Experience with Software Architectures and Configured Software Descriptions

Walt Scacchi
Institute for Software Research
University of California Irvine
Wscacchi@ics.uci.edu

James S. Choi
Computer Science Dept.
Calif. State Univ. Fullerton
Sjchoi@cse.fullerton.edu
Overview

- Conceptual software architectures
- Logical software architectures
- Concrete software architectures
- Configured software descriptions
- Hybrid software architectural descriptions
Conceptual software architectures

- Pervasive, most common and widely used
- “Rendered“ as (un)directed attributed graph
  – box and arrow diagrams
- Generally fit on one page
- Useful for communication and first-order evaluation
- Can obscure architectural mis-matches
Logical software architectures

• Described in a MIL/ADL
  – Components/ connectors may have complex properties like pre/post-conditions
  – Formal properties beyond the skill of average software developers (especially in future!)

• Successful MIL/ADLs for very large systems will need to be light and automatically extracted from implementations
Concrete software architectures

• Address configurations of executable software components
  – compile-time, run-time, reachable run-time

• Involve multiple languages and associated information
  – repository (user_id, access control, timestamp, host_id), versions, pragmas, registries, initialization and configuration files, etc.
Concrete software architectures

• Multiple concrete architectures associated with single logical architecture
• Multiple logical architectures can be derived from single concrete architecture
• Concrete software architectures not well understood in all forms
Configured software descriptions

• Software *production* artifacts: software products, processes, network infrastructure, development organizations, documents, etc.
  – Each has its own architecture and configuration
  – Development of each impinges on others

• Software architectural design and configuration management address *the same problem with different abstractions*
Software Process Architecture: Decomposition View
(Scacchi 1999)
Hybrid software architectural descriptions

• Composition and integration of *multiple* software system architectures and configured software descriptions
  – Software *acquisition* architecture: software production architecture for *virtual system acquisition* across network of virtual enterprises
  – VSA incrementally builds software system architectures that use models, simulations and programs as components