Model-Driven Configuration and Deployment of Publisher/Subscriber Services

George Edwards
g.edwards@vanderbilt.edu

Gan Deng
gan.deng@vanderbilt.edu

Douglas C. Schmidt
d.schmidt@vanderbilt.edu

Bala Natarajan
bala@dre.vanderbilt.edu

Aniruddha Gokhale
a.gokhale@vanderbilt.edu

Vanderbilt University
Institute for Software Integrated Systems

Work supported by grants from:
NSF ITR CCR-0312859
Siemens
DARPA/AFRL Contract #F33615-03-C-4112

June 16, 2004
Presentation Outline

1. Challenges of Component Middleware Publisher/Subscriber Services

2. The Component-based Publisher/Subscriber Service Framework

3. The MDA Compliant Modeling Tool

4. Evaluate the Merits of the MDA Compliant Tool

5. Concluding Remarks
Publisher/Subscriber Service
Integration Challenges

• Context
  – The *container* manages application component access to common middleware services

• Problems
  – Prevents access to advanced publisher/subscriber service capabilities
  – No way to select among various publisher/subscriber services

• Solution
  – Enhance containers to encapsulate, implement, and configure a family of services.
Publisher/Subscriber Service Configuration Challenges

- **Context**
  - Publisher/subscriber services are highly configurable
  - XML-based specification of QoS properties

- **Problems**
  - Multiple dissimilar services
  - Semantically invalid operating policies
  - Error-prone handwritten XML

- **Solution**
  - Use models to enforce policy constraints and synthesize configuration files
Publisher/Subscriber Service Deployment Challenges

• Context
  – Federated publisher/subscriber services provide scalability
    • Optimize placement of event channels

• Problems
  – Multiple types of federations
  – Assignment of channels to hosts
  – Error-prone handwritten XML

• Solution
  – Use models to synthesize deployment descriptors
The MDA Compliant Modeling Tool

• It is a *Model-Driven Middleware* (MDM) tool
  – Addresses publisher/subscriber service configuration and deployment challenges
    • *Models* specify service configurations and deployments
    • *Aspects* decouple D&C concerns
    • *Constraints* ensure semantic validity
    • *Interpreters* generate descriptor files
Evaluating the Merits of MDA Tool

- Eliminates the need to write some C++ code
  - Manages event channel lifecycles
  - Initializes suppliers, consumers, and gateways
  - Specifies service properties
- Alleviates key sources of complexity
  - Modularizes cross-cutting concerns
    • Service configuration vs. component functionality
  - Reduces ad-hoc component development
    • Provides a methodology for creating specifications
  - Identifies design flaws earlier
    • Developers are notified of invalid configurations
  - Provides a reusable and maintainable representation of application properties
    • Models are easy to understand and evolve