minimumCORBA Tutorial

Bill Beckwith
Objective Interface Systems, Inc.
+1 703 295 6500
bill.beckwith@ois.com
http://www.ois.com
OMG Real-time and Embedded Workshop
July 2002

Nature of minimumCORBA specification

- This presentation is from CORBA v2.6 specification
- Subset of CORBA designed for systems with limited resources
- Trade-off between usability and conserving resources
- Fully interoperable with both minimumCORBA and full CORBA applications
- Single profile that preserves the key benefits of CORBA
  - portability of applications
  - interoperability between ORBs
- Omission of features
  - Features omitted because of “cost, in terms of resources”
  - minimumCORBA defines a profile (or subset) of CORBA
  - Whereas CORBAservices & CORBAsecurity define extensions to CORBA spec
  - Features that support the dynamic aspects of CORBA are omitted
- IDL
  - All IDL features included
Omitted from CORBA::ORB
psuedo-interface

- DII related
  - create_list() operation
  - create_operation_list() operation
  - Context object
  - get_default_context() operation
- Not required for basic ORB operation
  - work_pending() operation
  - perform_work() operation
  - shutdown() operation
- Note that run() is retained
- Deprecated in CORBA 2.2
  - get_current() operation

Omitted from CORBA::Object
psuedo-interface

- Interface Repository related (IR removed)
  - get_interface() operation
  - Deprecated in CORBA 2.2.
    - get_implementation() operation
  - is_a() operation
    - Avoids holding detailed type information in object reference or for getting type information over the wire
    - minimumCORBA relies on design time resolution of type checking
  - non_existent() operation
    - make more decisions statically at design time
- DII related
  - create_request() operation
Additional major features removed

- Dynamic Invocation Interface (DII)
  - Omitted entirely
  - NamedValue type removed
  - NVList type removed
- Dynamic Skeleton Interface (DSI)
  - Omitted entirely
- Dynamic Any type
  - Omitted entirely

Interface Repository (IR)

- Majority omitted
- Except
  - RepositoryId
  - TypeCode interface
- RepositoryId pragmas retained
  - Enable compact type naming
Interface Repository - TypeCode

- Retained for type Any
- Thus only certain operations required
  - `id()` operations
  - `kind()` operations
  - `name()` operations
- Operations for arbitrary constructed and template types omitted
  - `member_count()` operation
  - `member_name()` operation
  - `member_type()` operation
  - `discriminator_type()` operation
  - `member_label`,
  - `default_index()` operation
  - `length()` operation
  - `content_type()` operation
  - `fixed_digits()` operation
  - `fixed_scale()` operation
  - `param_count()` operation
  - `parameter()` operation
  - Bounds exception
  - All create operations in ORB interface

TypeCodes constant creation operations
- `create_struct_tc()` operation
- `create_union_tc()` operation
- `create_enum_tc()` operation
- `create_alias_tc()` operation
- `create_exception_tc()` operation
- `create_interface_tc()` operation
- `create_string_tc()` operation
- `create_wstring_tc()` operation
- `create_sequence_tc()` operation
- `create_recursive_sequence_tc()` operation
- `create_array_tc()` operation

Interface Repository (cont.)

- Operations for arbitrary constructed and template types omitted
  - `member_count()` operation
  - `member_name()` operation
  - `member_type()` operation
  - `member_label`,
  - `discriminator_type()` operation
  - `default_index()` operation
  - `length()` operation
  - `content_type()` operation
  - `fixed_digits()` operation
  - `fixed_scale()` operation
  - `param_count()` operation
  - `parameter()` operation
  - Bounds exception
  - All create operations in ORB interface

TypeCodes constant creation operations
- `create_struct_tc()` operation
- `create_union_tc()` operation
- `create_enum_tc()` operation
- `create_alias_tc()` operation
- `create_exception_tc()` operation
- `create_interface_tc()` operation
- `create_string_tc()` operation
- `create_wstring_tc()` operation
- `create_sequence_tc()` operation
- `create_recursive_sequence_tc()` operation
- `create_array_tc()` operation

© 2002 Objective Interface Systems, Inc.
Portable Object Adapter – Interfaces

- PortableServer::POA Interface
  - Omitted from minimum CORBA version of module PortableServer
    - POA interface dynamic mode of operation
  - Only default values for POA policies support so no need for
    - policy object factory operations
      - create_thread_policy() operation
      - create_implicit_activation_policy() operation
      - create_servant_retention_policy() operation
      - create_request_processing_policy() operation
  - Dynamic (on demand) activation of POAs omitted
    - the_activator attribute
  - ServantManager omitted
    - get_servant_manager() operation
    - set_servant_manager() operation
  - USE_DEFAULT_SERVANT option for the RequestProcessingPolicy omitted
    - get_servant() operation
    - set_servant() operation

- PortableServer::Current interface
  - Fully supported

Portable Object Adapter – Interfaces (cont.)

- PortableServer::POAManager interface
  - All containing declarations omitted except
    - activate() operation
    - AdapterInactive exception

- PortableServer::AdapterActivator interface
  - AdapterActivator entirely omitted
    - supports dynamic mode of POA operation that is not required for basic ORB operation

- PortableServer::ServantManager interface
  - ServantManager interface omitted
    - Dynamic mode of operation that not required for basic ORB operation
  - ServantActivator interface omitted
  - ServantLocator interface omitted
  - PortableServer::ForwardRequest omitted

© 2002 Objective Interface Systems, Inc.
Portable Object Adapter – Policies

- supported policies include only the default policy values
- minimumCORBA RootPOA is subset of CORBA RootPOA
  - more restrictive than its CORBA RootPOA counterpart
- application built on the minimumCORBA RootPOA will run on the CORBA RootPOA
- PortableServer::ThreadPolicy
  - only ThreadPolicy is ORB_CTRL_MODEL
  - SINGLE_THREAD_MODEL policy omitted
- PortableServer::LifespanPolicy
  - supports both values of LifespanPolicy - TRANSIENT and PERSISTENT
- PortableServer::ObjectIdUniquenessPolicy
  - supports both values of ObjectIdUniquenessPolicy
    - UNIQUE_ID
    - MULTIPLE_ID
- PortableServer::IdAssignmentPolicy
  - supports both values of IdAssignmentPolicy
    - SYSTEM_ID
    - USER_ID

Portable Object Adapter – Policies (cont.)

- PortableServer::ServantRetentionPolicy
  - supports only the RETAIN ServantRetentionPolicy
  - NON_RETAIN policy omitted - unnecessary dynamic behaviors
- PortableServer::RequestProcessingPolicy
  - supports only the USE_ACTIVE_OBJECT_MAP_ONLY RequestProcessingPolicy
  - USE_DEFAULT_SERVANT policy omitted - unnecessary dynamic behaviors
  - USE_SERVANT_MANAGER policy omitted - unnecessary dynamic behaviors
- PortableServer::ImplicitActivationPolicy
  - supports only the NO_IMPLICIT_ACTIVATION policy
  - IMPLICIT_ACTIVATION omitted - unnecessary dynamic behaviors
  - CORBA RootPOA has an ImplicitActivationPolicy of IMPLICIT_ACTIVATION
  - minimumCORBA RootPOA default is NO_IMPLICIT_ACTIVATION policy
Miscellaneous

- Interoperability
  - DCE ESIOP omitted
- COM/CORBA Interworking
  - COM and CORBA interworking omitted
- Interceptors
  - Interceptors omitted

Language Mappings

- must support at least one language mapping
- no specific language binding is mandated
- the full mapping must be supported except for those omitted core objects
- further omissions from the C++ and Java mappings
- C++ Mapping Specific Issues
  - implicit activation via _this()
  - minimumCORBA implementations may offer implementation-specific removal of code required for
    - type-safe narrowing
    - multiple inheritance of IDL interfaces
- Java Mapping Specific Issues
  - Java ORB Portability Interfaces omitted - depend on DII and DSI

© 2002 Objective Interface Systems, Inc.
Implementation Experiences

- Many have criticized minimumCORBA for retaining too much
- Result is that several vendors have defined their own "RootPOA" profile
  - Smaller than minimumCORBA
  - Can't create any child POAs at all
- Market result is three profiles
  - Proprietary Root POA profiles
  - minimumCORBA profile
  - Full CORBA profile

Implementation Experiences – Target System Requirements

- Very small: < 512K
  - 8 and 16 bit processors
  - Automotive
  - Sensors
  - Board-level component
  - Either Root POA profile or too small for CORBA
- Small: 512K - 4 MB
  - 32 bit processors
  - Cell phones, radios
  - Root POA profile or minimumCORBA profile
- Medium: 4 MB - 16 MB
  - 32 bit processors
  - Most common
  - minimumCORBA profile (+ Real-time CORBA + ...)
- Large: 16 MB - 256 MB
  - 32 and 64 bit processors
  - Image processing
  - Radar processing
  - Signal processing
  - minimumCORBA profile or full CORBA profile
    - (+ Real-time CORBA + Data Parallel CORBA + ...)
- Very Large: 256+ MB
  - 64 bit processors
  - Embedded supercomputers
  - minimumCORBA profile or full CORBA profile
    - (+ Real-time CORBA + Data Parallel CORBA + ...)

© 2002 Objective Interface Systems, Inc.
Future of specification

- Minimum CORBA 2002 Revision Task Force
  - RTF Chair: bill.beckwith@ois.com
  - OMG mailing list: mincorba-rtf@omg.org
  - To add an issue: issues@omg.org
  - RTF Public Comment Deadline: December 2, 2002
  - RTF Revision Deadline: February 3, 2003

- Items to address
  - Value Types
  - Abstract Interfaces
  - Interoperability with Non-CORBA Systems
  - Portable Interceptors (and not "Interceptors")
  - CORBA Messaging

Further Information

- minimumCORBA
  - OMG CORBA 2.6.1 specification, chapter 23

- Information about CORBA for Real-Time, Embedded, and High Performance Applications
  - http://www.ois.com/resources/corb-1.asp

- Real-time and embedded CORBA discussion forum
  - http://www.realtime-corba.com