Introduction to DDS

Washington DC, December 2006

Gerardo Pardo-Castellote, Ph.D.
Co-chair OMG DDS SIG
gerardo.pardo@rti.com

www.rti.com
Agenda

- History
- What is Data-Centricity?
- The Future
  - Enabling Unified Global Data
  - A Real-Time “Service” Bus
History: DDS the Standard

- Data Distribution Service for Real-Time Systems
  - Adopted in June 2003
  - Finalized in June 2004
  - Revised June 2005, June 2006
  - Joint submission (RTI, THALES, OIS)
  - Specification of API for Data-Centric Publish-Subscribe in real-time distributed systems.

- Multiple Implementations
  - 3 commercial
  - 3 open source
  - Several more in-house

- Interoperability in progress at OMG
  - Recommended for adoption in July 2006
DDS mandated for data-distribution

- **DISR (formerly JTA)**
  - DoD Information Technology Standards Registry

- **US Navy Open Architecture**

- **FCS SOSCOE**
  - Future Combat System – System of System Common Operating Environment

- **In Progress**
  - **RETF**
    - Railroad Electronics Task Force
  - **UK MOD**
    - Advocating Open Systems
US Navy Programs

- DDG 1000 – previously DD(X)
- LCS – Littoral Combat Ship
- SSDS – Ship Self Defense System
- SPY OA – Aegis System
- LPD 17
- Sea Slice
- E2-C Hawkeye
- ...
DDS Adoption

- EU Air Traffic Management
- Train Communications
- Tokyo Japan Traffic Control
- Boeing Army Future Combat System
- Boeing AWACS program
- US Navy, DD(X) LCS, LPD-17 SeaSlice and 13 other Navies
DDS Adoption

- **Aerospace & Defense**
  - BAE (Joint Strike Fighter avionics)
  - USA, CAE, NADS, Boeing (Simulators)
  - TCG, Lincoln Labs, General Dynamics (C4ISR)
  - Boeing, Lockheed, Northrop (Navy OA)
  - SAIC (Ground vehicle control)

- **Industrial Automation**
  - Schneider (Factory automation)
  - Applied Materials, Nikon (Semiconductor equipment)
  - Ferag (Post printing assembling and binding)
  - Schilling (Robotics)
  - Max Planck (Power research)

- **Telecomm/Datacomm**
  - Accom (Digital video control)
  - Tekelec (Network test equipment)
  - IPC (Telecomm equipment)
  - Infinera (Optical switch control)
Agenda

- History
- What is Data-Centricity?
- The Future
  - Enabling Unified Global Data
  - A Real-Time “Service” Bus
What is DDS? DDS/DCPS

Provides a “Global Data Space” that is accessible to all interested applications.

- Data objects addressed by **DomainId, Topic**, and **Key**
- Subscriptions are **decoupled** from Publications
- Contracts established by means of **QoS**
- Automatic **discovery** and **configuration**
What is DDS? DDS/DLRL

Provides “Local Object Caches” built from the Global Data Space.

- Objects manipulated with a “natural” language binding
  - Inheritance, Object Graphs, supported as language objects
- Actions on local objects cause updates to DCPS Global Data
- No need for a “global” object model
- QoS contracts still available via underlying DCPS
DDS Global Data

- Address in Global Data Space = (DomainId, Topic, Key)
  - Each topic corresponds to a multiple data instances
  - Each DataWriter can write to multiple instances of a single topic
  - Multiple DataWriters may write to the same instance
  - Each DataReader can receive updates from multiple instances of a single topic
  - Multiple DataReaders may read from the same instances
DDS communications model

- Publisher declares information it has and specifies the Topic
  - and the offered QoS contract
  - and an associated listener to be alerted of any significant status changes

- Subscriber declares information it wants and specifies the Topic
  - and the requested QoS contract
  - and an associated listener to be alerted of any significant status changes

- DDS automatically discovers publishers and subscribers
  - DDS ensures QoS matching and alerts of inconsistencies
## QoS: Quality of Service

<table>
<thead>
<tr>
<th>Infrastructure</th>
<th>Delivery</th>
<th>User QoS</th>
<th>Presentation</th>
<th>Redundancy</th>
<th>Transport</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volatility</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DURABILITY</td>
<td>USER DATA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HISTORY</td>
<td>TOPIC DATA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>READER DATA LIFECYCLE</td>
<td>GROUP DATA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WRITER DATA LIFECYCLE</td>
<td>PARTITION</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LIFESPAN</td>
<td>PRESENTATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENTITY FACTORY</td>
<td>DESTINATION ORDER</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESOURCE LIMITS</td>
<td>OWNERSHIP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RELIABILITY</td>
<td>OWNERSHIP STRENGTH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIME BASED FILTER</td>
<td>LIVELINESS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEADLINE</td>
<td>LATENCY BUDGET</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONTENT FILTERS</td>
<td>TRANSPORT PRIORITY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
QoS: Deadline

**Deadline Topic**

- **Publisher**
  - Commits to provide data each deadline period.

- **Data Writer**

- **Data Reader**

- **Subscriber**
  - Expects data every deadline period.

- **Listener**
  - Failed to get data

**Deadline**

- **S**
- **X**
- **S**
- **S**
- **S**
- **S**
- **S**
- **S**

**DEADLINE “deadline period”**
What makes DDS different?

- **Data-centricity**
  - High level of data abstraction: Topic, Key
  - Proven scalable model for RT systems
  - “Smart” services such as:
    - Ownership, ContentFilteredTopics, KeepLast History
  - Automatic discovery
  - Directly supports state propagation/caching

- **Configurability by QoS**
  - Wide range of applicability: Enterprise to real-time
  - P2P infrastructure:
    - High-performance and scalability
    - Fault-tolerance
    - Scalability
  - Subsumes message-oriented and data-centric

- **Object model built as local cache**
Agenda

- History
- What is Data-Centricity?
- The Future
  - Enabling Unified Global Data
  - A Real-Time “Service” Bus
Data-Distribution and Real-Time

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

Adapted from NSWC-DD OA Documentation
Until now: Different Data Solutions

- Database Management Systems
  - Good for: Complex queries, dynamic sorting, standard SQL I/F, enterprise solution
  - But... No RT performance, centralized, non-distributed

- Data Distribution Services
  - Good for: High performance, dynamic architectures, real-time solution
  - But... what do you do with the data once you get it there?
A new model is possible: Standards-Based Global Data Space

- Data accessible to all interested applications:
  - Data distribution (publishers and subscribers): DDS
  - Data management (storage, retrieval, queries): SQL
  - Rich QoS, automatic discovery and configuration
  - Real-time and/or high-performance access to data
Global Data & End-to-End Integration

- Data access from the Web Services or Enterprise networks does not hinder the real-time performance Network.
- Additional portals to other systems can be added dynamically.
DDS Opportunities

Net-centric interface to tactical systems

Simulation Systems

Surveillance Systems

Financial Systems
OMG Opportunity

• Open, Standard Platform Enabling Integration
  – from the Enterprise Service Bus (ESB)
  – to the Real-Time Service Bus (RTSB)
Thank you

Gerardo Pardo-Castellote, Ph.D.
gerardo.pardo@rti.com

www.rti.com