

Panel On Software Component Architectures

Mark J. Christensen, Chair
Northrop Grumman Corporation
Electronic and Systems Integration Division
Rolling Meadows, Illinois 60008, USA
chrisma@mail.northgrum.com

Why a panel on Software Component Architectures now?

The idea of software components and the architectural models, explicit or implicit, of the systems to be built from them has been around since the 1960's. The earliest examples of component were mathematical subroutines, with business application generators representing some of the earliest standardized architecture (e.g. template-based) systems. During the intervening years three technical developments have occurred. First, the range of application platforms has exploded, with embedded, desktop, network, and Internet based systems emerging. Second, the range of applications being build has expanded along with the breadth of the computer using public, with rich, real-time graphical implementations of those applications becoming the norm for systems with significant person/machine interaction. Thirdly, the programming languages, programming models, and tools have increased tremendously in their sophistication and abstraction capabilities. The final and fourth development is economic: The number of potential sales for a component or for products build from components has exploded as computers have infused all elements of life. The challenge to this panel is to shed light on these and other factors, together with the risks ahead.

Questions For The Panelists

The panelists will give their position papers and will then be asked to address the subset of the following questions that they feel are the most important within the allocated time:

1. What do you believe will be discriminating properties of components and the underlying tools and technologies over the next five years? What models dominate today?
2. What are the trade-offs and risks associated with proprietary or market-dominant systems? Is one case better for the component builder as distinct from the component user?
3. What are the special challenges for maintenance of systems utilizing component-based architectures? Maintenance begins during testing.
4. What principles should be applied to the selection of a component architecture as a function of the application and delivery platform?
5. How should components be specified, both statically (semantics) and dynamically (run-time)? Should this vary as a function of the component architecture?
6. Are there any 'Silver Bullets' for dealing with integrating multiple architectural and interface models into systems?

We expect that none of these questions will be answered completely but instead hope that the discussion will shed light on the complexity and opportunities for research in this area.

The Panel

Mikio Aoyama, Department of Information and Electronics Engineering, Niigata Institute of Technology, 1719 Fujihashi, Kashiwazaki, 945-11, Japan. mikio@iee.niit.ac.jp

Eric Hughes, The Mitre Corporation, Bedford, MA, 01739-1420, USA. hughes@mitre.org

Gilda Pour, Department of Computer, Information, and Systems Engineering, San Jose State University, One Washington Square, San Jose, CA 95192-0180. gpour@email.sjsu.edu