Agent-based Support Environment for Flexible Architectural Analysis of Embedded Component-based Systems

Phillip Schmidt
The Aerospace Corporation
El Segundo, CA 90245-4691
Phillip.P.Schmidt@aero.org

Stephanie August
Loyola Marymount University
Los Angeles, CA 90045
august@acm.org
Outline

- Some Trends in Software Architectural Assessment
- Classic Embedded Development
- Challenges for Embedded Component-based Systems
- An Idea
- Approach
- An Architectural Analysis Environment
- An Agent-based Support Environment
- Implications for Future Architecture Design and Development
- Future Plans
Some Trends in Architectural Assessment Tools

- **Evolving UML-based tools**
  - Highly competitive market (betas every 2-3 months)
  - More complete UML representations
  - Integration with simulation/modeling support
  - Forward engineering support for component-based development environments
  - Embedded, real-time architectural support

- **More sophisticated Reverse engineering**
  - Class and sequence diagram support
  - Performance improvements

- **Improved event analysis**
  - Context events for architectural context
  - Transition events for improved granularity
  - Measured events for system state
Classic Embedded Development

- Development Environment
  - Target Development
    - VxWorks Tools
    - VxSim Environment
  - Target Server
- Target
  - VxWorks platform
  - Real-time App
  - Target Agent
  - Board Support Package

- Source
- Contract provided
- Commercial Tools
- Functional connectivity
- Physical connectivity
Challenges for Embedded Component-based Systems

- Ability to easily configure embedded application environment
  - Often tedious and error-prone
  - Environment often configurable but proprietary and constrained.
  - Custom solutions can be expensive
  - Multiple, disparate analysis tools

- Ability to dynamically assess execution context information to identify architectural shortfalls early is needed.
  - How best to study the composability of software components under different reconfigurable scenarios
  - Ability to ‘on-the-fly’ evaluate and manage context-dependent information

- Trend toward component-based architectures is promising for flexibility but better understanding of how best to dynamically configure them for analysis is needed.
An Idea!

- Leverage rapid evolution of architectural representation information with performance evaluation
  - Enabling assessment independent of development methodology
- Exploit internet technologies to simplify implementation
  - Application servers of dynamic content
  - Java VM environment (No Java support for direct memory access)
  - Resource description XML metalanguage
  - Active Java activities in embedded environments

Support an architectural assessment environment!
Approach

• **Adopt an embedded web server approach:**
  - Simplified webserver for memory and file constrained environments
  - Java integration with JNI required.

• **Use mobile agents to control, monitor context and configuration**
  - Support environment is developed in Java to exploit powerful language capabilities (Object-serialization, etc.)

• **Advantages:**
  - Standard protocol choices for data exchange:
    - Dynamic HTML, XML for device management,
  - Can leverage recent trend of using JVMs on embedded microprocessors as well as direct-execution Java technology
Architectural Analysis Environment

- Development Host
  - Architecture Assessment
  - Performance Models

- Experimentation
  - Simulation Control
  - Browser

- Target
  - VxWorks platform
  - Target Environment
    - Real-time App
    - Web Server
    - Agent Support

- Source

- Contractor provided
- Open Assessment
- Functional related
  - Functional connectivity
  - Physical connectivity

- Architectural Models
  - Architecture Documents

The Aerospace Corporation
Agent-based Support Environment for Architectural Assessment
Implications for Future Architecture Design and Development

- Support for mobile agent integration not difficult in Java
- Analysis rides on Java technologies
  - Object-serialization
  - JNI
  - Interest in RT Java will help experimentation as well
- Application dependent reconfiguration control interfaces
  - Something that just works is not good enough
Future Plans

- Configurable Rule-based context semantics
- Integrate the semantic information with dynamic assessment