Using DDS to Enable The Real-Time Enterprise Service Bus (RT-ESB)

Rajive Joshi, Ph. D.
Gerardo Pardo-Castellote, Ph.D.
Real-Time Innovations, Inc

OMG Real-time and Embedded Systems Workshop
Arlington, VA, July 13, 2006
Agenda

- Life at the Edge

- The Gap between the Enterprise and the Edge

- Bridging the Gap: The Real-Time ESB
  - Critical technologies
  - Integrating DDS and JMS
  - Integrating DDS and SQL
  - Integrating DDS and Web Services
Life at the Edge: Air traffic control example

Airplane LAN

Control Tower LAN

Airport LAN

Real-Time Gateway

Gateway

Embedded

Gateway

Enterprise

Real-Time

Avionics

Alarms

Arrival time

Enterprise Gateway

Gateway

Backbone

(c) 2006 Real-Time Innovations, Inc.
The Gap: Enterprise surrounds the Edge

Presentation Layer: Portals, Web Apps, Mobile Apps & Thick Clients

Orchestration Layer: Composite Applications, BPM and Workflow

Business Logic Layer: Web Services-based Business Objects

Integration Layer: EII, EAI and Web Services

Data Layer: Systems of Record, Enterprise Applications and Data Systems

Data accessed from here

Multiple Networks & Middleware in between

Data originates here

Diagram Source: Redshaw, Motorola
The Gap: Differing Requirements

**Edge**
- High performance a must
  - low latency
  - high throughput
- Same middleware stack on both sides acceptable
- LAN environment / Trust boundary
- Scalability is important

**Enterprise**
- Effective Interoperability a must
- Different stacks on either side
- Lower performance acceptable
- May cross LAN/trust boundaries
- Scalability is important

*Impedance mismatch!*
Agenda

- Life at the Edge
- The Gap between the Enterprise and the Edge
- Bridging the Gap: The Real-Time ESB
  - Critical technologies
  - Integrating DDS and JMS
  - Integrating DDS and SQL
  - Integrating DDS and Web Services
Critical Technologies

Information Distribution Technology

- Live Data Access API
  - DDS
  - JMS

Data Storage Technology

- Stored Data Access API
  - SQL
  - ODBC
  - JDBC

Service Integration Technology

- Service Access API
  - Web Services
    - XML
    - WSDL
    - SOAP
Information Distribution

- Web Services
- Java
- Java/JMS
- CORBA
- RTSJ (soft RT)
- RTSJ (hard RT)
- Data Distribution Service / DDS
- RT CORBA
- MPI

Adapted from NSWC-DD OA Documentation
(c) 2006 Real-Time Innovations, Inc.
Really only three choices

- Use proprietary middleware
  - Eg. MQ Series, Tibco, In house, …
- JMS
  - Standards based
  - Popular in Enterprise
    - Java only
    - API only, no wire interoperability spec
- DDS
  - Standards based
  - Popular in “Edge” applications
    - Multiple platforms and languages
    - Wire interoperability
    - Real-Time QoS
    - High performance
Publish-Subscribe Middleware

Decouples Producers and Consumers

Declare Intent
Register Interest
Deliver
Data Distribution: DDS Synopsis

**Peer-to-Peer Interaction**

- **Data-centric**
- **Matching Topic, Compatible QoS** Data samples are routed directly from the *Publisher* to the *Subscriber*

**Application**

- DDS Peer (Producer)
  - Application
    - write()
  - DomainParticipant
    - DataWriter
    - Publisher

- DDS Peer (Consumer)
  - Application
    - read()
    - take()
  - DomainParticipant
    - DataReader
    - Subscriber

**DDS Domain (Global Data Space)**

- DataSample
- Key
- Data-object
- DDS Domain (Global Data Space)
DDS Synopsis: Requested/Offered Semantics

- **DDS Peer (Producer)**
  - Application
  - DataWriter (Publisher)
  - DomainParticipant
  - Offered QoS

- **Peer-to-Peer Interaction**
  - Data-centric
  - Matching Topic, Incompatible QoS
  - Communication is not established

- **DDS Peer (Consumer)**
  - Application
  - DataReader (Subscriber)
  - DomainParticipant
  - Requested QoS

- **Requested QoS Incompatible with Offered QoS**

- **Communication Not established**

DDSGlobal Data Space
JMS Synopsis

JMS Client (Producer)

Application

Message

Message Producer

Session

Connection

send()

Client-Server Interaction

Message-oriented

Messages are routed via the Destination

JMS Provider (Server)

Destination

Message

JMS Client (Consumer)

Application

MessageListener.onMessage(Message m)

receive()

Message

Message Consumer

Session

Connection

Administrative Tool (JNDI Namespace)
## JMS-DDS Semantic Mapping: Equivalents

<table>
<thead>
<tr>
<th>JMS</th>
<th>DDS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Client</strong></td>
<td><strong>Application</strong></td>
</tr>
<tr>
<td><strong>Provider = client runtime + server (if any)</strong></td>
<td><strong>Middleware, Service</strong></td>
</tr>
<tr>
<td><strong>Domains are PtP and Pub/Sub</strong></td>
<td>Domain represents a global data space, comprised of a set of communicating user applications</td>
</tr>
<tr>
<td><strong>ConnectionFactory</strong></td>
<td><strong>DomainParticipantFactory</strong></td>
</tr>
<tr>
<td><strong>Connection</strong></td>
<td><strong>DomainParticipant</strong></td>
</tr>
<tr>
<td>start()</td>
<td>enable()</td>
</tr>
<tr>
<td><strong>Session</strong></td>
<td><strong>Publisher, Subscriber</strong></td>
</tr>
<tr>
<td><strong>Destination</strong></td>
<td><strong>Topic (of type “Foo”)</strong></td>
</tr>
<tr>
<td>A named physical resource that gathers and disseminates messages addressed to it</td>
<td>An abstraction with a unique name, data-type, and QoS, used to connect matching DataWriters and DataReaders</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td><strong>Foo (data-object)</strong></td>
</tr>
<tr>
<td></td>
<td>An instance of type ‘Foo’</td>
</tr>
<tr>
<td><strong>MessageProducer</strong></td>
<td><strong>FooDataWriter extends DataWriter</strong></td>
</tr>
<tr>
<td><strong>MessageConsumer</strong></td>
<td><strong>FooDataReader extends DataReader</strong></td>
</tr>
</tbody>
</table>

FooTypeSupport extends TypeSupport
Used to register a user type ‘Foo’ with a DomainParticipant
## JMS-DDS Semantic Mapping: Differences

<table>
<thead>
<tr>
<th></th>
<th>JMS</th>
<th>DDS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Modeling</strong></td>
<td>Autonomous messages, Message-oriented, Payload is opaque</td>
<td>Data-objects, Data-centric, Relational data model</td>
</tr>
<tr>
<td><strong>Dataflow Routing</strong></td>
<td>Specific destinations</td>
<td>Matching Topics and Compatible QoS</td>
</tr>
<tr>
<td><strong>Discovery</strong></td>
<td>Administered</td>
<td>Spontaneous, Access to endpoint meta-data</td>
</tr>
<tr>
<td><strong>Data Typing</strong></td>
<td>Predefined message types</td>
<td>Arbitrary user data types</td>
</tr>
</tbody>
</table>
# JMS and DDS: Practical Considerations

<table>
<thead>
<tr>
<th></th>
<th>JMS</th>
<th>DDS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Architecture</strong></td>
<td>Client/Server, Centralized, Asymmetric</td>
<td>Peer-to-Peer, Decentralized, Symmetric</td>
</tr>
<tr>
<td><strong>Platforms</strong></td>
<td>Java</td>
<td>Multiple languages (C, C++, Java, …) &amp; platforms</td>
</tr>
<tr>
<td><strong>Interoperability</strong></td>
<td>None. No plans.</td>
<td>Wire protocol standardization in progress</td>
</tr>
<tr>
<td><strong>Transports</strong></td>
<td>Connection oriented. Requires in-order acknowledged delivery.</td>
<td>Connection-less unreliable transport like UDP</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>(Username, Password)</td>
<td>Extensible mechanism</td>
</tr>
<tr>
<td><strong>Administration</strong></td>
<td>JNDI, External to JMS</td>
<td>None: plug-n-play</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td></td>
<td>Lower latency, higher throughput. Independent 10x better claims</td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td>Limited by: PtP, Destinations</td>
<td></td>
</tr>
<tr>
<td><strong>Real-Time support</strong></td>
<td>Transactions, Application acknowledgements, Java EE</td>
<td>Many capabilities designed for real-time, low latency apps</td>
</tr>
<tr>
<td><strong>Enterprise support</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Using DDS and JMS together

- **JMS-DDS Bridging**
  - Translate between DDS updates \(\leftrightarrow\) JMS messages
  - Data conversion overhead
  - Least common denominator
  - Integrate different sub-systems

- **JMS/DDS Bindings**
  - Wrap DDS middleware with JMS APIs
  - Port JMS applications to DDS
  - JMS apps interoperable with DDS
  - Higher performance JMS
Data Storage: Integrating DDS and SQL
Service Integration: Integrating DDS and Web Services

Information Distribution Technology

Live Data Access API

DDS JMS

Data Storage Technology

Stored Data Access API

SQL
ODBC
JDBC

WSDL ↔ SQL Mapping

WSDL ↔ DDS Mapping

Service Integration Technology

Service Access API

Web Services
XML
WSDL
SOAP
What is a Web Service?

- **W3C Definition**
  - A software system designed to support interoperable machine-to-machine interaction over a network

- **Service Data Model: XML Schema**
  - XML = “Extensible Markup Language”
  - XML Schema for describing XML documents (Data Types)

- **Service Contract: WSDL**
  - “Web Services Description Language”
  - Defines all details about a service contract (Interfaces)

- **Encoding of Messages: SOAP**
  - “Simple Object Access Protocol”
  - How to format XML documents for transmission between applications (Serialize/Deserialize)

Family of Extensible Specifications
An Example Web Service

Aircraft Tracking Service
  Data: XML
  Interface: WSDL

- Service Data Model
  - TrackData
    - x
    - y
    - z

- Service Contract
  - GetPosition
    - Message In
      - Aircraft Id
    - Message Out
      - TrackData

XML Schema
<complexType name="TrackData">
  <element name="x" type="xsd:double"/>
  <element name="y" type="xsd:double"/>
  <element name="z" type="xsd:double"/>
</complexType>

WSDL
<portType name="AircraftTrackingServicePort">
  <operation name="GetPosition">
    <input message="AircraftIdMessage" name="aircraftId"/>
    <output message="TrackDataMessage" name="trackData"/>
  </operation>
</portType>

<message name="AircraftIdMessage">
  <part element="xsd:int" name="aircraftId"/>
</message>

<message name="TrackDataMessage">
  <part element="TrackData" name="trackData"/>
</message>
A Web Service is Just a Specification

- Service Requester
- Service Provider
- Service Proxy
- Service Platform
- Service Contract
- Service Registry/Lookup
- Service Proxies & Skeletons
- Service Contracts
- Service Registry/Lookup
- Service Proxies & Skeletons
- Service-Level Data Model
- Service-Level Security
- Service-Level QoS
- Service-Level Management
- Service-Level QoS
- Service-Level Comm Model
- Multi-language bindings

Technology Independent Service Level Data Model
Technology Independent Service Level Contract

Binding

Mapping of Requestor’s Internal implementation Interfaces and data model to Service Level Contract and Data Model

Mapping of Service Level Contract and Data Model to Provider’s internal implementation Interfaces and data model

Adapted from: Newcomer & Lomow

(c) 2006 Real-Time Innovations, Inc.
Binding a Web Service to a Transport

WSDL Defines...

- Operations, Structure and Typing of Messages
- Encoding & Transport of Messages
- Physical Endpoint Address

Service Communications Requirements:

- XML Schema

Service Contract

Runtime execution

Service Requester

SOAP Request

XML

Service Provider

SOAP Response

HTTP, CORBA, JMS, DDS, ...

Adapted from: Newcomer & Lomow
The Enterprise Service Bus (ESB) Pattern

- Collection of distributed, interconnected end-points
- Support multiple transports / middleware
- No internal canonical data format
- Instead: support transformation and routing

Adapted from: Iona, ‘Next Generation ESBs’
Enabling the Real-Time ESB

- Using DDS as a Transport for a Web Service
  - WSDL/DDS Bindings
  - WSDL/SQL Bindings
    - Using DDS-SQL Integration to access the global data space
- Exposing DDS as a Web Service
  - Exposes the underlying relational data model
Using DDS as a Web Service Transport

Web Service Requestor

WSDL/SOAP Request

WSDL/SOAP Reply

Web Service Provider
(Web Server or App Server)

JAX-WS (Java WSDP)

WSDL\leftrightarrow DDS Data Mapping

Java DDS API Bindings

Application Code

Auto-generated

DDS Domain (Global Data Space)
Using DDS-SQL Integration as a Web Service Transport

Web Service Requestor
- WSDL/SOAP Request
- WSDL/SOAP Reply

Web Service Provider (Web Server or App Server)
- JAX-WS (Java WSDP)
- WSDL↔DB Data Mapping
- JDBC

DB
- SELECT
- UPDATE
- DELETE
- CREATE TABLE
- DROP TABLE

DDSQL/SkyBoard

Auto-generated Application Code

DDS Domain (Global Data Space)
DDS as a Web Service: Expose the Data Model

**Service Data Model**
- ParticipantInfo
- DataTypeInfo
- TopicInfo
- PublisherInfo
- DataWriterInfo
- SubscriberInfo
- DataReaderInfo
- DataReaderListenerInfo

**Service Contract**
- CreateParticipant
- RegisterType
- CreateTopic
- CreatePublisher
- CreateDataWriter
- CreateSubscriber
- CreateDataReader
- SetDataReaderListener

**XML Schema Example**
```
<complexType name="ParticipantInfo">
  <element name="domain_id" type="xsd:int"/>
</complexType>
```

**WSDL Example**
```
<portType name="DDSPort">
  <operation name="CreateParticipant">
    <input message="ParticipantInfoMessage" name="participantInfo"/>
    <output message="URIMessage" name="participantURI"/>
  </operation>
</portType>
```

(c) 2006 Real-Time Innovations, Inc.
New Possibilities using the Real-Time ESB

- Bridge the gap between the Edge and the Enterprise
  - Resolve the impedance mismatch
    - Real-Time Database Cache
  - Exchange data between the edge and the enterprise
  - Access enterprise data from embedded devices
- Integrate with disparate technologies
  - Leverage integration of other technologies with Web Services
  - Realize the promise of an “ESB” from the edge to the enterprise
- Real-Time SOA
  - Enable high performance or real-time Web Services
  - Using DDS as a transport for a Web Service
- Remote monitoring and control of DDS applications
  - Using DDS as a Web Service
Bridging the Gap: Air traffic control example redux

- Airplane LAN
- Control Tower LAN
- Airport LAN

- Avionics
- Alarms
- Arrival time

- Edge
- Gateway
- Enterprise

- DDS
- RT-ESB
- WSDL or SQL
- ESB
Bridging the Gap: Integrating DDS with JMS, SQL, Web Services

Information Distribution Technology

- Live Data Access API
- DDS Mapping
- DDS JMS

Relational Data Model
- WSDL ↔ DDS Mapping
- WSDL ↔ SQL Mapping

Data Storage Technology
- Stored Data Access API
- SQL ODBC JDBC

Service Integration Technology
- Service Access API
- Web Services XML WSDL SOAP

Bridges between the Enterprise and the Edge Technologies
The Real-Time Enterprise Service Bus

Publish-Subscribe

Producer (DDS or JMS)

Consumer (DDS or JMS)

Databases

SQL Client

SOA

Web Service Requestor

Real-Time Enterprise Service Bus

“Pervasive Data”
Global Data Space
Thank You

Q & A

Rajive.Joshi@rti.com

Real-Time Innovations, Inc

For more Information:

- www.rti.com
  - RTI Whitepaper: A Comparison and Mapping of Data Distribution Service (DDS) and Java Message Service (JMS)