- Allow for infinite sets of colors for each control state but where the color sets and the dynamics have sufficient structure to permit symbolic representations and manipulations.

## References

- [1] R. Alur, R. Grosu, and M. McDougall. Efficient reachability analysis of hierarchic reactive machines. In *12th International Conference on Computer-Aided Verification, LNCS 1855*, 2000.
- [2] R. Alur, R. Grosu, and M. McDougall. Analysis of recursive state machines. In *13th International Conference on Computer-Aided Verification*, 2001.
- [3] J. Desel and J. Esparza. *Free Choice Petri Nets*. Cambridge University Press, 1995.
- [4] D. Harel. Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8, 1987.
- [5] K. Jensen. *Colored Petri nets. Basic concepts, Analysis methods and Practical use*. Springer-Verlag, 1997.