



CORBA/e
CORBA for embedded

*OBJECTIVE
INTERFACE
SYSTEMS*





Why CORBA for Embedded Systems

*REAL-TIME AND EMBEDDED
MIDDLEWARE PRODUCTS*

- Common Object Request Broker Architecture – CORBA
 - Open standard maintained by Object Management Group consortium
 - High performance, low footprint implementations
- Why middleware?
 - Standalone systems are a thing of the past
 - Platform flexibility is essential
 - Software is becoming larger part of embedded systems
- Why CORBA?
 - Distribution – integrate into “system of systems” and enterprise management environment
 - Portability – migrate across processors, operating systems, communications infrastructures
 - Interoperability – interoperate with older models and new enterprise systems

- Embedded systems are constrained
 - SWaP – size, weight and power
 - Higher unit counts – different economics
- Different processing infrastructures
 - Processors
 - Peripherals
 - Communications
- CORBA/e – “CORBA *for embedded*”
 - Standalone specification
 - Proven performance and utility in the embedded market
 - Retains interoperability



- Purpose – to reduce
 - Resource usage – to extend CORBA's applicability to systems constrained by
 - Footprint
 - CPU
 - Complexity of implementation – higher assurance
 - Complexity of specification – fewer choices
- Result - two profiles containing "static" parts of CORBA
 - Compact profile – embedded 32-bit μ processors
 - Micro profile – smaller μ processors, high-end DSPs



- *CORBA/e Compact Profile*
 - Replaces “minimum CORBA” profile
 - Includes
 - “Static” parts of CORBA
 - Essential parts of Real-time CORBA
 - Basic services – Naming, Events, Logging
- *CORBA/e Micro Profile*
 - Restricts types
 - Further simplifies servers
 - Real-time predictability



CORBA/e Table of Contents

*REAL-TIME AND EMBEDDED
MIDDLEWARE PRODUCTS*

1. CORBA Overview
2. Object Model
3. OMG IDL Syntax and Semantics
4. Repository IDs
5. ORB Interface
6. Object Interfaces
7. Policies
8. The Portable Object Adapter
9. Real-time
10. Naming Service
11. Event Service
12. Lightweight Logging
13. General Inter-ORB Protocol
14. CDR Transfer Syntax
15. GIOP Messages
16. Internet Interoperability Protocol (IIOP)
 - A. OMG IDL Tags and Exceptions
 - B. Legal Information



1. CORBA Overview
2. Object Model
3. OMG IDL Syntax and Semantics
 - Full grammar must be accepted (parsed)
 - Can ignore
 - context clauses
 - abstract interfaces
 - value boxes
 - custom valuetypes
 - value "supports" interface
 - import



4. Repository IDs – OMG IDL formats only
5. ORB Interface
 - No dynamic TypeCode creation
 - No domain managers
 - Added back: shutdown, destroy
6. Object Interfaces
 - Object - added back: is_a, nonexistent
 - ValueType Semantics
 - Lightweight valuetype
 - No “poly lithic” valuetype instances



CORBA/e Compact Profile Lightweight Value Types

*REAL-TIME AND EMBEDDED
MIDDLEWARE PRODUCTS*

- Syntactically eliminates
 - Value boxes
 - Abstract interfaces
 - Derivation from interfaces
- Semantically eliminate “polyolithic” valuetypes containing
 - Other valuetypes
 - Type Any – which could contain other value type
 - Any other components (structs, unions, etc.) transitively containing
 - Other valuetypes
 - Type Any
- Eliminates need for garbage collection (complexity, unpredictability)



7. Policies

- Policy objects
- Effectiveness
- Messaging QoS Policies
 - Rebind support
 - SyncScope support
 - RoundTrip Timeout support



8. The Portable Object Adapter

- “Static parts” of POA specification
 - *A la* MinCORBA
 - No servant managers
 - No default servants
 - No POA policies to support above



9. Real-time Features

- “Hard real-time” parts of RTCORBA
 - Priorities
 - Definition
 - Mapping
 - Client propagated model
 - Mutexes
 - Binding
 - Banding
 - Invocation timeout



- Services
 - 10.Naming Service
 - 11.Event Services – Untyped only
 - 12.Lightweight Log Service



- Interoperability
 - 13. General Inter-ORB Protocol (GIOP)
 - 14. CDR Transfer Syntax
 - 15. GIOP Messages
 - 16. Internet Interoperability Protocol (IIOP)
- Annexes
 - A. OMG IDL Tags and Exception
 - B. Legal Information



1. CORBA Overview
2. Object Model
3. OMG IDL Syntax and Semantics
 - Full grammar must be accepted (parsed)
 - Can ignore
 - context clauses
 - abstract interfaces
 - value types
 - type Any

4. Repository IDs – OMG IDL formats only
5. ORB Interface
 - All sub-clauses required
 - No dynamic TypeCode creation
 - No domain managers
 - Added back shutdown, destroy
6. Object Interfaces
 - Object - added back: is_a, nonexistent
 - No ValueType Semantics



7. Policies

- All sub-clauses required
- Policy objects
- Effectiveness
- Messaging QoS Policies
 - Rebind support
 - SyncScope support
 - RoundTrip Timeout support

8. The Portable Object Adapter
 - Single root POA – default policies
 - “Static parts” of POA specification
 - Ala MinCORBA
 - No servant managers
 - No default servants
 - No POA policies to support above
9. Real-time
 - Mutexes only
 - Services – not required



- Interoperability (common with CORBA/1)- all sub-clauses
 - 13.General Inter-ORB Protocol (GIOP)
 - 14.CDR Transfer Syntax
 - 15.GIOP Messages
 - 16.Internet Interoperability Protocol (IIOP)
- Annexes
 - A. OMG IDL Tags and Exception
 - B. Legal Information
 - C. Acknowledgement

- CORBA/e Compact Profile
 - Compact, yet powerful – targeted to resource-constrained 32-bit μ p
 - Deterministic
 - Combines static core of CORBA with core of Real-time CORBA
 - Priority propagation, priority banding
 - Server options – transient or persistent retained servants
 - CORBA Services – Naming, Events, Logging
 - Interoperable – native IIOP
- CORBA/e Micro Profile
 - Truly micro – mobile or low-power μ p, high-end DSP
 - Deterministic core – real-time internals
 - Compact server-side
 - Interoperable – native IIOP



- CORBA/e Specification
 - <http://doc.omg.org/realtime/2006-02-02>
- PIDL & IDL for profiles
 - <http://doc.omg.org/realtime/2006-01-03>