Realizing Data Distribution Services through MDA

Sam Mancarella
Chief Technical Officer
Sparx Systems
Overview

- Model-Driven Architecture
- UML Extensibility
- Example DDS Realization
- Other Applications
- Concluding Remarks
MDA – What is it?

Model-Driven Architecture
- Models to ‘define’ architecture
  Use Case models, Interaction Overviews, Requirements
- Models to ‘describe’ architecture
  Class models, Activity models, Deployment models
- Models to ‘drive’ architecture
  One model to influence another (automated)

Longitudinal versus Latitudinal MDA
- ‘Across’ modeling languages
  BPMN, UML, SysML, CWM
- ‘Within’ a modeling language
  UML Use Case, UML Class

Apply Model Transformation
MDA – What is it?

- Design abstract services/design model, independent of the underlying technology
  - Platform Independent Model (PIM)

- Generate concrete services/design model, specific to the underlying technology
  - Platform Specific Model (PSM)

Automated Transformation

http://www.sparxsystems.com
MDA – Why is this important to DDS?

- Describe a DDS system in an abstract design (PIM)
  - DDS Domain
  - DDS Subscribers & Publishers
  - DDS Topics
  - Application, or system logic

- Automatically generate the implementation (PSM)
  - C++, Java, C Class models
  - Vendor-specific FreeDDS, RTI DDS, Thales DDS
MDA – Why is this important to DDS?

Benefits

- Reduces Complexity
  Application logic separate from implementation

- Reduces Design cycle time
  DDS design change can more easily be propagated down to implementation by retransforming PSM

- Increases Design Maintainability
  Segregation of logic design and implementation easier to troubleshoot
UML Extensibility

- UML Profiles
  - Provide a domain-specific taxonomy to UML
  - Define stereotypes that apply to various UML constructs
  - Define domain-specific semantics to those constructs
    - Properties (Tagged Values)
    - Constraints (human, machine)
  - Define the appearance of constructs

- We can define a UML Profile for DDS
UML Extensibility

Benefits

- **Standardizes Nomenclature**
  Domain, Topic, Subscriber as DDS profile constructs

- **Standardizes Semantics**
  A DataReader can only read one topic

- **Promotes Industry Standardization**
  Throughout RealTime-embedded community. DDS designers, consultants across vendors, implementations, geographies all share a common domain model

- **Maximizes Communication**
  A DataReader can only read one topic
Example DDS Realization

“Hello World” Example
- Simple example of 1 Topic, 1 Publisher, 1 Subscriber
- Modeled using a UML Profile for DDS

DDS Application Design
- How DDS Entities are defined and bound to each other
- How QoS Policies are allocated to entities

Implementation
- How the DDS design is used as a PIM to generate a PSM
- C++ Domain Class model
  - Data Dictionary
  - Factory
  - QoS
- Generate to executable code
Example DDS Realization
Example DDS Realization
Example DDS Realization
Example DDS Realization
Example DDS Realization
Example DDS Realization

http://www.sparxsystems.com
Other Applications

Maintenance
- Using a UML Agent to interface to a DDS
- Agent can perform a DDS ‘discovery’
- Publish the results to a DDS model visually
  - Common presentation of DDS discoveries
  - Visually inspect, or execute a ‘model check’ to flag faults
Other Applications

Prototyping

- Using a UML Agent to interface to a DDS as per Maintenance
- Automatically instantiate DDS Entities from a model
- Real-Time update of QoS policies
- Observe behavior
- Useful process in eliciting QoS policies for a deployment
  Which values work, or do not work
Concluding Remarks

Summary

- Understanding of how the MDA process functions between models
  PIM → PSM using automated transformation

- Extensibility of UML through Profiles
  UML Stereotypes for constructs to specify domain-specific modeling taxonomy

- Example of DDS Realization using MDA and UML Profile
  HelloWorld Example, how a simple PIM can generate a complex PSM without human intervention

- Other Applications
  Using a UML agent to visualize a ‘live’ DDS model
Concluding Remarks

- **Importance of Realizing DDS through MDA**
  - MDA Process to reduce complexity, reduce cycle times
  - UML Profile to standardize nomenclature, and increase industry acceptance

- **Current & Future Works**
  - Sparx Systems developing an integration application for RTI’s DDS implementation – publicly available September 2006
  - Sparx Systems plans to initiate an RFP for a UML Profile for DDS through the OMG