BIP: Language and Tools for Component-based Construction

Sergio Yovine
VERIMAG/CRNS
Sergio.Yovine@imag.fr
http://www-verimag.imag.fr/~yovine

1 Introduction

A central idea in systems engineering is that complex systems are built by assembling components. System designers deal with a large variety of components, each having different characteristics, from a large variety of viewpoints, each highlighting different dimensions of a system. A central problem is the meaningful composition of heterogeneous components to ensure their correct interoperation.

BIP (Behavior, Interaction, Priority) has been designed to overcome the difficulties of state-of-the-art component-based approaches. BIP provides a language and a theory for incremental composition of heterogeneous components, ensuring correctness-by-construction for essential system properties such as mutual exclusion, deadlock-freedom and progress. Besides, it enables formal verification.

BIP is the outcome of more than five years of research on a theory for compositional modelling real-time systems capable of handling urgency, parallel composition, priorities and deadlock-freedom by construction.

2 Description

BIP supports a methodology for building components from atomic ones, using connectors, to specify interaction patterns between ports of atomic components, and priority relations, to select amongst possible interactions.

BIP supports a component construction methodology based on the thesis that components are obtained as the superposition of three layers. The lower layer describes behavior. The intermediate layer includes a set of connectors describing the interactions between transitions of the behavior. The upper layer is a set of priority rules describing scheduling policies for interactions. Layering implies a clear separation between behavior and structure (connectors and priority rules).

BIP uses a parameterized binary composition operator on components. The product of two components consists in composing their corresponding layers separately. Parameters are used to define new interactions as well as new priority rules between the composed compo- nents. The use of such a composition operator allows incremental construction. That is, any compound component can be obtained by successive composition of its constituents. This is a generalization of the associativity/commutativity property for composition operators whose parameters depend on the order of composition.

BIP encompasses heterogeneity. It provides a powerful mechanism for structuring interactions involving strong synchronization (rendezvous) or weak synchronization (broadcast). Synchronous execution is characterized as a combination of properties of the three lay- ers. Finally, timed components can be obtained from untimed components by applying a structure preserving transformation of the three layers.

BIP allows considering the system construction process as a sequence of transformations in a three-dimensional space: Behavior Interaction Priority. A transformation is the result of the superposition of elementary transformations for each dimension. This provides a basis for the study of property preserving transformations or transformations between subclasses of systems such as untimed/timed, asynchronous/synchronous and event-triggered/data- triggered.

The BIP tool-chain allows the generation from a system

model in the BIP language, a description (C++ code) executable on a dedicated platform. The BIP platform supports both single-threaded and multi-threaded execution as well as enumerative state space exploration and simulation. The generated state graphs can be analyzed by using IF-platform.

3 Conclusions and Future

Current work on BIP deals with both theoretical issues and applications.

Theoretical work directions include the study of a notion of glue for components and its properties as well as a notion of expressiveness for component-based description languages.

Ongoing applications of BIP include: (1) "componentizing" existing real-time software written in C++ including an MPEG encoder and an adaptive robotic application; (2) implementing a tool for translating a subset of the BIP language into THINK, from which implementations on bare machines can be generated; and (3) designing a methodology for modelling and verification of networked systems written in NesC/TinyOS is also under development.

4 Links and References

URL: http://www-verimag.imag.fr/http://www-verimag.imag.fr/~async/bip.php

1. Simon Bliudze and Joseph Sifakis The Algebra of Connectors Structuring Interaction in BIP Proc. EmSoft07, ACM & IEEE, Oct. 1-3, 2007, Salzburg, Austria, pp 11-20.

2. Ananda Basu, Sergio Yovine and Marcelo Zanconi. An approach to derivation of component-based implementations from data-oriented specifications. APGES 2007, Oct. 4th 2007, Salzburg, Austria.

3. Ananda Basu, Laurent Mounier, Marc Poulhies, Jacques Pulou and Joseph Sifakis. Using BIP for Modeling and Verification of Networked Systems - A Case Study on TinyOS-based Networks Sixth IEEE International Symposium on Network Computing and Applications (NCA 2007), 12 - 14 July 2007, Cambridge, MA, USA, pages 257-260.

4. Susanne Graf and Sophie Quinton. Contracts for BIP: Hierarchical Interaction Models for Compositional Verification. In FORTE, pages 1-18, 2007.

5. Simon Bliudze and Joseph Sifakis. The Algebra of Connectors - Structuring Interaction in BIP. Technical Report TR-2007-3, VERIMAG, 2007.

6. Ananda Basu, Laurent Mounier, Marc Poulhis, Jacques Pulou, and Joseph Sifakis. Using BIP for Modeling and Verification of Networked Systems - A Case Study on TinyOS-based Networks. In NCA, pages 257-260, 2007.

7. Marc Poulhies, Jacques Pulou, Christophe Rippert and Joseph Sifakis. A Methodology and Supporting Tools for the Development of Component-Based Embedded Systems 13th Monterey Workshop 2006, Paris, October 2006, pp 75-96, LNCS 4888.

8. Ananda Basu, Marius Bozga, and Joseph Sifakis. Modeling Heterogeneous Real-time Components in BIP. In SEFM '06: Proceedings of the Fourth IEEE International Conference on Software Engineering and Formal Methods, pages 3-12, Washington, DC, USA, 2006. IEEE Computer Society.

9. Joseph Sifakis. A Framework for Component-based Construction Extended Abstract. In SEFM '05: Proceedings of the Third IEEE International Conference on Software Engineering and Formal Methods, pages 293-300, Washington, DC, USA, 2005. IEEE Computer Society.

10. Gregor Gossler and Joseph Sifakis. Composition for component-based modeling. Sci. Comput. Program., 55(1-3):161-183, 2005.