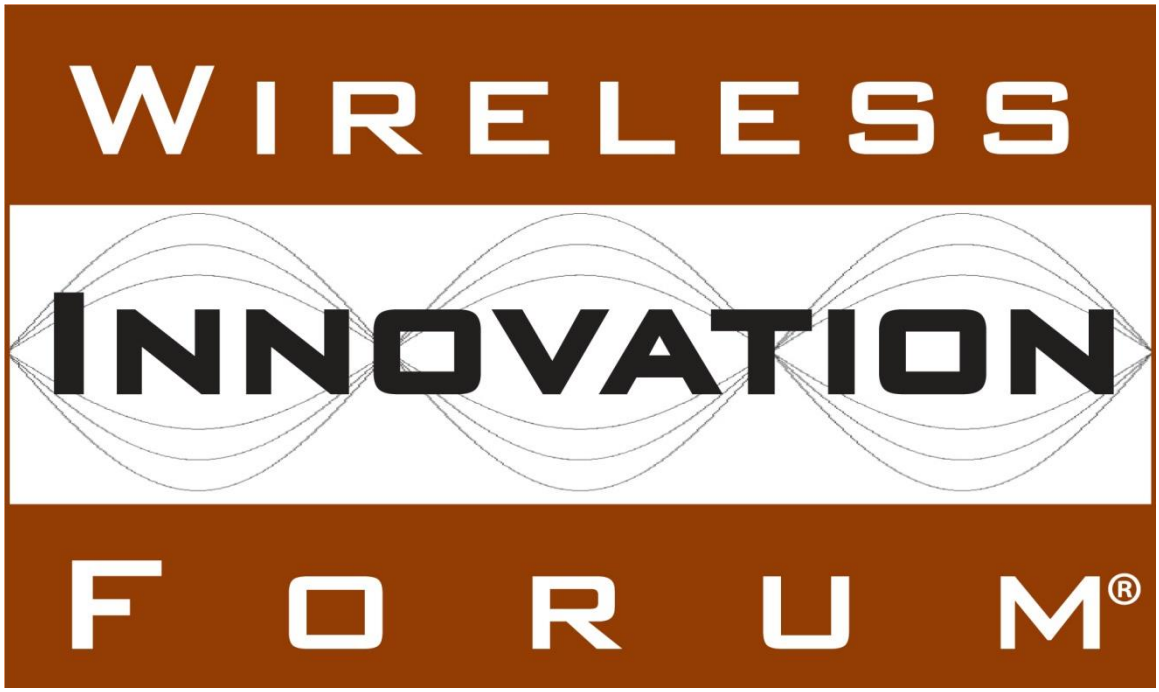

IDL Profiles for Platform-Independent Modeling of SDR Applications

Document WINNF-14-S-0016

Version V1.0.0

25 Aug 2014

TERMS, CONDITIONS & NOTICES

This document has been prepared by the work group of WInnF project SCA-2013-002 "IDL Profiles Improvements for SCA 4.1" 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 It may be amended or withdrawn at a later time and it is not binding on any member of the Forum or of the IDL Profiles Improvements for SCA 4.1 work group.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.

This document was developed following the Forum's policy on restricted or controlled information (Policy 009) to ensure that that the document can be shared openly with other member organizations around the world. Additional Information on this policy can be found here: http://www.wirelessinnovation.org/page/Policies_and_Procedures

Although this document contains no restricted or controlled information, the specific implementation of concepts contain herein may be controlled under the laws of the country of origin for that implementation. Readers are encouraged, therefore, to consult with a cognizant authority prior to any further development.

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

Table of Contents

List of Figures

List of Tables

Executive Summary

This specification addresses specification of IDL Profiles setting designers' possibility for specification of application-specific interfaces of SDR Applications Components in a PIM (Platform Independent Model) way.

It has been elaborated by the Wireless Innovation Forum as a contribution to SCA 4.1 development, and aims at providing standard solution for the broadest range of actors of the SDR eco-system.

It defines one "Full" and one "Ultra-Lightweight" / "ULw" IDL Profiles, that set the range of possible IDL declarations attached to specification of PIM interfaces.

The Full profile is maximizing the range of possible IDL declarations and types for PIM specification of application-specific interfaces, while the ULw further narrows the allowed possibilities to a minimal set that only fits the essential needs of processing-intensive embedded components, such as components of physical layers of waveform applications.

As support content, an overview of the specification is provided, by positioning content in front of the entire IDL Standard of the OMG.

Besides, a rationale section explains the applied development paradigm, and provides a detailed rationale for the choices, on a feature-by-feature basis within the IDL standard.

This specification is a solution that took into account and harmonized two publicly available specifications: the ESSOR Architecture Common IDL Profile for DSP and FPGA and the SCA 4.0.1 CORBA ULw CORBA PSM.

It is expected that this specification will be broadly used within the international SDR eco-system structured through usage of SDR standards supporting SDR Applications portability.

Contributors

The direct contributors to this specification were:

- Aeroflex, with Dave Hagood as main contributor,
- Harris Corporation, with Dave Carlson as main contributor,
- Hitachi Kokusai Electric, with Dave Murotake as main contributor,
- NordiaSoft, with François Levesque as main contributor,
- Objective Interface Solutions, with Charles Rush as main contributor,
- Prismtech, with Ted Poole as main contributor,
- Raytheon, with Jerry Bickle as main contributor,
- SELEX, with Fabio Casalino as main contributor,
- THALES Communications & Security, with Eric Nicollet as main contributor.

Direct contributors to the specification have been supported by contributors working with JTNC Standards, who provided methodological guidance in order to facilitate later exploitation of the specification in the elaboration process of SCA 4.1.

IDL Profiles for Platform-Independent Modeling of SDR Applications

1 Introduction

1.1 Aim of the specification

1.1.1 Positioning

This specification addresses definition of standard IDL (Interface Definition Language) Profiles enabling *PIM (Platform Independent Modeling) specification* of *application-specific interfaces* of SDR (Software Defined Radio) Applications.

More specifically, the specified IDL Profiles are identifying the exact scope of IDL features, taken from the complete IDL language as defined by the OMG (Object Management Group), that designers of conformant SDR Applications can use for defining *Application-Specific Interfaces* of the *SDR Application Components*.

Strictly speaking, by *Application-Specific Interfaces* are meant the interfaces of *SDR Application Components* that are **specifically defined** for the design of a SDR Application, by contrast to SCA (Software Communications Architecture) Standard Interfaces, which relate components of the application to the deployment and configuration framework, and by contrast to standard Radio Devices and/or Radio Devices APIs as shown in Figure 1.

By extension, the terminology *Application-Specific Interfaces* can be applied to Radio Device and/or Radio Service standard APIs specifically used in the design of the considered SDR Application as shown in Figure 1.

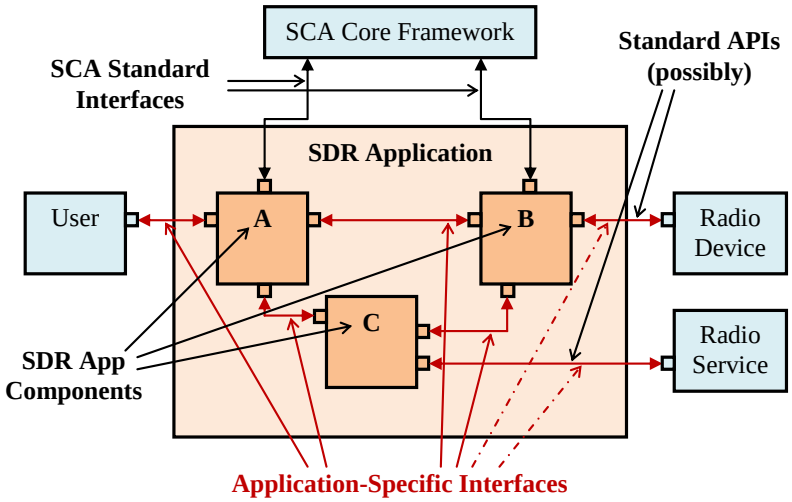

Figure 1 Notion of Application-Specific Interfaces

Compliance of SDR Application and SDR Engineering Tools with this specification provides important increase in the portability of SDR Applications.

This specification is NOT directly impacting conformance of the OE (Operating Environment).

1.1.2 PIM and PSM foundations for portability improvement

This section exposes the theoretical foundations that ground the benefits in portability provided by this specification. The state benefits for portability are specifically presented in § 1.4.4.

1.1.2.1 PIM-level and PSM-level engineering stages

By *PIM-level engineering stage* is meant an engineering stage where SDR components are considered without making assumptions on the subsequent choices required for their software implementation.

By *PSM-level (Platform Specific Modeling) engineering stages* are meant subsequent engineering stages, where implementation-related choices are made for software implementation purposes.

At PIM-level, application-specific interfaces can be defined using the PIM IDL Profiles of this specification with independence from the following PSM-level choices: (i) **Programing Language** used for implementation of the *SDR Component*, and (ii) **Connectivity** provided by the OE to implement connections between *SDR Components*.

Typical PSM-level *Programming Languages* are common languages for distributed embedded systems. C and C++ are typical of languages for instruction-set processors, such as GPPs (General Purpose Processors) and DSPs (Digital Signal Processors). VHDL and Verilog are typical examples for programmable logic processors, i.e. FPGAs (Field Programmable Gate Arrays).

Typical PSM-level *Connectivity* can be *integrated*, e.g. in CORBA or brokers cases, where Engineering Tools provide automatic code generators for Used / Provided ports that avoid developer to directly access the underlying *Transport Mechanism*, or *raw*, where the developer has to directly use *Transport Mechanism* to implement the Used / Provided ports.

Typical standard *Connectivity* useable when components are running on different processors are CORBA (see **[Ref4]**), JTNC MHAL Connectivity (see **[Ref5]**) or JTNC MOCB (see **[Ref6]**).

Typical solutions when components are running on the same processor are Inter-Process Communication (see entry-level information at **[Ref7]**).

Non-standard *Transport Mechanisms*, Commercially-Off-The-Shelf (COTS) or proprietary, may be used as well.

1.1.2.2 PIM to PSM migration

Design of PSM-level Application Specific Interfaces can be based on PIM-level Application Specific Interfaces, based upon a PIM-to-PSM migration strategy.

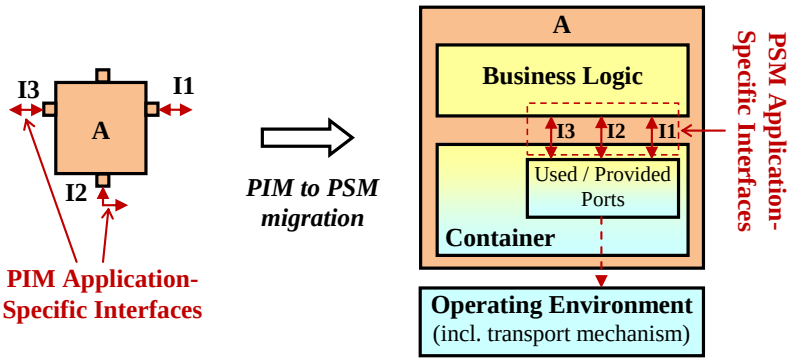

Figure 2 Notion of PIM to PSM migration

As illustrated in Figure 2, the PSM Application-Specific interfaces separate the Business Logic of a SDR component from the Used / Provided Ports of the component Container. The Used / Provided Ports take in charge the adaptation between PSM Application Specific Interfaces and the Transport Mechanism available in the chosen OE.

This enables the Business Logic to be developed with high independence from assumptions on: (i) the Transport Mechanism of the OE, and (ii) the implementation choices and localization of the SDR component to which the component of interest is connected to. This increases level of portability of the developed Business Logic software.

The level of portability of the Business Logic depends on a variety of other factors which this specification does not address, such as the PIM to PSM interface mapping, the business logic implementation and dependencies (e.g. POSIX AEPs compliant) or the complete design pattern applied between the Business Logic with Container.

Application of formal mapping rules from IDL to programming language (e.g., OMG IDL to language mapping standards) enable, when available, to entirely derive the definition of PSM Application-Specific Interfaces from the specifications established at PIM-level.

1.1.3 Elaboration context and initial objectives

The IDL Profiles addressed by this specification are primarily aimed at supporting elaboration of SCA 4.1. Besides, they are subject to be used by any interested user within the SDR eco-system.

The reference IDL specification taken as basis for elaboration of the specification is entirely and exclusively captured by “chapter 7” of the OMG CORBA v3.2 standard, as published by the Object Management Group (see **[Ref1]**). Being independent from the rest of the CORBA standard, usage IDL is therefore NOT implying usage of CORBA.

The initial elaboration approach for this specification was to improve and harmonize into a reference specification the achievements of two pre-existing publicly available specifications: the ULw CORBA Profile defined in Appendix E-1 of SCA 4.0.1 (see **[Ref2]**) and the Common IDL Profile for DSP and FPGA of the ESSOR Architecture (see **[Ref3]**).

1.2 Implementation possibilities

Conformant products can namely be *SDR Application artifacts* or *SDR Engineering Tools*.

Conforming *SDR Application artifacts* can be at any PIM or subsequent PSM stage of the development cycle: PIM modeling, interface requirement specification, software interfaces design, component source code development, executable code generation…

Conforming *SDR Engineering Tools* can support any of the PIM of subsequent PSM stage, enabling designers and developers to handle conformant SDR Application artifacts thanks to features such as modeling, code generation, code verification…

The principle criteria for evaluation of products conformance with the PIM IDL Profiles are presented in § 1.4.

1.3 Specification content

The specification defines two IDL PIM Profiles:

- The “Full” PIM IDL Profile,
- The “Ultra-Lightweight” / “ULw” PIM IDL Profile.

The two PIM IDL Profiles isolate early PIM-level design of application-specific interfaces from later PSM-level choices, giving a large set of possibilities for SDR Application porting.

The essential design achievement enabling this has been to determine which features of the IDL language were relevant for the purpose of PIM-level specification of application-specific interfaces.

The Full PIM IDL Profile is using the complete set of related features, introducing no further limitations in possible declarations or used types.

The ULw PIM IDL Profile is using a more restricted set of features, specifically defined to only suit the needs of computation intensive components, such as waveforms physical layers components.

Usage of either the Full or ULw PIM IDL Profile is possible for resource-constrained components or processors, depending on user’s development strategies.

Normative content for the two PIM IDL Profiles is in § 2, support information (content overview and perspective towards entire PIM specification of components) is provided in § 3, and rationales are provided in § 4.

1.4 Conformance

1.4.1 Applicable PIM Profile

Users of the specification have to determine the *applicable PIM Profile* for their conformant work products, taken among the specified "Full" or "ULw" PIM IDL Profiles.

1.4.2 Conformance of SDR Applications

Conformity criteria

A SDR Application is *conformant* with one *applicable IDL Profile* if it **exclusively uses**, as far as its application-specific interfaces are concerned, capabilities of the *applicable IDL Profile*.

Verification procedure

Establishment of the *verification procedure* applicable for verification of the conformity criteria is beyond the scope of this specification.

1.4.3 Conformance of Engineering Tools

Conformity criteria

An Engineering Tool is *conformant* with one *applicable IDL Profile* if it **supports production of SDR Application artifacts** that are *conformant*, as SDR Application, with the applicable *IDL Profile*.

Verification procedure

Establishment of the *verification procedure* applicable for verification of those conformity criteria is beyond the scope of this specification.

The nature of the definition above subordinates establishment of such verification procedure to existence of underlying verification procedure for SDR Applications.

1.4.4 Benefits of conformance

1.4.4.1 Conformance of SDR Applications

For a *SDR Application*, conformance to one *applicable IDL Profile* enables to design the *SDR Application* with high independence from implementation choices. Depending on the work product development context and constraints, this can brings various benefits related to portability improvements.

This enables improving design of SDR Applications aimed to be **distributed** across several processors, and is even more beneficial when considering situations where the set of processors if potentially **heterogeneous**, mixing processors of different natures and/or different programming languages.

This enables developed component having strong independency from the transport mechanisms available in the OE where the SDR component is integrated. This corresponds to better **elementary portability** of SDR components.

This enables replacing, as and when needed, one component of the SDR Application by another implementation, while maintaining overall consistency of the SDR Application design.

This enables benefiting from SDR Engineering Tool supporting the high level design and the further steps of software implementation for the SDR Applications.

1.4.4.2 Conformance of SDR Engineering Tools

Conformance of SDR Engineering Tool provides *SDR Application* designers and software developers with code generation and automated models transformation features that can bring considerable productivity gains. A number of features possibly supported by conformant SDR Engineering Tools are identified below, for illustration purpose.

A conformant tool could enforce that the **IDL specification** written by application designer for specification of the application-specific interfaces complies with one applicable PIM IDL Profile, bringing the benefits exposed in previous section to the SDR Application design.

A conformant tool could **automatically generate**, in application of standard OMG mapping rules from IDL to the programming language selected for software implementation, the **header files** associated to the interfaces that were defined at PIM level.

A conformant tool could **automatically generate** the **used / provided ports** of the container software for encapsulation of the business logic on top of the transport mechanism of the OE available on the target radio set.

A conformant tool could automatically generate the **implementation skeleton** for the business logic of the developed SDR components, based on the header files associated to the application-specific interfaces.

2 PIM IDL Profiles

2.1 Introduction

This section provides the normative content for PIM IDL Profiles.

The IDL (Interface Description Language) specification of the OMG is the reference from which the PIM IDL Profiles are defined.

The terminology "IDL specification" is used in the rest of this document to designate § 7 "IDL Syntax and Semantics" of the OMG standard CORBA v3.2 specification (see **[Ref1]**), that entirely and exclusively provides standard specification for the IDL language.

The specification defines two profiles:

- The "Full" PIM IDL Profile,
- The "Ultra-Lightweight" / "ULw" PIM IDL Profile.

Each PIM IDL Profile is defined by the subset of declarations supported within the entire set of declarations allowed by the IDL specification. Those are:

- § 7.6 Import Declaration,
- § 7.7 Module Declaration,
- § 7.8 Interface Declaration,
- § 7.9 Value Declaration,
- § 7.10 Constant Declaration,
- § 7.11 Type Declaration,
- § 7.12 Exception Declaration,
- § 7.13 Operation Declaration,
- § 7.14 Attribute Declaration,
- § 7.15 Repository Identity related Declarations,
- § 7.16 Event Declaration,
- § 7.17 Component Declaration,
- § 7.18 Home Declaration.

The chapters of the IDL specification that deal with the IDL structure are not addressed by the profiles definition. Those are:

- § 7.1 Overview,
- § 7.2 Lexical Convention,
- § 7.3 Preprocessing,
- § 7.4 IDL Grammar,
- § 7.5 IDL Specification,
- § 7.20 Names and Scoping.

Content corresponding to a feature taken out of the subset defined for the considered profile is not applicable for profile conformity. Any other features are applicable.

The CORBA-related chapter § 7.19 CORBA Module is not applicable for the defined PIM IDL Profiles.

2.2 Import Declaration

Import Declaration as specified in § 7.6 of **[Ref1]** is **NOT part** of PIM IDL Profiles.

As a consequence, the keyword ***import*** is **OUT** of the keywords set associated to the profiles.

2.3 Module Declaration

Module Declaration as specified in § 7.7 of **[Ref1]** is **fully part** of PIM IDL Profiles.

As a consequence, the keyword ***module*** is **IN** the keywords set associated to the profiles.

2.4 Interface Declaration

Interface Declaration as specified in § 7.8 of **[Ref1]** is **partially part** of PIM IDL Profiles.

The following features are **fully part** of all PIM IDL Profiles:

- § 7.8.1 Interface Header,
- § 7.8.3 Interface Body,
- § 7.8.4 Forward declaration.

As a consequence, the keyword ***interface*** is **IN** the keywords set associated to the profiles.

The following features are **fully part** of the Full PIM IDL Profile, and are **NOT part** of the ULw PIM IDL Profile:

- § 7.8.2 Interface Inheritance Specification,
- § 7.8.5 Interface Inheritance.

The following features are **NOT part** of **any** the PIM IDL Profiles:

- § 7.8.6 Abstract Interface,
- § 7.8.7 Local Interface.

As a consequence, the keywords ***abstract*** and ***local*** are **OUT** the keywords set associated to the profiles.

2.5 Value Declaration

Value Declaration as specified in § 7.9 of **[Ref1]** is **NOT part** of PIM IDL Profiles.

As a consequence, the keywords ***custom, truncatable, supports, private*** and ***public*** are **OUT** of the keywords set associated to the profiles.

2.6 Constant Declaration

Constant Declaration as specified in § 7.10 of **[Ref1]** is **fully part** of PIM IDL Profiles.

As a consequence, the keyword ***const*** is **IN** the keywords set associated to the profiles.

2.7 Type Declaration

Interface Declaration as specified in § 7.11 of **[Ref1]** is **partially part** of PIM IDL Profiles.

2.7.1 Basic Types

Basic Types of the IDL are specified in § 7.11.1 of **[Ref1]**.

The following Basic Types are **IN** the PIM IDL Profiles:

- Integer Types: ***short***, ***unsigned short***, ***long*** and ***unsigned long***,
- Boolean Type: ***boolean*** (with attached **TRUE** and **FALSE** keywords),
- Octet Type: ***octet***.

The following Basic Types are **IN** the Full PIM IDL Profile, and are **OUT** of the ULw PIM IDL Profile:

- Integer Types: ***long long*** and ***unsigned long long***.
- Floating-Point Types: ***float***, ***double*** and ***long double***.

The following Basic Types are **OUT** the PIM IDL Profiles:

- Wide Char Type: ***wchar***,
- Any Type: ***any***.

2.7.2 Constructed Types

Usage of Constructed Types (through associated keyword ***typedef***), as specified in § 7.11.2 of **[Ref1]**, is **IN** the PIM IDL Profiles.

Structures

Usage of Structures (through associated keyword ***struct***), as specified in § 7.11.2.1 of **[Ref1]**, is **IN** the Full PIM IDL Profile, and **IN, with restrictions,** the ULw PIM IDL Profile:

ULw PIM Profile restriction: the members of structures can only be of Basic Types allowed by the ULw profile.

Discriminated unions

Discriminated unions, as specified in § 7.11.2.2 of **[Ref1]**, are **IN** the Full PIM IDL Profile, and **OUT** of the ULw PIM IDL Profile.

As a consequence, the related keywords ***union***, ***switch***, ***case***, ***default*** are **IN** the Full Profile keywords set and **OUT** the ULw keywords set.

Constructed Recursive Types and IForward Declarations

Constructed Recursive Types and IForward Declarations, as specified in § 7.11.2.3 of **[Ref1]**, are **IN** the Full PIM IDL Profile, and **OUT** of the ULw PIM IDL Profile.

2.7.3 Template Types

Template Types of the IDL are specified in § 7.11.3 of **[Ref1]**.

Sequences

Sequences (with associated keyword ***sequence***), as specified in § 7.11.3.1 of **[Ref1]** are **IN** the Full PIM IDL Profile, and **IN, with restrictions,** the ULw PIM IDL Profile:

ULw PIM Profile restriction:

- Type of the sequence can only be a **valid Basic Types of the ULw Profile**, or a valid **structure** as per ULw PIM IDL limitations set beforehand (only allowed **Basic Types)**,
- The sequence has to be a **bounded sequence**.

Strings

Strings (with associated keyword ***string***), as specified in § 7.11.3.2 of **[Ref1]** are **IN** the Full PIM IDL Profile, and **OUT** of the ULw PIM IDL Profile.

Wide strings and Fixed

WStrings and Fixed (with associated keywords ***wstring*** and ***fixed***), as specified in § 7.11.3.3 and § 7.11.3.4 of **[Ref1]**, are **OUT** the PIM IDL Profiles.

2.7.4 Complex Declarator

Support of Complex Declarator (corresponding to usage of **arrays**), as specified in § 7.11.4 of **[Ref1]**, is **IN** the Full PIM IDL Profile, and **OUT** of the ULw PIM IDL Profile.

2.7.5 Native Types

Native Types (with associated keyword ***native***), as specified in § 7.11.5 of **[Ref1]**, is **OUT** the PIM IDL Profiles.

2.7.6 Deprecated Anonymous Types

Usage of anonymous types is not supported by PIM IDL Profiles, in coherence with § 7.11.6 of **[Ref1]**.

2.7.7 Object Type

Type Object is **IN** the Full PIM IDL Profile.

The Object keyword can be used as a type specification to indicate an untyped object reference. For example, it can be used as the type for an operation parameter, in which case that parameter would be able to accept a reference to an object supporting any interface.

Type Object is **OUT** of the ULw PIM IDL Profile.

2.8 Exception Declaration

Exception Declaration as specified in § 7.12 of **[Ref1]** is **fully part** of Full PIM IDL Profile, and is **NOT part** of the ULw PIM IDL Profile.

As a consequence, the keyword ***exception*** is **IN** the keywords set associated to the Full profile and **OUT** of the keywords set associated to the ULw profile.

2.9 Operation Declaration

Operation Declaration as specified in § 7.13 of **[Ref1]** is **partially part** of PIM IDL Profiles, as detailed in the following sub-sections.

2.9.1 Return values

Specification of return values is **IN** the PIM IDL Profiles.

The keyword ***void*** is **IN** the keywords set associated to the profiles.

No type restriction apply for the Full PIM IDL Profile, while restriction are defined for the ULw PIM Profile.

ULw PIM Profile restriction: return values can only be **void** or of a **Basic Type** as allowed by the ULw profile.

2.9.2 Operation Attribute and Parameter Declarations

The following features are **fully part** of all PIM IDL Profiles:

- § 7.13.1 Operation Attribute,
- § 7.13.2 Parameter Declarations.

As a consequence, the keyword ***oneway***, ***in***, ***out*** and ***inout*** are **IN** the keywords set associated to the profiles.

2.9.3 Exceptions

The following feature is **part, with restrictions,** of Full PIM IDL Profile, and is **NOT part** of the ULw PIM IDL Profile:

- § 7.13.3 Raises Expressions.

Full IDL Profile restriction: only *regular* exceptions keyword are supported, while exceptions associated to IDL attributes are NOT supported.

As a consequence, the keyword ***raises*** is **IN** the keywords of the Full Profile and **OUT** of the keywords of the ULw Profile, and keywords ***getraises*** and ***setraises*** are **OUT** of all keywords set.

2.9.4 Context

The following feature is **NOT part** of **any** the PIM IDL Profiles:

- § 7.13.4 Context Expressions.

As a consequence, the keyword ***context*** is **OUT** of the keywords set associated to the profiles.

2.10 Attribute Declaration

Attribute Declaration as specified in § 7.14 of **[Ref1]** is **NOT part** of PIM IDL Profiles.

As a consequence, the keywords ***attribute*** and ***readonly*** are **OUT** of the keywords set associated to the profiles.

2.11 Repository Identity Related Declarations

Repository Identity Related Declarations as specified in § 7.15 of **[Ref1]** are **NOT part** of PIM IDL Profiles.

As a consequence, the keywords ***typeid*** and ***typeprefix*** are **OUT** of the keywords set associated to the profiles.

2.12 Event Declaration

Event Declaration as specified in § 7.16 of **[Ref1]** is **NOT part** of PIM IDL Profiles.

As a consequence, the keywords ***eventtype*** and ***abstract*** are **OUT** of the keywords set associated to the profiles.

2.13 Component Declaration

Component Declaration as specified in § 7.17 of **[Ref1]** is **NOT part** of PIM IDL Profiles.

As a consequence, the keywords ***component***, ***supports***, ***provides***, ***uses***, ***multiple***, ***publishes***, ***emits*** and ***consumes*** are **OUT** of the keywords set associated to the profiles.

2.14 Home Declaration

Home Declaration as specified in § 7.18 of **[Ref1]** is **NOT part** of PIM IDL Profiles.

As a consequence, the keywords ***home***, ***manages***, ***primarykey***, ***factory*** and ***finder*** are **OUT** of the keywords set associated to the profiles.

3 Support information

3.1 Overview of IDL Profiles content

This section provides a comparative overview of the defined Profiles content, taking the OMG IDL standard **[Ref1]** as reference.

The status is provided according to the following convention:

- IN: the feature is IN the profile,
- OUT: the function is OUT of the profile,
- RES: the function is in the profile, with a number of REStrictions.

Table 1 Overview of supported Declarations (except Types)

IDL Feature	Associated keyword	Full Profile	ULw Profile	Diff Full/ULw
§ 7.6 Import Declaration	Import	**OUT**	**OUT**	
§ 7.7 Module Declaration	Module	**IN**	**IN**	
§ 7.8 Interface Declaration				
§ 7.8.1 Interface Header	Interface	**IN**	**IN**	
§ 7.8.2 Interface Inheritance Specification		**IN**	**OUT**	**YES**
§ 7.8.3 Interface Body		**IN**	**IN**	
§ 7.8.4 Forward declaration		**IN**	**IN**	
§ 7.8.5 Interface Inheritance		**IN**	**OUT**	**YES**
§ 7.8.6 Abstract Interface	Abstract	**OUT**	**OUT**	
§ 7.8.7 Local Interface	Local	**OUT**	**OUT**	
§ 7.9 Value Declaration	valuetype, custom, truncatable, supports, private, public	**OUT**	**OUT**	
§ 7.10 Constant Declaration	Const	**IN**	**IN**	
§ 7.12 Exception Declaration	Exception	**IN**	**OUT**	**YES**
§ 7.13 Operation Declaration				
§ 7.13 Operation Declaration	Void	**IN**	**IN**	
§ 7.13 Operation Declaration	*return values*	**IN**	**RES***	**YES**
§ 7.13.1 Operation Attribute	Oneway	**IN**	**IN**	
§ 7.13.2 Parameter Declarations	in, out, inout	**IN**	**IN**	
§ 7.13.3 Raises Expressions	Raises	**IN**	**OUT**	**YES**
§ 7.13.3 Raises Expressions	getraises, setraises	**OUT**	**OUT**	
§ 7.13.4 Context Expressions	Context	**OUT**	**OUT**	
§ 7.14 Attribute Declaration	readonly, attribute	**OUT**	**OUT**	
§ 7.15 Repository Identify Related Declaration	typeid	**OUT**	**OUT**	
§ 7.16 Event Declaration	custom, eventtype	**OUT**	**OUT**	
§ 7.17 Component Declaration	component, supports, provides, uses, emits, publishes, consumes	**OUT**	**OUT**	
§ 7.18 Home Declaration	home	**OUT**	**OUT**	

* Return values for ULw are restricted to allowed basic types of the ULw.

Table 2 Overview of supported Types Declarations

IDL Feature	Associated keyword	Full Profile	ULw Profile	Diff Full/ULw
§ 7.11 Type Declaration				
§ 7.11.1 Basic Types				
§ 7.11.1.1 Integer Types	short, unsigned short, long, unsigned long	**IN**	**IN**	
§ 7.11.1.1 Integer Types	long long, unsigned long long	**IN**	**OUT**	**YES**
§ 7.11.1.2 Floating-Point Types	float, double, long double	**IN**	**OUT**	**YES**
§ 7.11.1.3 Char Type	Char	**IN**	**OUT**	**YES**
§ 7.11.1.4 Wide Char Type	wchar	**OUT**	**OUT**	
§ 7.11.1.5 Boolean Type	boolean,TRUE, FALSE	**IN**	**IN**	
§ 7.11.1.6 Octet Type	octet	**IN**	**IN**	
§ 7.11.1.7 Any Type	any	**OUT**	**OUT**	
N/A	Object	**IN**	**OUT**	**YES**
§ 7.11.2 Constructed Types				
§ 7.11.2 Constructed Types	typdedef	**IN**	**IN**	
§ 7.11.2.1 Structures	struct	**IN**	**RES***	**YES**
§ 7.11.2.2 Discriminated Unions	Union switch, case, default	**IN**	**OUT**	**YES**
§ 7.11.2.3 Constructed Recursive Types and IForward Declarations		**IN**	**OUT**	**YES**
§ 7.11.2.4 Enumerations	enum	**IN**	**IN**	
§ 7.11.3 Template Types				
§ 7.11.3.1 Sequences	sequence	**IN**	**RES****	**YES**
§ 7.11.3.2 Strings	string	**IN**	**OUT**	**YES**
§ 7.11.3.3 WStrings	wstring	**OUT**	**OUT**	
§ 7.11.3.4 Fixed Type	fixed	**OUT**	**OUT**	
§ 7.11.4 Complex Declarator	*arrays*	**IN**	**OUT**	**YES**
§ 7.11.5 Native Types	native	**OUT**	**OUT**	
§ 7.11.6 (Deprecated) Anonymous Types		**OUT*****	**OUT*****	

* Members for ULw structures are restricted to allowed basic types of the ULw.

** ULw sequences are answering to the following restrictions:

- Types are restricted to basic types or structures as allowed by the ULw,
- Sequences have to be bounded.

*** Usage of anonymous types is not allowed by the IDL Profiles, in conformance with the deprecation stated in the referenced IDL specification.

3.2 Towards full PIM components specification

Although some declarations allowed by the IDL languages, such as Components and Attributes Declarations, enable to specify one entire SDR Application, the defined PIM IDL profiles are voluntarily limited to specification of interfaces of SDR components. Expansion of the PIM IDL Profiles to address the entire definition of SDR Applications could be considered as possible extension of this specification.

UML-based equivalent of the specified IDL PIM IDL Profiles could be considered as possible complement to this specification.

4 Rationales

4.1 Design paradigm

This section is providing general explanations concerning the approach followed when defining the profiles.

4.1.1 Definition of the Full PIM IDL

The definition of the Full PIM IDL Profile has been centered in identification of what was making a set of IDL features adapted to a "Platform-Independent" specification of application-specific interfaces, identifying the features of the IDL language having to be removed for that purpose.

Such removed features mostly belong to two families: those implying usage of too particular types (such as Any or ValueType), those related to other purposes than specification of software interfaces (such as Attributes and Components declarations).

4.1.2 Definition of the ULw PIM IDL

The main driver for design of ULw Profile has been to use a more restricted set of features, specifically defined to only suit the *essential needs* of computation intensive components, such as waveforms physical layers components.

Other considerations, such as limited penetration of some digital representations (64 bits, floating point) in embedded resource constrained processors market and such as limited availability of IDL to VHDL code generation solutions has played a role in deciding about integration or exclusion of a number of features from the ULw IDL Profile.

4.2 Design detailed rationale

This section is providing a detailed rationale for the design choices realized in accordance with the general paradigm exposed in previous section.

As per the profiles definition approach, only the declarations allowed by the IDL have been profiled in the specification effort. Other chapters of the IDL specification (see **[Ref1]**) are applicable.

On a chapter-by-chapter basis, it first addresses definition of the Full PIM Profile, and eventually explains the additional limitations provided in the ULw PIM Profile.

4.2.1 Import Declarations

Import Declaration is not directly related to specification of interfaces. The point has therefore been put OUT of the specified profile.

This said, construction of IDL files could use it on some environment, at the expense of not being capable to directly re-use the said IDL files in environments only supporting the defined Profile.

4.2.2 Module Declarations

They have been put IN the two profiles, since bringing namespace discrimination capabilities that can prove to be useful even in resource constrained environments.

4.2.3 Interfaces Declarations

Interfaces are enabling grouping of operations. The capability is therefore IN and seen as essential to enable expression of consistency between different operations.

Abstract interfaces have been put OUT since specifically tied to object-oriented programming, and therefore seen software programming as concepts possibly introduced at PSM level where refinements of the model takes place.

Local interfaces have been put OUT for similar reasons.

While *interfaces inheritance* is supported by the Full profile, the capability has been excluded from the ULw profile, in order not to over-constrain development environments attached to FPGA, while the complexity of associated interfaces is not seen as justifying the case for inheritance support.

4.2.4 Value Declarations

Values have been put OUT of the two profiles since not valuable enough a design artifacts, as far as known SDR Applications are concerned.

4.2.5 Constant Declarations

Constants have been put IN the two profiles, since a valuable and not difficult to implement feature.

4.2.6 Type Declarations

4.2.6.1 Basic types

All the possible **Basic Types** have been included in the Full Profile, except **any** and **wchar**.

The type **any** has been excluded since not specific to CORBA technology, while an important driver of the specification effort is elaboration of CORBA-agnostic solutions for specification of interfaces.

At the same time, it was well understood that the CORBA PSM profiles will likely integrate the type **any**, since it is used in definition of SCA standard interfaces where type PropertySet appears.

The **wchar** has been excluded since not a useful feature for embedded SDR Applications, in general.

ULw restrictions

The types corresponding to 64 bits architectures are OUT the ULw PIM Profile, since 64 bits architectures are not commonly available in resource constrained processors. This concerns keywords **long long** and **unsigned long long**.

For similar reasons as for 64-bits architectures, the support of floating point is OUT the ULw PIM Profile. This concerns keywords **float**, **double** and **long double**.

Type **char** is OUT since handling of characters is not a relevant feature for signal processing resource constrained SDR components.

4.2.6.2 Constructed Types

Structures

The ULw structures are limited to members with Basic Types as allowed by the Applicable Profile.

Discriminated Unions

Discriminated unions and all associated keywords **union**, **switch**, **case** and **default** are IN the Full, since quoted cases of known existing waveforms have been reported in discussions.

They are OUT of the ULw since the support can represent significant development effort on FPGA while no essential use case was reported under the perspective of constrained SDR components.

Constructed Recursive Types and IForward Declarations

The support of recursive types construction is included IN the Full Profile, since a commonly supported feature.

They are OUT of the ULw Profile since resource constrained environments are not requiring such elaborated typing capabilities.

Enumerations

The ULw structures **enumerations** are IN the two profiles.

4.2.6.3 Template Types

Sequences

No restriction of any kind applies to sequences as required by the Full IDL Profile.

For ULw, a restriction to allowed Basic Types and allowed Structures as types used in sequence is defined. This was specifically grounded by consideration of technology limits in available solutions for IDL to VHDL code generation solutions (such as FPGA ORBs), and seen as consistent limitations in light of the known usages for resource constrained components.

For ULw, another restriction applies to the nature of supported sequences, with only support of bounded sequences. This limitation was grounded by specific limitations reported in light of currently available IDL to VHDL code generation solutions (such as FPGA ORBs).

Strings

Type string is fully supported in the Full Profile.

Type **string** is OUT the ULw Profile since handling of characters strings is not a relevant feature for signal processing resource constrained SDR components.

WStrings and Fixed Types

Due to absence of reported usage in SDR Applications, the corresponding types are OUT the two Profiles.

4.2.6.4 Complex Declarator

This concept corresponds to support of ***arrays*** declaration.

It is IN the Full Profile, but is OUT the ULw due to specific technology maturity issues for FPGA code generation suites such as FPGA ORBs.

4.2.6.5 Native Types

Native types are **OUT** of the PIM IDL Profiles, since native types imply usage of programming language-specific constructs, which are specifically to be abstracted by the PIM specification.

4.2.6.6 Deprecated Anonymous Types

In coherence with the deprecation of the anonymous types indicated in **[Ref1]**, usage of anonymous types is **OUT** of the PIM IDL Profiles.

4.2.6.7 Object Type

In order to allow the PIM to support generic programming (see http://www.generic-programming.org/), the keyword **Object** is **IN** the full IDL profile.

Such advanced programming feature not answering with the definition approach of the ULw PIM IDL Profile, keyword **Object** keyword is **OUT** the ULw PIM IDL profile.

4.2.7 Exception Declarations

The support of exceptions has been included IN the Full Profile since possibility for components to react upon abnormal signaling from server components was considered as a valid general perspective, that was not specific to PSM considerations.

The capability is OUT the ULw since not natively supported in a number of popular programming languages for resource constrained environments, noticeably C and VHDL.

4.2.8 Operation Declarations

The fundamental **in**, **out** and **inout** keywords for signatures declaration are IN.

Similarly **return values** are included, with a specific limitation decided in the ULw not to overload designs of resource constrained components.

The **oneway** keyword is supported, since offering to explicitly specify execution concurrency at PIM specification level.

For consistency with the choices made regarding exceptions, **raises** in IN the Full Profile and OUT the ULw Profile.

For consistency with the choices made regarding Attributes, **getraises** are **setraises** are OUT the two Profiles.

The support of **context** capability is not present since not commonly used, and not supported by ORBs and insufficiently specified in the IDL standard.

4.2.9 Attribute Declarations

Attribute Declarations are NOT in the specified IDL Profiles, since the purpose of the specification is limited to specification of software interfaces of SDR components.

Attribute Declaration is identified as a possible part of a **future extension** of this specification that would enable to express, beyond the SDR component interfaces, the entire SDR Application structure (identification of SDR components, of their attributes and their used and provided interfaces).

4.2.10 Repository Identify Related Declarations

Repository Identify Related Declarations are NOT in the specified IDL Profiles, since related to run time access to software parts that are not related to PIM-level specification of Application-Specific interfaces.

4.2.11 Event Declarations

Event Declarations are NOT in the specified IDL Profiles, since the purpose of the specification is limited to specification of software interfaces of SDR components.

Event Declaration is identified as a possible part of a **future extension** of this specification that would enable to express, beyond the SDR component interfaces, the entire SDR Application structure (identification of SDR components, of their attributes and their used and provided interfaces).

4.2.12 Component Declarations

Component Declarations are NOT in the specified IDL Profiles, since the purpose of the specification is limited to specification of software interfaces of SDR components.

Attribute Declaration is identified as a possible part of a **future extension** of this specification that would enable to express, beyond the SDR component interfaces, the entire SDR Application structure (identification of SDR components, of their attributes and their used and provided interfaces).

4.2.13 Home Declarations

Home Declarations are NOT in the specified IDL Profiles, since they correspond to deployment-related features that are not in the scope of this specification.

5 Acronyms List

API	Application Programming Interface
CORBA	Common Object Request Broker Architecture
COTS	Commercially-Off-The-Shelf
DSP	Digital Signal Processor
ESSOR	European Secure Software Radio
FPGA	Field Programmable Gate Array
GPP	General Purpose Processor
IDL	Interface Definition Language
IPC	Inter-Process Communication
JTNC	Joint Tactical Networking Center
MHAL	Modem Hardware Abstraction Layer
MOCB	MHAL-On-Chip Bus
OE	Operating Environment
OMG	Object Management Group
ORB	Object Request Broker
SCA	Software Communications Architecture
SDR	Software Defined Radio
ULw	Ultra-lightweight
WInnF/WINNF	Wireless Innovation Forum

6 References

The URL are provided as convenience, with hyperlinks valid at date of publication of the specification.

6.1 IDL standard in CORBA

[Ref1] *Chapter 7 "IDL Syntax and Semantics"*, in *Common Object Request Broker Architecture (CORBA) Specification, Part 1: CORBA Interfaces*, © 2011 Object Management Group, Version 3.2, November 2011.

URL: http://www.omg.org/spec/CORBA/3.2/Interfaces/PDF (.pdf file).

6.2 SCA 4.0.1 Appendix E-1

[Ref2] *Software Communications Architecture Specification, Appendix E-1: Platform Specific Model (PSM) – Common Object Request Broker Architecture (CORBA)*, Joint Tactical Networking Center, Version 4.0.1, 01 October 2012.

URL: http://jtnc.mil/sca/Pages/sca1.aspx (.pdf file).

6.3 ESSOR Common Profile for DSP and FPGA

[Ref3] *Extract from ESSOR SDR Architecture relative to CORBA Profiles for SCA Next*, input document to WInnF submitted by SELEX ELSAG on behalf of ESSOR Industries, WINNF-11-I-0008.

URL: http://groups.winnforum.org/d/do/4691 (.pdf file).

6.4 CORBA

[Ref4] *Common Object Request Broker Architecture (CORBA) Specification, Part 1: CORBA Interfaces*, © 2011 Object Management Group, Version 3.2, November 2011.

URL: http://www.omg.org/spec/CORBA/3.2/Interfaces/PDF (.pdf file).

6.5 MHAL Communication Service

[Ref5] *Chapters A, B, C and D* of *Modem Hardware Abstraction Layer (MHAL) API*, Joint Tactical Networking Center, Version 3.0, 02 Oct 2013.

URL: http://jtnc.mil/sca/Pages/api1.aspx (.pdf file)

6.6 MOCB

[Ref6] *MHAL on Chip Bus API*, Joint Tactical Networking Center, Version 1.1.5, 26 June 2013.

URL: http://jtnc.mil/sca/Pages/api1.aspx (.pdf file)

6.7 Inter-Process Communication (IPC)

[Ref7] Inter-process communication, © Wikipedia

URL: http://en.wikipedia.org/wiki/Inter-process_communication (on-line article)

END OF THE DOCUMENT