Developing MilSOFT DDS Middleware
OMG Real-time and Embedded Systems Workshop
Arlington-VA USA-July-9-12,2007

Huseyin Kutluca
MilSOFT A.S, Teknokent ODTU, Ankara/Turkey
hkutluca@milsoft.com.tr

Izzet Emre Cetin, Ertan Deniz, Baris Bal, Murat Kilic, Ugur Cakir
Outline

- Introduction
- MilSOFT DDS
  - Why
  - Features
  - Architecture
  - RTPS
- Tools
  - DDS Code Generator
  - DDS SPY
  - DDS Tester
- Performance Results
- DDS Applications
- Conclusion
Introduction - Why DDS implementation

• Standards based middleware
  • No legacy system dependencies
• Corba vs. DDS
• High performance middleware for real time distributed systems
• Product certification requirements
• Easy adoption to different platforms including avionics systems
Introduction-History

Gemkomsis Combat Management System

- R&D project from 2004 to 2007
- Partially supported by Turkish Research and Development Institute
- CMS system based on OMG & Open Architecture Computing Environment standards
- Ship common Functions and Ship Common Services
- DDS middleware development

- OACE Computing environment (including DDS) finds its use in multiple projects
  - Turkish Coast Guard SAR Ship CMS
  - Multilink Data link Processor
  - UAV Image Exploitation system
MiISOFT DDS

- Started with DDS specification 1.0 adopted version
- Current version compatible with DDS specification version 1.2
- C++ API for applications
- RTPS
MilSOFT DDS Implemented Features

- **Minimum Profile (Complete)**
  - Minimum profile contains just the mandatory features of the DCPS layer. None of the optional features are included.

- **Persistence Profile (Complete)**
  - Persistency profile is completely implemented. Durability service is designed to be fault tolerant.

- **Content Subscription Profile (Partially Implemented)**
  - ContentFilteredTopic and QueryCondition features have been implemented. MultiTopic has not been implemented yet.

- **Ownership Profile (Partially Implemented)**
  - Only History QoS with depth > 0 feature has been implemented.

- **Object Model Profile (Not Implemented)**
  - Contains DLRL and Presentation QoS access_scope = GROUP
MilSOFT DDS Architecture

- Applications use DDS as library
  - No shared memory or DDS service per node
  - Completely decentralized
- RTPS implemented as the underlying communication protocol
  - DDS Interoperability wire-protocol
- High performance
- Zero-copy
- No dynamic resource (memory, thread, etc) allocation after initialization
- Multicast address per topic concept
More information on http://dds.milsoft.com.tr
What Is RTPS?

- DDS interoperability wire protocol
  - Real-Time Publish Subscribe (RTPS)
- Adopted by OMG in June 2006
- MilSOFT started to develop RTPS from draft versions
- Run over multicast and connectionless best-effort transports
- Current MilSOFT DDS implements RTPS over UDP
RTPS modules

• Structure Module
  • Communication entities/endpoints
  • “Static” Relationships
  • Configuration parameters/QoS

• Messages Module
  • List of messages
  • Contents of messages
  • Interpretation

• Behavior Module
  • Dynamic behavior of protocol

• Discovery Module
  • Bootstrapping (Discovery of participants)
  • Configuration of endpoints
RTPS implementation - Design Decisions

- Used UDP protocol for message exchange
  - Other protocols like Shared Memory or TCP can be adopted
- IP multicast for transmission to multiple subscribers
- Dynamic discovery of RTPS endpoints
  - Implemented using DDS BuiltinTopics
  - No centralized service for discovery
- Listener interface over RTPS
  - Separate RTPS from DDS implementation
- Zero copy support
  - No copy of data once it’s in RTPS buffers
  - Users can access the data without copying using DDS API
DDS Tools

- MilSOFT DDS Automatic Code Generator
- MilSOFT DDS SPY
- MilSOFT DDS Tester
DDS Automatic Code Generator

- Type specific Code
  - FooTypeSupport
  - FooDataReader
  - FooDataWriter
- IDL like definition from GUI
- Definition of DDS types
- IDL like struct data stored into XML file
- Generation of type specific files
  - From DDS Code Generator GUI
  - From command line using XML file
DDS Automatic Code Generator

Generating middleware specific code using DDS Automatic Code Generator has the following benefits:

• Application developers do not need to know programming details of DDS API
• Tool usage speeds up software development as major code is generated by tool
• It eliminates errors generated while implementing middleware interface
• It ensures type consistency and interface control through the project
• Easy management of system internal interfaces from configuration control point of view
DDS SPY-Display Data

- Listens network and displays DDS related information
  - Participating applications to specific DDS
  - Topics each application publishes or subscribes
  - Data itself
- User selectable listening mechanism
  - Auto
  - Manual
- Shows data in multiple views
  - Current snapshot of the topic
  - All historical data
DDS SPY-Display Data
DDS SPY-Inject DDS Data

• Injects data into DDS domain
  • Copy-Paste previously transmitted data and retransmit
  • User can load data from excel file or copy-paste from excel file.
• Send more than one topic simultaneously with timestamps
• Great tool for system integration and testing
# DDS SPY-Inject DDS Data

## DDS SPY - Domain ID: 100

### Process Status Topic

<table>
<thead>
<tr>
<th>Time Stamp</th>
<th>Message Type</th>
<th>Global PID</th>
<th>Local PID</th>
<th>Hostname</th>
<th>Process State</th>
<th>Error Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>DISPOSE</td>
<td>1598</td>
<td>1478</td>
<td>gemkomsisdev4</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

### Process Status

- **Global PID**: 1598
- **Local PID**: 1478
- **Hostname**: gemkomsisdev4
- **Process State**: 1
- **Error Type**: 2

![DDS SPY - Domain ID: 100](image)
MilSOFT DDS Tester

- Automated test tool for applications communicating DDS middleware
- Allows definition test suites and test cases
- Send and receive topic data to/from middleware
- Compare test results with excepted results and show them as highlighted
- Show test results (failed tests, successful tests)
MiSOFT DDS Performance Test Results

• Test environment
  • 2 computers with Intel Xeon 3.0 Ghz processors.
  • Fedora Linux 4
  • Gigabit ethernet switch

• Test 1: Latency
  • Ping (Message_size)/pong time(message_size)/2

• Test2: Round trip with 4 byte ACK
  • Ping different message sizes pong is always 4 bytes

• Test3: Throughput
  • Send as fast as possible
  • Measure data size received in one second
Two Host Average Latency

TwoHosts,
Avg. Latency vs Packet Size

Packet Size (Bytes)

Avg. Latency (Microseconds)

Best-Effort-ZeroCopy
Best-Effort-Copy
Reliable-ZeroCopy
TwoHosts,
Round-Trip Time vs Packet Size (4 Byte Reply)

- BestEffort-Copy
- BestEffort-ZeroCopy
- Reliable-ZeroCopy

Packet Size (Bytes) vs. Round-Trip time for 4 Byte Reply (microseconds)
### Throughput

**Throughput vs Packet Size**

- **Throughput (KB/sec)**
  - 100
  - 1000
  - 10000
  - 100000
  - 1000000

- **Packet Size (Bytes)**
  - 10
  - 100
  - 1000
  - 10000
  - 100000

- **Lines**
  - One Subscriber
  - Two Subscribers, Subscriber #1
  - Two Subscribers, Subscriber #2
MilSOFT DDS Applications

• GEMKOMSIS CMS
  • Single middleware for handling all communication
  • RM and common services on top of DDS

• SARSHIP CMS

• UAV Video Exploitation System
  • Stream data/video over DDS middleware

• CAVLIS
  • Multilink DLIP

• Middleware for CMS, UAV and DLIP product lines
GEMKOMSIS CMS

Platform Unique Functions and Interfaces

Ship Common Functions

Ship Common Services

Middleware
- Distribution Middleware (DDS)
- Adaptation Middleware (ACE)
- Frameworks

COTS Computing Technology
- LINUX Operating System
- INTEL Processors
- Dual Gigabit Ethernet

Resource Management
- Application
- Time
- Node
- Network
DDS in GEMKOMSIS CMS

Single standards based middleware for CMS data distribution

- Resource Manager over DDS
  - Compatible with OMG AMSM specification
- Alarm Manager over DDS
  - Compatible with OMG ALMAS specification
- Debug Log Framework
  - Collection of log information to centralized location or maintenance application over DDS
  - Even USE DDS SPY for displaying log information
- Record Replay over DDS
  - Training
  - Post-mission analysis

- Multiple word using DDS domains
  - Real-world
  - Simulation world
  - Replay world

- High performance DDS middleware enables transmitting time critical and high frequency data through middleware
Quality Attributes Addressed by MilSOFT DDS

- **Performance**
  - High performance publish-subscribe based DDS middleware
  - Multicast based data distribution with DDS
  - Zero-copy and no-dynamic resource allocation implementation

- **Reliability**

- **Modifiability / Scalability**
  - Layered architecture
    - Middleware separates application logic from computing resources
  - Data driven publish/subscribe system enables
    - No component interdependencies
    - New component additions without changing other components

- **Availability**
  - Fault Recovery
    - DDS Durability Service (transient and persistent data)
    - Liveliness QoS

- **Testability**
  - DDS Spy as test and diagnostic tool
    - Capture and display data
    - Retransmit data for testing
DDS on UAV Image Exploitation System

Sensor Data (EO/IR, SAR, GMTI)

Telemetry

Data Receiver

Raw Data Recording

High Capacity Storage Unit

Pre-processed & Georecorded Data

Pre-processor

DDS based Data Streaming

Raw Data

Screening

Exploitation 1

Exploitation 2

All Data in DB

Artifacts

Copyright © MilSOFT, Turkey

UNCLASSIFIED
Conclusion

• Standards based DDS implementation
• Successfully used in multiple projects
• Opportunities
  • Avionics
  • Java World
• Questions?