Brief Announcement: Incremental Component-Based Modeling, Verification, and Performance Evaluation of Distributed Reset*

Ananda Basu, Borzoo Bonakdarpour, Marius Bozga, and Joseph Sifakis

VERIMAG, Centre Équation, 2 ave de Vignate, 38610 Gières, France

1 Motivation

Design and implementation of distributed algorithms often involve many subtleties due to their complex structure, nondeterminism, and low atomicity as well as occurrence of unanticipated physical events such as faults. Thus, constructing correct distributed systems has always been a challenge and often subject to serious errors. This is essentially due to the fact that we currently lack disciplined methods for the rigorous design and correct implementation of distributed systems, mainly for two reasons: (1) formal methods are not easy to use by designers and developers; and (2) there is a wide gap between modeling formalisms and automated verification tools on one side, and practical development and deployment tools on the other side.

In this paper, we apply a methodology which consistently integrates modeling, verification, and performance evaluation techniques, based on the BIP (Behavior, Interaction, Priority) component framework developed at Verimag [2,3]. BIP is based on a semantic model encompassing composition of heterogeneous components. Partial state semantics of BIP allows generating from a high-level component-based model in BIP an observationally equivalent distributed implementation [2]. BIP uses two families of composition operators for expressing coordination between components: *interactions* and *priorities*. Interactions may involve multiple components (unlike traditional point-to-point formalisms) and are expressed by combining two protocols: *rendezvous* and *broadcast*. We note that addition of interactions among components adds no extra behaviors.

We illustrate our methodology using the self-stabilizing distributed reset algorithm due to Arora and Gouda [1]. The algorithm consists of two layers: (1) the tree layer, where adjacent processes communicate in order to construct and maintain a rooted spanning tree throughout the alive processes, and (2) the wave layer, which achieves a global reset through a diffusing computation. We demonstrate how BIP allows independent modeling, verification, and analysis of the tree layer and wave layer and ultimately their safe composition in order to construct a correct model of distributed reset. This composition involves in addition to interactions, scheduling constraints expressed as dynamic priorities among interactions.

* This work is sponsored by the COMBEST European project. For all correspondence about this please contact Borzoo Bonakdarpour at borzoo@imag.fr.

2 Approach and Results

Modeling. We model distributed reset according to the BIP system construction methodology: (1) designing the *behavior* of each atomic component (i.e., an automaton extended by variables, ports, and possibly C++ functionality), (2) applying synchronization mechanisms for ensuring coordination of components through *interactions* (i.e., broadcasts and rendezvous), and (3) specifying scheduling constraints by using *priorities*. We model each layer based on its normal operation in the absence of faults and self-stabilizing mechanism in the presence of faults. Each layer consists of a set of processes modeled by BIP behavioral components. The notion of faults such as process failures and variable corruptions is captured by internal transitions inside components. Processes in each layer communicate through interactions constrained by priorities. Upon the occurrence of faults, components execute their recovery mechanism to reach a legitimate state within a finite number of steps using the embedded interactions.
Verification. In order to model check the distributed reset algorithm, we construct a finite representation of the overall behavior of the model as a flat labeled transition systems Lts using BIP state-space explorer. States correspond to configurations reached by the algorithm, and transitions are labeled by the interactions taken to move from one configuration to another. Our properties of interest are: *closure*, *deadlock-freedom*, and *finite reachability* of the set of legitimate states starting from any arbitrary state. To reduce the complexity of verification, we incorporate a compositional approach by showing interference-freedom between the layers and manually apply model checking techniques on the BIP model such as *abstraction*, *live analysis*, and *sequence simplification*.
Performance Analysis. The BIP toolset provides us with means for generating C++ multi-threaded code from high-level BIP models. This feature enables us to evaluate the performance of distributed algorithms described by high-level models. It allows us to evaluate the impact of changes to the high-level model without getting involved with its actual C++ implementation. We emphasize that the logical properties and dynamics of the C++ model conform with the high-level model and an actual C++ implementation. In this context, we measure the degree of parallelism (i.e., the number of processes working simultaneously), that the BIP scheduler allows to achieve under different parallelism policies. Moreover, we analyze the severity of different types of faults and the effect of specifying stabilizing priorities in performance of the distributed reset algorithm.

References

1. A. Arora and M. Gouda. Distributed reset. *IEEE Transactions on Computers*, 43:316–331, 1994.
2. A. Basu, P. Bidinger, M. Bozga, and J. Sifakis. Distributed semantics and implementation for systems with interaction and priority. In *Formal Techniques for Networked and Distributed Systems (FORTE)*, pages 116–133, 2008.
3. A. Basu, M. Bozga, and J. Sifakis. Modeling heterogeneous real-time components in BIP. In *IEEE International Conference on Software Engineering and Formal Methods (SEFM)*, pages 3–12, 2006.