Change Proposal to Draft SCA Next CORBA Profiles for Harmonization with ESSOR Architecture

Document WINNF-11-R-0006

Version V1.0.0

14 September 2011

TERMS, CONDITIONS & NOTICES

This document has been prepared by the SCA Next Working Group to assist The Software Defined Radio Forum Inc. (or its successors or assigns, hereafter “the Forum”). It may be amended or withdrawn at a later time and it is not binding on any member of the Forum or of the SCA Next Working Group.

Contributors to this document that have submitted copyrighted materials (the Submission) to the Forum for use in this document retain copyright ownership of their original work, while at the same time granting the Forum a non-exclusive, irrevocable, worldwide, perpetual, royalty-free license under the Submitter’s copyrights in the Submission to reproduce, distribute, publish, display, perform, and create derivative works of the Submission based on that original work for the purpose of developing this document under the Forum's own copyright.

Permission is granted to the Forum’s participants to copy any portion of this document for legitimate purposes of the Forum. Copying for monetary gain or for other non-Forum related purposes is prohibited.

THIS DOCUMENT IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY USE OF THIS SPECIFICATION SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE FORUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS DOCUMENT.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.

Attention is also drawn to the possibility that the Forum shall not be responsible for identifying any or all such IPR.

Wireless Innovation Forum ™ and SDR Forum ™ are trademarks of the Software Defined Radio Forum Inc.

Table of Contents

TERMS, CONDITIONS & NOTICES i
1 References 1
2 Introduction 1
2.1 Purpose of the document 1
2.2 Scope of the document 1
2.3 Layout of the document 1
3 General Orientations 3
4 Detailed Rationale 4
4.1 CORBA for FPGA Environments 4
4.2 Ultra lightweight profile for DSP and FPGA processing elements 6
5 Change Proposal 9
5.1 Impacted Draft SCA Next documentation 9
5.2 Proposed modification #1 9
5.3 Proposed modification #2 9
5.4 Proposed modification #3 10
5.5 Proposed modification #4 10
5.6 Proposed modification #5 10
5.7 Proposed modification #6 11
5.8 Proposed modification #7 11
5.9 Proposed modification #8 17

Preface

The JTRS JPEO has released in December 2010 the Draft SCA Next specification.

Simultaneously, the ESSOR Industries have finalized the elaboration of the ESSOR SCA-based Architecture.

The SCA Next Work Group of the WInnF prepared this change proposal in an attempt to maximize the harmonization between ESSOR and JTRS achievements, using as an input an ESSOR contribution submitted to WInnF SCA-Next WG.

This document on Ultra Lightweight CORBA Profile is being submitted by the WInnF to the JTRS SCA Next Working Panel as a recommendation for adoption. It is the SCA-Next WG expectation that it will be followed by JTRS SCA Next Working Panel in order to achieve harmonization on the topic of interest.

Request for Comment on Amendment to SCA Next CORBA Profiles for harmonization with ESSOR Architecture

1 References

[Ref 1] – Draft SCA Next

[Ref 2] – WINNF-11-I-0008

2 Introduction

2.1 Purpose of the document

As indicated in Preface, this document aims for the WInnF SCA Next WG to provide a Change Proposal in order to harmonize parts of the SCA Next with the ESSOR SDR Architecture.

It takes the following inputs as references for the harmonization effort:

- **[Ref 1]** Draft SCA Next, as published by JTRS and available on its website,
- **[Ref 2]** WINNF-11-I-0008, Extract from ESSOR SDR Architecture relative to CORBA Profiles for SCA Next.

WINNF-11-I-0008 contains an extract from the ESSOR SDR Architecture submitted by ESSOR Industries to WInnF SCA Next WG for consideration in the frame of SCA Next elaboration process.

2.2 Scope of the document

This document describes the main characteristics and features that should be provided by DSP and FPGA ORB environments, in order to support both CORBA and non-CORBA based waveform application components running on such environments and to enhance portability of such components on CORBA-capable Processing Elements (PE).

This document refers to the new "Ultra lightweight" (U) profile for DSP and FPGA.

Previously, this document relates to Change Proposal S047 of the SCA Next. It used as a basis the related presentation delivered by JTRS during the SCA Next Rollout meeting held on August 2010:

http://sca.jpeojtrs.mil/_downloads.asp?folder=scanext&file=113CORBAEvolution.pdf.

2.3 Layout of the document

The document is structured in three parts:

- General Orientations,
- Detailed Rationale,
- Recommended Change Proposal.

3 General Orientations

1) CORBA for FPGA environments (§ 3.1):

In order to have an ORB on an FPGA an Hardware-ORB solution is recommended; the features such ORB should support are the ones reported for the Ultra LW profile described in this document.

2) Ultra lightweight profile for DSP and FPGA processing elements (§ 3.2)

The “Ultra lightweight” is intended for DSP and FPGA processing elements and is a subset of the other two profiles currently present in the SCA Next draft specification, which are typically applicable for GPP and DSP environment.

The definition of such profile is needed since typically not all the IDL features are needed in DSP or FPGA, nor easily implementable on these kinds of processing elements; furthermore this profile improves waveform portability by limiting waveform-components interfaces complexity.

4 Detailed Rationale

The “CORBA Ultra lightweight” profile is intended for DSP and FPGA and it aims at improving waveform portability, also when going from a DSP to an FPGA and viceversa.

For example, a WF component developed for a DSP environment, and whose IDL interfaces are designed according to the Ultra lightweight Profile, can be ported to an FPGA, starting from the same IDL interfaces, and keeping the behaviour of the component transparent towards the rest of the system, independently from its location. The opposite is valid too, i.e. an FPGA component can be easily ported to a DSP.

4.1 CORBA for FPGA Environments

FPGAs are programmable logic processing elements, fairly different from instruction-set-based processors as GPPs and DSPs, however they are able to host an ORB middleware as well.

CORBA on FPGAs can be essentially achieved in two ways:

1. SW ORB - by using a processor core embedded in the FPGA
2. HW ORB - by using a hardware-ORB, i.e. an ad-hoc IP-core implemented in the common resources present in FPGAs that carry out the functionalities of an ORB.

The first approach can be considered as having a GPP inside an FPGA and hence the standard SCA approach applies to it. So here only the second approach is considered.

As far as it concerns the second approach FPGA are not characterized by the presence of an Operating System that provides basic services to Waveform components. FPGAs functionalities are performed by reconfigurable hardware structures, physically distributed on the chip surface and having “true parallel” operations execution: the intrinsic parallelism of such devices in fact allows functionalities to be performed simultaneously.

Since last years IP vendors have been marketing HW ORBs for FPGA; these products are to be considered as other common FPGA IPs commercially available, such as fifos, ethernet MAC, etc.

The Hardware-ORB approach has substantial performance advantages in relation with a Software-ORB. It solves many issues of the first approach that fall into categories such as performance (e.g. not being able to be clocked fast enough to deal with the ever-increasing performance requirements of SDR applications), size (e.g. the IP processor core taking up large amounts of gates) or development times (as processors embedded in FPGAs represent a more complicated HW/SW technology than separated processors and FPGAs).

The hardware ORB engine can be delivered as an IP core and is responsible for implementing the transfer syntax at GIOP (General Inter Orb Protocol) level, used in CORBA messages. The engine unmarshals an incoming GIOP stream and extracts header and data fields.

Endianness conversion (if needed) is performed on incoming data, based on information in the GIOP message header. In the incoming direction, the engine performs operation named demultiplexing to determine which object the data in the GIOP message is being transferred to. Message data is then extracted and transferred to the appropriate logic.

If the incoming GIOP message indicates that a response is expected, the ORB engine generates a reply message. The ORB-engine performs a read operation to the involved object in order to obtain data for the reply. When a reply message has been built, the ORB-engine transfers the data to the outside world via the transport interface.

Similar to IDL compilers provided by software ORBs that maps IDL into software languages, an IDL-to-HDL compiler is part of the hardware ORB development environment. This compiler is also responsible for generating configuration parameters needed by the ORB engine to perform the operation demultiplexing and data routing described earlier.

By means of this IDL-compiler, IDL interfaces specification, such as the WF component interfaces (CF Resource and user-defined), can be compiled and supported in a native FPGA implementation of the SCA WF component.

The hardware developer treats the hardware ORB as any other IP interface core. The ORB-core can be instantiated (in the HDL capture of the FPGA design) between the native waveform logic and the Transport side. The transport side of the ORB-core appears typically as a FIFO interface. The WF-logic side of the core has a simple and open interface to communicate with the waveform logic, depending on the supported IDL interfaces.

Error! Reference source not found. illustrates the basic CORBA environment on FPGA, related with the transport facility, that is one of the most critical and essential part of the support facilities to be provided.

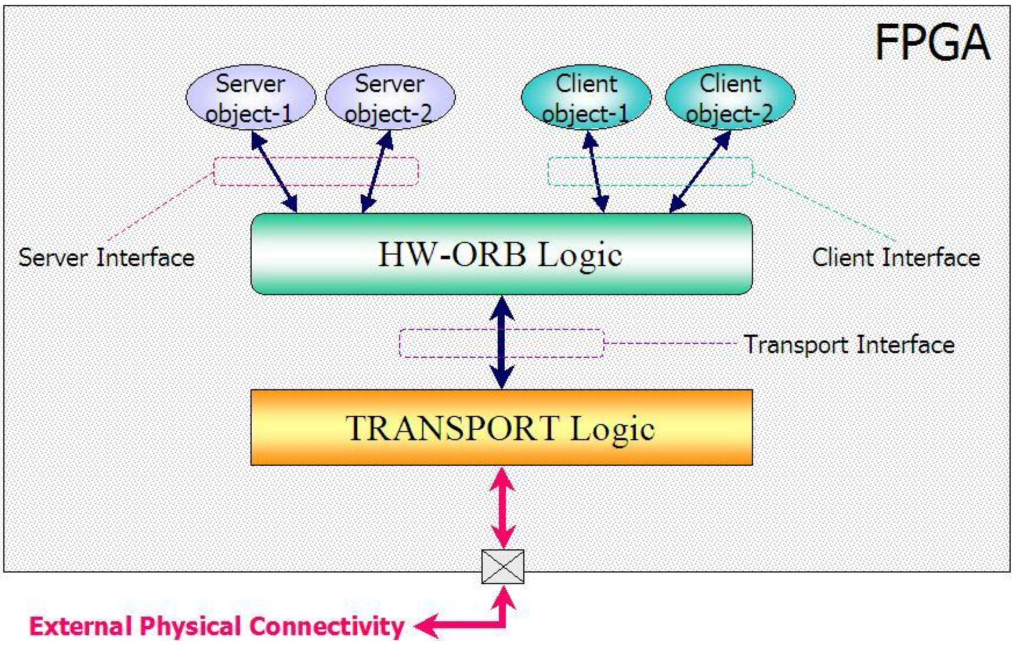

Figure 1 - FPGA basic environment for CORBA support

Software developers can treat hardware-implemented CORBA components as they would treat any other CORBA object. This design approach makes communication between the SW and HW objects seamless. Using a hardware ORB, waveform developers can host WF elements in an FPGA, which can still be addressed and called by an SCA-compliant software as though it was a standard SCA object, without any perception that it is residing in an FPGA.

4.2 Ultra lightweight profile for DSP and FPGA processing elements

DSP and FPGA hosted waveform components are typically used for the fast processing capabilities that these kinds of processing elements present.

All the IDL features are neither typically needed in DSP or FPGA processing nor easily implementable on these kinds of processing elements. In order to ease waveform portability it is defined the following “Ultra lightweight”, for both CORBA and Non-CORBA approaches; it is composed by supported features and optional features.

The Ultra lightweight profile is defined in terms of IDL features.

The IDL features of the Ultra lightweight profile have been selected considering the following points:

- typical DSP/FPGA algorithms implementations have to operate very fast, but on simple data types
- it is possible to convey information (back and forth) by using simple data containers
- the IDL features of the Ultra lightweight profile have to be supported by COTS ORBs current implementations (also in FPGAs that represent the more constrained processing elements in terms of ORB support)
- IDL interfaces based on such a profile shall be not too limited, but present an adequate flexibility to allow the required data communication

These features are a subset of the other two profiles currently present in the SCA Next draft specification, which are the “Full” and “Lightweight”.

IDL basic data types	`Short`
	`Long`
	`unsigned short`
	`unsigned long`
	`Boolean`
	`Octet`
IDL complex data types	`struct` (restricted to supported basic data types)
	`sequence` (restricted to supported basic data types)
	`Enum`
IDL keywords	`Module`
	`Interface`
	`In`
	`Out`
	`Inout`
	`Void`
	`Typedef`
	`oneway`
Return value	Return values of a basic data type to be supported

Table 1 – Ultra lightweight profile – IDL supported features

Feature	Description
Usage of struct in sequence.	A sequence can contain also struct complex type (of supported basic data types), in addition to supported basic data types, as already specified in the supported features.
Unbounded sequence	A sequence should be bounded whenever it is possible. In other words, the usage of unbounded sequences should be advised only if unavoidable.
Const, FALSE, TRUE.	Constant data values can be specified by using supported basic data types. FALSE and TRUE are the values needed by the boolean basic data type.
raises (exception)	The usage of exceptions should be avoided whenever possible. In fact in most cases their function can be carried out and so replaced by using a 'out' parameter or a return value.

Table 2 – Ultra lightweight profile – IDL Optional features

5 Change Proposal

5.1 Impacted Draft SCA Next documentation

APPENDIX D.1: PLATFORM SPECIFIC MODEL (PSM) - COMMON OBJECT REQUEST BROKER ARCHITECTURE (CORBA) of the SCA NEXT draft (30 November 2010).

5.2 Proposed modification #1

Proposed version for D.1.2 CORBA PROFILES paragraph:

The specification includes three SCA CORBA profiles based on the CORBA/*e* specification with
additional features from RT CORBA. The SCA CORBA profiles are characterized as follows:
1. Full CORBA Profile – is the Full SCA CORBA profile
2. Lightweight CORBA Profile – is more constrained than the Full profile and is targeted towards environments with limited computing support.
3. Ultra lightweight CORBA Profile – is more constrained than the Lightweight profile and is specifically intended for processing elements with even more limited computing support.

In particular:

- the Full CORBA Profile is intended for SCA Applications that will be hosted on most General Purpose Processor (GPP) platforms
- the Lightweight CORBA Profile is intended for Applications that will be hosted on platforms such as DSPs
- the Ultra lightweight Profile is intended for Applications hosted on both DSPs and FPGAs. Being focused on the IDL, such profile allows portability.among such Processing elements, for both CORBA and MHAL approaches.

These profiles include features that have been chosen to support SCA Applications while avoiding features that require excessive processing resources. While platform designers will often know the resource availability and may choose to use resource intensive features, Applications are usually intended to be portable between platforms and so it is desirable to minimize their demand on resources to ease porting to more constrained environments. Some of the resource intensive features that have been omitted will reduce resource demands even with an ORB that supports them if the features are not used. However, to achieve the full goal of reducing demand on system resources, ORBs omitting support for these features will be required.
Because platforms may use additional features, these three “profiles” are not intended to specify complete ORBs for hosting SCA systems.

5.3 Proposed modification #2

Proposed version for Table 1:

The following “Ultra lightweight CORBA” column should be added as last column:

Ultra lightweight CORBA
PRT[1]
NRQ

5.4 Proposed modification #3

Proposed version for D.1.2.4.1 Features from CORBA/e paragraph:

The features included in the Full, Lightweight and Ultra lightweight CORBA Profiles listed in Attachment 1 shall behave as described in the applicable clauses of the CORBA/*e* specification.

For convenience, columns showing the features included in some other profiles (noted with an 'x') have been included: Minimum CORBA and CORBA/*e* compact.

5.5 Proposed modification #4

Proposed version for D.1.2.4.2 Features from RTCORBA paragraph:

This specification permits the use of a few RT CORBA features that provide useful ways to system tune performance but are not supported by the CORBA/*e* specification. The features included in the Full, Lightweight and Ultra lightweight CORBA Profiles listed in Attachment 2 shall behave as described in the applicable clauses of the RT CORBA specification.

5.6 Proposed modification #5

Proposed version for D.1.2.4.2.1 ORB_init Parameters paragraph:

The Full CORBA profile includes methods to create certain POA policies, but these methods are only supported on child POAs. The root POA has default settings for these policies that cannot be changed during the life of the root POA. The Lightweight CORBA profile does not support the creation of child POAs or calls to the policy creation methods. In some systems it is useful to use a policy other than the default even when it cannot be changed dynamically. The creation of child POAs only to allow static policies other than the default adds undesirable overhead. Therefore it is desirable to allow creating the root POA with policies other than the default.

The Full and Lightweight profiles shall support the additional standardized parameters identified in Table 2 to the ORB_init call to allow the root POA to be created with non-default policies. These additional parameters are not standardized in the CORBA/*e* specification.

For the Ultra lightweight Profile, the above considerations are not applicable, as ORB_init is not supported in this profile

The following "Ultra lightweight CORBA" column should be added as last column in Table 2:

Ultra lightweight CORBA
NRQ
NRQ
NRQ

5.7 Proposed modification #6

Proposed version for D.1.2.5 Attachments paragraph:

This appendix includes the following:

- ATTACHMENT 1. Full/Lightweight/Ultra lightweight CORBA Features (from CORBA/*e*)
- ATTACHMENT 2. Full/Lightweight/Ultra lightweight CORBA Features (from RT CORBA)

These attachments include the Full/Lightweight/Ultra lightweight CORBA Features from the CORBA/*e* and RT CORBA specifications.

5.8 Proposed modification #7

Proposed version for Appendix E.1 - Attachment 1: Full and Lightweight CORBA Profiles (basically it was added the last "Ultra lightweight CORBA" column and added the two rows "keywords: module, interface, in, out, inout, void, typedef, oneway. Return values: basic data types " and "All other keywords"):

IDL allowed in profiles

Three IDL profiles are defined:

1. Full IDL profile

2. Lightweight IDL profile (subset of the Full profile)

3. Ultra Lightweight IDL profile (subset of the Lightweight profile)

Each profile characterizes the IDL features allowed for definition of interfaces between application components.

Such IDL descriptions are enabling code generators (i.e. CORBA IDL compilers) to undertake inter components interactions, thus facilitating application components portability.

Full IDL profile provides largest capabilities in interfaces definition.

Lightweight IDL narrows those capabilities in order to limit the processing overheads caused by a number of types present in the Full profile. UltraLightWeight narrows even further in order to accommode typical limitations of DSP and FPGA environments.

Usage of reduced profiles, when specifying interfaces of a given application, enables to preserve portability of developed components from highest capable environments towards most constrained environments.

Operation/Feature	**Full IDL**	**Lightweight IDL**	**Ultra lightweight IDL**
Abstract Interfaces	NRQ	NRQ	NRQ
Value Type	NRQ	NRQ	NRQ
Any[1]	MAN	NRQ	NRQ
operation context clauses	NRQ	NRQ	NRQ
boolean, octet, short, unsigned short, long, unsigned long, enum	MAN	MAN	MAN
float, double, long double, long long, unsigned long long, char, string	MAN	MAN*	NRQ
wide character/string	NRQ	NRQ	NRQ
unions	MAN	MAN*	NRQ
arrays	MAN	MAN*	NRQ
struct[2]	MAN	MAN	MAN
sequence[2, 3]	MAN	MAN	MAN
import	NRQ	NRQ	NRQ
keywords: module, interface, in, out, inout, void, typedef, oneway. return values: basic data types	MAN	MAN	MAN
All other keywords	MAN	MAN	NRQ

Table 3 - IDL in Full, Lightweight and Ultra Lightweight Profiles

Note 1: See Section E.1.2.4.1.1 Complex Types in Any
Note 2: For the Ultra LW profile the `struct` and `sequence` types are restricted to support the basic data types.
Note 3: For the Ultra LW profile the sequence data type should be bounded.

5.9 CORBA ORB Management Functions

Note: The UltraLightWeight profile does not require any ORB management Functions

Table 4: CORBA ORB Management Functions

Module	CORBA Interface	Operation/Feature	CORBA/e Para #	Minimum CORBA	CORBA/e Compact	Full CORBA	Lightweight CORBA	Ultra lightweight CORBA
CORBA	ORB	orb_init[1]	8.3.1	x	x	MAN	MAN	NRQ
		id	8.2.1.1	x	x	NRQ	NRQ	NRQ
		object_to_string	8.2.2.1	x	x	MAN	MAN	NRQ
		string_to_object	8.2.2.2	x	x	MAN	MAN	NRQ
		get_service_information	8.2	x	x	NRQ	NRQ	NRQ
		list_initial_services	8.3.2.1	x	x	NRQ	NRQ	NRQ
		resolve_initial_references	8.3.2.2	x	x	MAN	NRQ	NRQ
		work_pending	8.2.3.1		x	MAN	MAN	NRQ
		perform_work	8.2.3.2		x	MAN	MAN	NRQ
		run	8.2.3.3	x	x	MAN	MAN	NRQ
		shutdown	8.2.3.4		x	MAN	MAN	NRQ
		destroy	8.2.3.5	x	x	MAN	MAN	NRQ
		create_policy	10.2.2.3	x	x	MAN	NRQ	NRQ
		register_value_factory	8.2, 9.3.3.3	x	x	NRQ	NRQ	NRQ
		unregister_value_factory	8.2, 9.3.3.3	x	x	NRQ	NRQ	NRQ
		lookup_value_factory	8.2, 9.3.3.3	x	x	NRQ	NRQ	NRQ
		register_initial_reference	8.3.3.1	x	x	NRQ	NRQ	NRQ

[1] Note 1: See Section E.1.2.4.2.1 ORB_init Parameters

Module	CORBA Interface	Operation/Feature	CORBA/e Para #	Minimum CORBA	CORBA/e Compact	Full CORBA	Lightweight CORBA	Ultra lightweight CORBA
CORBA	Object	get_interface	11.3.1	x	x	NRQ	NRQ	NRQ
		is_nil	9.2.3.1	x	x	MAN	NRQ	NRQ
		duplicate	9.2.2.1	x	x	MAN	NRQ	NRQ
		release	9.2.2.2	x	x	MAN	NRQ	NRQ
		is_a	9.2.4.1		x	MAN	NRQ	NRQ
		non_existent	9.2.5.1		x	MAN	NRQ	NRQ
		is_equivalent	9.2.6.2	x	x	MAN	NRQ	NRQ
		hash	9.2.6.1	x	x	NRQ	NRQ	NRQ
		get_policy	9.2.8.1	x	x	MAN	NRQ	NRQ
		set_policy_overrides	9.2.9.1	x	x	MAN	NRQ	NRQ
		get_client_policy	9.2.8.2	x	x	MAN	NRQ	NRQ
		get_policy_overrides	9.2.8.3	x	x	MAN	NRQ	NRQ
		validate_connection	9.2.10.1	x	x	MAN	MAN	NRQ
		get_orb	9.2.11.1	x	x	MAN	NRQ	NRQ
		get_component		x		NRQ	NRQ	NRQ
	Policy	policy_type	10.2.1.3	x	x	MAN	NRQ	NRQ
		copy	10.2.1.1	x	x	MAN	NRQ	NRQ
		destroy	10.2.1.2	x	x	MAN	NRQ	NRQ
	PolicyManager	get_policy_overrides	10.3.3.1		x	MAN	NRQ	NRQ
		set_policy_overrides	10.3.3.2		x	MAN	NRQ	NRQ
	TypeCode		8.5	x	x	MAN	NRQ	NRQ
	PolicyCurrent		10.3.3.2		x	MAN	NRQ	NRQ
Messaging	RebindPolicy	rebind_mode	10.4.1.2		x	NRQ	NRQ	NRQ
	SyncScopePolicy	synchronization	10.4.2		x	MAN	NRQ	NRQ
	RequestEndTimePolicy	end_time	10.4.3.1		x	NRQ	NRQ	NRQ
	ReplyEndTimePolicy	end_time	10.4.3.2		x	NRQ	NRQ	NRQ
	RelativeRequestTimeoutPolicy	relative_expiry	10.4.3.3		x	NRQ	NRQ	NRQ
	RelativeRoundtripTimeoutPolicy	relative_expiry	10.4.3.4		x	NRQ	NRQ	NRQ

Module	CORBA Interface	Operation/Feature	CORBA/e Para #	Minimum CORBA	CORBA/e Compact	Full CORBA	Lightweight CORBA	Ultra lightweight CORBA
PortableServer	LifespanPolicy	value	11.3.3.1	x	x	MAN	NRQ	NRQ
	IdUniquenessPolicy	value	11.3.3.1	x	x	NRQ	NRQ	NRQ
	IdAssignmentPolicy	value	11.3.3.1	x	x	MAN	NRQ	NRQ
	POAManager	activate	11.3.2.2	x	x	MAN	MAN	NRQ
		get_state	11.3.2.3		x	NRQ	NRQ	NRQ
		get_id	11.4.1		x	MAN	NRQ	NRQ
	POA	create_POA	11.3.4.1	x	x	MAN	NRQ	NRQ
		find_POA	11.3.4.2	x	x	MAN	NRQ	NRQ
		destroy	11.3.4.3	x	x	MAN	NRQ	NRQ
		create_lifespan_policy	11.3.5	x	x	MAN	NRQ	NRQ
		create_id_uniqueness_policy	11.3.5	x	x	MAN	NRQ	NRQ
		create_id_assignment_policy	11.3.5	x	x	MAN	NRQ	NRQ
		the_name	11.3.5.1	x	x	MAN	NRQ	NRQ
		the_parent	11.3.5.2	x	x	MAN	NRQ	NRQ
		the_POAManager	11.3.5.3	x	x	MAN	MAN	NRQ
		activate_object	11.2.5.4	x	x	MAN	MAN	NRQ
		activate_object_with_id	11.3.5.5	x	x	MAN	MAN	NRQ
		deactivate_object	11.3.5.6	x	x	MAN	MAN	NRQ
		create_reference	11.2.9	x	x	MAN	NRQ	NRQ
		create_reference_with_id	11.2.9	x	x	MAN	NRQ	NRQ
		servant_to_id	11.3.5.7	x	x	MAN	NRQ	NRQ
		servant_to_reference	11.3.5.8	x	x	MAN	NRQ	NRQ
		reference_to_servant	11.3.5.9	x	x	MAN	NRQ	NRQ
		reference_to_id	11.3.5.10	x	x	MAN	NRQ	NRQ
		id_to_servant	11.3.5.11	x	x	MAN	NRQ	NRQ
		id_to_reference	11.3.5.12	x	x	MAN	NRQ	NRQ
	Current	get_POA	11.3.6.1	x	x	MAN	NRQ	NRQ
		get_object_id	11.3.6.2	x	x	MAN	NRQ	NRQ
		get_reference	11.3.6.3		x	MAN	NRQ	NRQ
		get_servant	11.3.6.4		x	MAN	NRQ	NRQ

Module	CORBA Interface	Operation/Feature	CORBA/e Para #	Minimum CORBA	CORBA/e Compact	Full CORBA	Lightweight CORBA	Ultra lightweight CORBA
RTCORBA	**ServerProtocolPolicy**		A.4			MAN	NRQ	NRQ
	PriorityModelPolicy	CLIENT_PROPAGATED	12.7.2		x	MAN	NRQ	NRQ
	PriorityBandedConnectionPolicy	priority_bands	12.1		x	MAN	NRQ	NRQ
	Current	the_priority	12.12		x	MAN	NRQ	NRQ
	Mutex	lock	12.8		x	NRQ	NRQ	NRQ
		unlock	12.8		x	NRQ	NRQ	NRQ
		try_lock	12.8		x	NRQ	NRQ	NRQ
	RTORB	create_mutex	12.8		x	NRQ	NRQ	NRQ
		destroy_mutex	12.8		x	NRQ	NRQ	NRQ
		create_priority_model_policy	12.7.1		x	MAN	NRQ	NRQ
		create_priority_banded_connection_policy	12.10		x	MAN	NRQ	NRQ
CosNaming	**NamingContext**		13.2		x	NRQ	NRQ	NRQ
	BindingIterator		13.3		x	NRQ	NRQ	NRQ
	NamingContextExt		13.5.4		x	NRQ	NRQ	NRQ
CosEventComm	**PushConsumer**		14.1.6.1		x	MAN	NRQ	NRQ
	PushSupplier		14.1.6.1		x	MAN	NRQ	NRQ
	PullSupplier		14.1.6.2		x	NRQ	NRQ	NRQ
	PullConsumer		14.1.6.2		x	NRQ	NRQ	NRQ
CosEventChannelAdmin	**ProxyPushConsumer**		14.4		x	NRQ	NRQ	NRQ
	ProxyPushSupplier		14.4		x	NRQ	NRQ	NRQ
	ProxyPullConsumer		14.4		x	NRQ	NRQ	NRQ
	ProxyPullSupplier		14.4		x	NRQ	NRQ	NRQ
	ConsumerAdmin		14.4		x	NRQ	NRQ	NRQ
	SupplierAdmin		14.4		x	NRQ	NRQ	NRQ
	EventChannel		14.4		x	NRQ	NRQ	NRQ
LW Log Service			15		x	MAN	NRQ	NRQ

5.10 Proposed modification #8

Proposed version for Appendix E.1 - Attachment 2: Full and Lightweight CORBA Profiles (from RT CORBA)

Table 5: RT CORBA ORB Management Functions

Module	CORBA Interface	Operation/Feature	RT CORBA Para #	Full CORBA	Lightweight CORBA	Ultra Lightweight CORBA
RTCORBA	**PriorityModelPolicy**	SERVER_DECLARED	RTCORBA.idl	MAN	NRQ	NRQ
	Thread Pools	create_threadpool	2.10	MAN	NRQ	NRQ
		create_threadpool_with_lanes	2.10	MAN	NRQ	NRQ
RTPortableServer	**POA**	activate_object_with_priority	2.7.5	MAN	NRQ	NRQ