

Distributed Object and Component-based Software Systems

Barrett R. Bryant
The University of Alabama at Birmingham
bryant@cis.uab.edu

Rajeev R. Rajee
Indiana University Purdue University Indianapolis
rraje@cs.iupui.edu

This mini-track addresses component-based software, middleware, and distributed object systems. The mini-track chairs are grateful to the authors and referees for their participation in making this mini-track possible. Each of the 20 submissions was reviewed by 3-5 referees and 7 of these submissions appear here.

As with all software systems, requirements play a critical role in the initial construction of component-based software systems. In addition to object-oriented and component-oriented architecture, Wang and Fung explore the emerging paradigm of service-oriented architecture and how all three of these architecture models contribute to the software engineering of distributed systems. The approach of Shishkov and Dietz to software construction is based upon the object-oriented paradigm, deriving the software model from generic business components.

The integration of software components into a complete software systems requires the development of middleware. A middleware library, called DSS (Distributed SubSystem), for the purpose of developing distributed systems is presented by Klintskog, El Banna, Brand, and Haridi. Jandl, Alber, Radinger, and Goeschka use XML to specify middleware interfaces to facilitate the integration of distributed legacy systems, in particular CORBA software with directory services and web services. The CORBA Component Model is extended to distributed real-time and embedded

systems by Wang and Gill. An important innovation of this work is the consideration not only of functional correctness but also Quality of Service.

To facilitate the construction of high-performance distributed systems, Pérez, Priol, and Ribes describe PACO++, a portable implementation of parallel CORBA objects. The goal is to have a suitable programming model providing transparency, interoperability, reliability, scalability and performance. Espindola, Becker and Zorzo are concerned with requirements and specification of the distribution aspects of component-based systems. They extend the UML Components to allow early identification of such distribution requirements thereby providing for distribution in component implementation.