Date: March 2007

PIM and PSM for Software Radio Components Specification Version 1.0

OMG Available Specification
formal/07-03-01

The PIM and PSM for Software Radio Components Specification is physically partitioned into 5 volumes:

Communication Channel and Equipment (formal/07-03-02)
Component Document Type Definitions (formal/07-03-03)
Component Framework (formal/07-03-04)
Common and Data Link Layer Facilities (formal/07-03-05)
POSIX Profiles (formal/07-03-06)

OBJECT MANAGEMENT GROUP

implemented or approved by Object Management Group, Inc., software developed using this specification may claim compliance or conformance with the specification only if the software satisfactorily completes the testing suites.

OMG's Issue Reporting Procedure

All OMG specifications are subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the Issue Reporting Form listed on the main web page *http://www.omg.org*, under Documents, Report a Bug/Issue (*http://www.omg.org/technology/agreement.htm*).

Table of Contents

Preface

About the Object Management Group

OMG

Founded in 1989, the Object Management Group, Inc. (OMG) is an open membership, not-for-profit computer industry standards consortium that produces and maintains computer industry specifications for interoperable, portable, and reusable enterprise applications in distributed, heterogeneous environments. Membership includes Information Technology vendors, end users, government agencies, and academia.

OMG member companies write, adopt, and maintain its specifications following a mature, open process. OMG's specifications implement the Model Driven Architecture® (MDA®), maximizing ROI through a full-lifecycle approach to enterprise integration that covers multiple operating systems, programming languages, middleware and networking infrastructures, and software development environments. OMG's specifications include: UML® (Unified Modeling Language™); CORBA® (Common Object Request Broker Architecture); CWM™ (Common Warehouse Metamodel); and industry-specific standards for dozens of vertical markets.

More information on the OMG is available at *http://www.omg.org/*.

OMG Specifications

As noted, OMG specifications address middleware, modeling and vertical domain frameworks. A Specifications Catalog is available from the OMG website at:

http://www.omg.org/technology/documents/spec_catalog.htm

Specifications within the Catalog are organized by the following categories:

OMG Modeling Specifications

- UML
- MOF
- XMI
- CWM
- Profile specifications.

OMG Middleware Specifications

- CORBA/IIOP
- IDL/Language Mappings
- Specialized CORBA specifications
- CORBA Component Model (CCM).

Platform Specific Model and Interface Specifications

- CORBAservices
- CORBAfacilities
- OMG Domain specifications
- OMG Embedded Intelligence specifications
- OMG Security specifications.

All of OMG's formal specifications may be downloaded without charge from our website. (Products implementing OMG specifications are available from individual suppliers.) Copies of specifications, available in PostScript and PDF format, may be obtained from the Specifications Catalog cited above or by contacting the Object Management Group, Inc. at:

OMG Headquarters
140 Kendrick Street
Building A, Suite 300
Needham, MA 02494
USA
Tel: +1-781-444-0404
Fax: +1-781-444-0320
Email: *pubs@omg.org*

Certain OMG specifications are also available as ISO standards. Please consult *http://www.iso.org*

Typographical Conventions

The type styles shown below are used in this document to distinguish programming statements from ordinary English. However, these conventions are not used in tables or section headings where no distinction is necessary.

Times/Times New Roman - 10 pt.: Standard body text

Helvetica/Arial - 10 pt. Bold: OMG Interface Definition Language (OMG IDL) and syntax elements.

Courier - 10 pt. Bold: Programming language elements.

Helvetica/Arial - 10 pt: Exceptions

Note – Terms that appear in *italics* are defined in the glossary. Italic text also represents the name of a document, specification, or other publication.

Issues

The reader is encouraged to report any technical or editing issues/problems with this specification to *http://www.omg.org/technology/agreement.htm*.

1 Scope

This specification responds to the requirements set by “Request for Proposals for a Platform Independent Model (PIM) and CORBA Platform Specific Model (PSM)” (swradio/02-06-02) of radio infrastructure facilities that can be utilized in developing waveforms, which promotes the portability of waveforms across Software Defined Radios (SDR). The terms Software Radio and Software Defined Radio are used to describe radios that are implemented with strong emphasis on software. This type of radio, which is called SWRadio in this specification.

The SWRadio specification is physically partitioned into 5 volumes: Communication Channel and Equipment, Component Document Type Definitions, Component Framework, Common and Data Link Layer Facilities, and POSIX Profiles.

The SWRadio specification defines a UML Profile for SWRadio that defines a language for modeling SWRadio elements by extending the UML language with radio domain specific definitions. This profile is defined in the Communication Channel and Equipment, and in the Component Framework volumes.

The SWRadio specification defines a set of Platform Independent Model (PIM) facilities that provides a model of SWRadio system behavior and standardized application program interfaces (APIs) as well as example component definitions that realize the provided interfaces. The PIM facilities are captured in the Component Framework, Communication Channel and Equipment, and Common and Data Link Layer Facilities volumes. These PIM Facilities are specified independently from the underlying middleware technology. UML and its extensions provided by the UML Profile for SWRadio were used for modeling a software radio system in the Facilities PIM.

This specification also provides a mechanism for transforming the elements of the PIM model into the platform specific model for CORBA IDL. This mapping definition is given in the PSM (Chapter 10).

Finally, the SWRadio specification provides different compliance points depending on the role the implementer of this specification plays. Those different roles and respective partitioning of this document is given in the Conformance section (see below).

2 Conformance

2.1 Conformance Criteria

Conformance with the OMG Software Radio specification can be achieved in multiple methods. Therefore, several separate conformance points are defined below. The conformance language references several parts of the specification:

- The PIM-to-PSM Mapping is defined in Chapter 10.
- The UML Profile for Software Radio is summarized in Chapter 8 and details are in 3.2.4 UML Profile for Component Channel and 3.2.5 UML Profile for Component Framework volume specifications.
- The Software Radio Platform Independent Model is summarized in Chapter 9 and details are in 3.2.1 Common and Data Link Layer Facilities Specification, 3.2.4 UML Profile for Component Channel and 3.2.5 UML Profile for Component Framework volume specifications.

2.2 Conformance on the Part of a Software Radio PSM

The interfaces and components as defined in section 8 and 9 of this specification are not required to be used for a given platform or application. A platform or application uses the interfaces and component definitions that meet their needs. Conformance is at the level of usage as follows:

- A PSM implementation (no matter what language) of an interface defined in this specification (3.2.1, 3.2.4, and 3.2.5) needs to be conformant to the interface definition as described in the specification.
- A PSM implementation (no matter what language) of a component defined in this specification needs to be conformant to the component definition (ports, interfaces realized, properties, etc.) as described in the specification (3.2.1, 3.2.4, and 3.2.5).

2.3 Conformance with the UML Profile for Software Radio Applications

There are two kinds of conformance with respect to the SWRadio profile: conformance on the part of a model of a specific software radio application, and conformance on the part of a software radio tool.

2.4 Conformance by a Model of a Specific Application

A UML model of a specific application either conforms to the SWRadio profile or it does not. There are no categories of this kind of conformance. Such a UML model conforms to the SWRadio profile if it satisfies all constraints imposed by the SWRadio profile package.

2.4.1 Conformance by a Software Radio Tool

2.4.1.1 Definition of Terms for Discussion of Tool Conformance

To support the discussion of conformance by a software radio tool, we define two terms: “identified subset of UML 2.0" and “all constructs defined by the SWRadio profile.”

The *identified subset of UML 2.0* for the SWRadio profile is the set of packages contained in the UML 2.0 Superstructure specification Part 1 (Structure). Part 1 includes the following packages and the transitive closure of all packages contained by these packages and of all packages upon which these packages depend:

- Classes
- Composite Structures
- Components
- Deployments

Hereafter we sometimes use the abbreviated term identified subset to refer to the identified subset of UML 2.0.

The term *all constructs defined by the SWRadio profile* is defined to mean all constructs that are part of the package's identified subset of UML 2.0, plus all extensions to that subset that the SWRadio profile defines. Thus this term includes UML constructs that are part of the identified subset but that are not extended by the SWRadio profile.

2.4.1.2 Categories of Tool Conformance

A tool is considered to be a conformant *simple modeling tool* for the SWRadio profile if it does both of the following:

- Supports expression of all constructs defined by a profile within the SWRadio profile, via UML 2.0 notation. As stated in chapter 8, the SWRadio profile consists of two profiles: Component framework and Communication Channel. A tool can be conformant to a subset of the SWRadio profile such as the 3.2.5 UML Profile for Component Framework volume specification
- Supports the UML 2.0 XMI exchange mechanism for the identified subset and for UML 2.0 profiles.

A tool is considered to be a conformant *CORBA/XML-based forward engineering tool* for the SWRadio profile if it does both of the following:

- Supports the PIM-to-PSM Mapping defined in Chapter 8.
- Produces applications that are conformant waveform applications, based on the definition of such conformance in the "Conformance on the Part of a Software Radio PSM" section above. Alternately, if a tool only produces an application skeleton, the skeleton must not make it impossible for a full application based on the skeleton to qualify as a conformant waveform application; in other words, the skeleton must be able to form the basis of a conformant waveform application.

A forward engineering tool that targets a platform technology other than CORBA/XML can legitimately claim a degree of conformance to the profile if it conforms to the Profile-to-Waveform PIM Mapping and produces applications that are conformant waveform applications, or produces application skeletons that can form the basis of conformant waveform applications. In practice this requires the definition of an alternate PIM-PSM mapping. A forward engineering tool of this nature for the platform "X" is considered to be a conformant *X-Based forward engineering tool* for the SWRadio profile.

2.5 Sample Conformance Statements

"XXX is a conformant waveform application for the CORBA/XML platform, in accordance with the OMG Software Radio specification."

"XXX is a conformant waveform application for the J2EE/XML platform, in accordance with the OMG Software Radio specification."

"XXX is a conformant software radio infrastructure for the CORBA/XML platform, in accordance with the OMG Software Radio specification."

"XXX is a conformant software radio infrastructure for the J2EE/XML platform, in accordance with the OMG Software Radio specification."

"XXX is a model of a specific waveform application. The model conforms to the UML Profile for Software Radio Waveform Applications, in accordance with the OMG Software Radio specification."

"XXX is a conformant simple modeling tool for the UML Profile for Software Radio Waveform Applications, in accordance with the OMG Software Radio specification."

"XXX is a conformant CORBA/XML-based forward engineering tool for the UML Profile for Software Radio Waveform Applications, in accordance with the OMG Software Radio specification."

"XXX is a conformant J2EE/XML-based forward engineering tool for the UML Profile for Software Radio Waveform Applications, in accordance with the OMG Software Radio specification."

3 References

3.1 Normative References

3.1.1 UML and Profile Specifications

3.1.1.1 UML Language Specification

Unified Modeling Language (UML) Superstructure Specification, Version 2.1.1
Formal OMG Specification, document number: formal/07-02-03
The Object Management Group, February 2007
[http://www.omg.org]

Unified Modeling Language (UML) Infrastructure Specification, Version 2.1.1
Formal OMG Specification, document number: formal/07-02-04
The Object Management Group, February 2007
[http://www.omg.org]

3.1.1.2 OCL Language Specification

Object Constraint Language (OCL) Specification, Version 2.0
Formal OMG Specification, document number: formal/2006-05-01
The Object Management Group, May 2006
[http://www.omg.org]

3.1.1.3 UML Profile for CORBA Specification

UML Profile for CORBA Specification, Version 1.0
Formal OMG Specification, document number: formal/2002-04-01
The Object Management Group, April 2002
[http://www.omg.org]

3.1.1.4 UML Profile for Modeling QoS and FT Characteristics and Mechanisms Specification

UML Profile for Modeling QoS and FT Characteristics and Mechanisms, Version 1.0
Formal OMG Specification, document number: formal/06-05-02
The Object Management Group, May 2006
[http://www.omg.org]

3.1.1.5 MOF 2.0/XMI Mapping Specification

Meta Object Facility (MOF) 2.0 XMI Mapping Specification, Version 2.1
Formal OMG Specification, document number: formal/05-09-01
The Object Management Group, September 2005
[http://www.omg.org]

3.1.2 CORBA Core Specifications

3.1.2.1 CORBA Specification

Common Object Request Broker (CORBA/IIOP), Version 3.0.3
Formal OMG Specification, document number: formal/2004-03-01
The Object Management Group, March 2004
[http://www.omg.org]

3.1.2.2 Real-time CORBA Specification

Real-time - CORBA Specification, Version 1.2
Formal OMG Specification, document number: formal/2005-01-04
The Object Management Group, January 2005
[http://www.omg.org]

3.1.2.3 CORBA/e Specification

CORBA/e Specification
Draft Adopted OMG Specification, document number: ptc/06-05-01
The Object Management Group, May 2006
[http://www.omg.org]

3.1.3 UML Models

3.1.3.1 UML Profile for Communication Channel

UML Profile for Communication Channel XMI File
Formal OMG document number: dtc/2006-04-10
The Object Management Group, August 2006
[http://www.omg.org]

3.1.3.2 UML Profile for Component Framework

UML Profile for Component Framework XMI File
Formal OMG document number: dtc/2006-04-09
The Object Management Group, August 2006
[http://www.omg.org]

3.1.3.3 Common and Data Link Layer Facilities PIM

Common and Data Link Layer Facilities PIM XMI File
Formal OMG document number: dtc/2006-04-11
The Object Management Group, August 2006
[http://www.omg.org]

3.1.3.4 UML Profile for Software Radio Profile

Software Radio Profile XMI File
Formal OMG document number: dtc/2006-10-02
The Object Management Group, December 2006
[http://www.omg.org]

Note: See also formal/07-03-07, a .zip archive of associated schema files.

3.2 Non-normative References

3.2.1 Common and Data Link Layer Facilities Specification

Common and Data Link Layer Facilities Specification, Version 1.0
Formal OMG document number: formal/07-03-05
The Object Management Group, March 2007
[http://www.omg.org]

3.2.2 Component Document Type Definitions Specification

Component Document Type Definitions Specification, Version 1.0
Formal OMG document number: formal/07-03-03
The Object Management Group, March 2007
[http://www.omg.org]

3.2.3 POSIX Profiles Specification

POSIX Profiles Specification, Version 1.0
Formal OMG document number: formal/07-03-06
The Object Management Group, March 2007
[http://www.omg.org]

3.2.4 UML Profile for Component Channel Specification

Communication Channel and Equipment Specification, Version 1.0
Formal OMG document number: formal/07-03-02
The Object Management Group, March 2007
[http://www.omg.org]

3.2.5 UML Profile for Component Framework Specification

Component Framework Specification, Version 1.0
Formal OMG document number: formal/07-03-04
The Object Management Group, March 2007
[http://www.omg.org]

4 Terms and Definitions

For the purposes of this document, the following terms and definitions apply.

Common Object Request Broker Architecture (CORBA)

An OMG distributed computing platform specification that is independent of implementation languages.

Component

A component can always be considered an autonomous unit within a system or subsystem. It has one or more ports, and its internals are hidden and inaccessible other than as provided by its interfaces. A component represents a modular part of a system that encapsulates its contents and whose manifestation is replaceable within its environment. A component exposes a set of ports that define the component specification in terms of provided and required interfaces. As such, a component serves as a type, whose conformance is defined by these provided and required interfaces (encompassing both their static as well as dynamic semantics).

Facility

The realization of certain functionality through a set of well defined interfaces.

Interface Definition Language (IDL)

An OMG and ISO standard language for specifying interfaces and associated data structures.

Logical Device

A software component that is an abstraction of a hardware device it represents.

Mapping

The Specification of a mechanism for transforming the elements of a model conforming to a particular metamodel into elements of another model that conforms to another (possibly the same) metamodel.

Metadata

The Data that represents models. For example, a UML model; a CORBA object model expressed in IDL; and a relational database schema expressed using CWM.

Metamodel

A model of models.

Meta Object Facility (MOF)

An OMG standard, closely related to UML, that enables metadata management and language definition.

Model

A formal specification of the function, structure and/or behavior of an application or system.

Model Driven Architecture (MDA)

An approach to IT system specification that separates the specification of functionality from the specification of the implementation of that functionality on a specific technology platform.

Platform

A set of subsystems/technologies that provide a coherent set of functionality through interfaces and specified usage patterns that any subsystem that depends on the platform can use without concern for the details of how the functionality provided by the platform is implemented.

Platform Independent Model (PIM)

A model of a subsystem that contains no information specific to the platform, or the technology that is used to realize it.

Platform Specific Model (PSM)

A model of a subsystem that includes information about the specific technology that is used in the realization of it on a specific platform, and hence possibly contains elements that are specific to the platform.

Radio Platform

The Radio Platform is made of a Hardware Platform and a Software Platform.

Radio Set

A single radio set unit that can be ground fixed, mounted on a mobile platform or held by hand.

Radio System

A networked set of radio sets that provide wireless communication facilities between callers and callees.

Request for Proposal (RFP)

A document requesting OMG members to submit proposals to the OMG's Technology Committee. Such proposals must be received by a certain deadline and are evaluated by the issuing task force.

Service

A set of functionality with common characteristics.

Unified Modeling Language (UML)

An OMG standard language for specifying the structure and behavior of systems. The standard defines an abstract syntax and a graphical concrete syntax.

UML Profile

A standardized set of extensions and constraints that tailors UML to particular use.

5 Symbols and abbreviated terms

Abbreviation	Definition
API	Application Program Interface
BIT	Built-In Test
BSP	Burst Schedule Packets
COMSEC	Communication Security
CORBA	Common Object Request Broker Architecture
COTS	Commercial Off the Shelf
CPU	Central Processing Unit
DLPI	Data Link Protocol Interface
DSP	Digital Signal Processor
FPGA	Field Programmable Gate Array
GIOP	General Inter-ORB Protocol
GPP	General Purpose Processor
GPRS	General Packet radio Services
GPS	Global Positioning System
GSM	Global System for Mobiles
HCI	Human-Computer Interface
HW	Hardware
I/O	Input/Output
ID	Identification, Identifier
IDL	Interface Definition Language
IIOP	Internet Inter-ORB Protocol
INFOSEC	Information Security
IOR	Interoperable Object Reference
IP	Internet Protocol
ISO	International Standards Organization

LAPx	Link Access Protocol x (where x represents 1 of several protocols defined by industry
MAC	Medium Access Control, a sublayer of the OSI Data Link Layer
N/A	Not Applicable
NAPI	Networking Application Programming Interface
OE	Operating Environment
OMG	Object Management Group
ORB	Object Request Broker
OS	Operating System
OSI	Open System Interconnection
PIM	Platform Independent Model
POSIX	Portable Operating System Interface
PPP	Point-to-Point Protocol
PSE52	Real-time Controller System Profile, defined in IEEE Std. 1003.13
PSM	Platform Specific Model
QoS	Quality of Service
RAM	Random Access Memory
RF	Radio Frequency
SDR	Software Defined Radio
SW	Software
TCP	Transmission Control Protocol
TOD	Time Of Day
TRANSEC	Transmission Security
UML	Unified Modeling Language
USB	Universal Serial Bus
UMTS	Universal Mobile Telecommunications System
XML	eXtensible Markup Language

6 Additional Information

6.1 Relationship to Existing OMG Specifications

- Bibliographic Query Service (BQS) - This specification uses attributes defined in BQS classes in order to describe publications. The relevant specification is available as OMG documents: formal/02-05-03, formal/02-05-04, and dtc/02-02-01.

- Life Sciences Identifiers (LSID) - There are many cross references in the model defined by this specification. They are expressed using LSIDs. The relevant specification is available as OMG documents: formal/04-12-01 and dtc/04-08-03.
- Genomic Map (GM) - The definitions in GM specification are either too limiting or too generic for SNPs purposes. The GM was designed at the time when genetic maps were the only genome wide maps. Current focus is on sequence maps.
- Model Driven Architecture (MDA) - MDA is used as a fundamental concept. Both platform-independent (PIM) and platform-specific model (PSM) are defined here.
- XML Interdata Interchange (XMI) - The platform-independent model for this specification for designed as a UML model that was converted into XMI model exchange format as defined in the XML Interdata Interchange specification.

6.2 Acknowledgements

The following organizations (listed in alphabetical order) contributed to this specification:

- BAE Systems
- The Boeing Company
- Blue Collar Objects
- Carleton University
- Communications Research Center Canada
- David Frankel Consulting
- École de Technologie Supérieure
- General Dynamics Decision Systems
- Harris
- ISR Technologies
- ITT Aerospace/Communications Division
- L-3 Communications Corporation
- Mercury Computer Systems
- The MITRE Corporation
- Mobile Smarts
- Northrup Grumman
- PrismTech
- Raytheon Corporation
- Rockwell Collins
- SCA Technica
- Space Coast Communication Systems
- Spectrum Signal Processing
- THALES
- Virginia Tech University
- Zeligsoft
- 88solutions

7 Introduction to SWRadio

7.1 Introduction

The terms Software Radio and Software Defined Radio (SDR) are used to describe radios that are implemented with strong emphasis on software. This type of radio, which is called SWRadio in this specification, offers important technical and commercial advantages. This non-normative section gives an overview of the rationale and architecture of software radios.

A Software Defined Radio is a wireless communication system (low-capability mobile phones to high-capability multi-channel radios), in which the particular communication and transmission characteristics are realized through specialized software running on flexible signal processing hardware. This is very different from the traditional approach of using specialized hardware and has the benefit of instant reuse or sharing of a single system platform for multiple communication purposes. Within the physical limits of the underlying hardware, virtually any communication task can be realized instantaneously through a software load, including the ability of extensive field-upgrades and maintenance.

SWRadio technology is changing every facet of communication system design and usage. SDR is not just another way to build radios with the same functions, but SDR designs support many new critical needs. SDR supports several waveforms inside the same box, eases bug fixing, enables reconfigurability, allows for digitized, IP-based, data transmissions, and improves security. It also enables an open market where waveform providers can be independent of platform providers. These improvements are essential for the radio military market but also meet the current and emerging needs of the civil radio market.

Enabling cost-effective technology insertion is a strong motivation for manufacturers suddenly faced with a more volatile market than in the past and where tomorrow's standards are unclear. The life cycle of new radio sets has become so short that the return-on investments cannot be ensured. Enabling cost-effective technology insertion is also a concern for many operators and for customers faced with exploding costs. Reconfigurability is a key feature for next generation radio systems. It involves adding, removing, and modifying radio functionality.

SWRadio supports multiple concurrent waveforms inside a single radio set is critical for the military market where numerous waveforms are used by warfighters from various services and countries. Civil market changes, and future civil radios, are likely to support both a cellular waveform and a high bandwidth local waveform on the same hardware. Moreover, future radio node equipment may have to concurrently support multiple cellular waveforms such as Global System for Mobiles (GSM), General Packet Radio Services (GPRS), Universal Mobile Telecommunications System (UMTS), and high throughput waveforms such as Bluetooth and WiFi.

SWRadio facilitates repair of system defects through over-the-air or over-the-wire reprogramming of software features. Repairing system defects is an important issue for radio manufacturers since this may involve returning thousands of radio sets to factories. Software downloads to fix bugs is a key need for radios. This need is being addressed on a waveform-by-waveform basis (3GPP), but this does not solve the problem for future multi-waveform radios. SWRadio technology addresses this need through defining standard component interfaces and defining a plug & play like deployment mechanism.

For the last century, radio was mainly used to transmit human voice in the commercial sector. In the future, these radios will transmit digitized data as well as analog voice. This is a major shift. For example, a radio that was mainly focused on point-to-point links will have to support new services such as networking and be used to manage the seamless qualities of radio function challenges. Radio is sometimes viewed as being the last hop inside a network, but it is likely to also be used as a flexible wireless backbone in more and more cases.

Radio security has changed. Security functions cannot be frozen for an entire radio system life. Security functions must be able to evolve to counter new and evolving threats and to keep the security chain safe. This is especially important when coupling radios with information systems becomes the norm.

In addition to the functional improvements described above, SDR targets also bring all those new functions within an architecture that supports a cost-effective engineering approach. Software Defined Radio will allow a waveform designer to provide the application that can run on a platform designed by a different vendor. In order to do this effectively the interface between the platform and the application must be well defined and published. This specification is intended to provide that definition.

Hereafter, we discuss four main elements that impacted the development of this specification: Software Communications Architecture, Model Driven Architecture, Platform, and Waveform definitions, and the SWRadio Architecture.

7.2 Software Communication Architecture

The Software Communication Architecture (SCA) is the software architecture developed by the US Military's Joint Tactical Radio System (JTRS) Joint Program Office (JPO) for the next generation of radio systems. SDR companies are currently developing radio systems based on this architecture. It is considered as the de-facto standard in the SDR industry. The SCA forms the basis for the development of this specification.

SCA has been published to meet the requirements for radios that will operate in multiple domains and frequency bands. SCA compliant radios shall be able to communicate with legacy systems to minimize the impact of platform integration. The architecture enables technology insertion, so that new technologies can be incorporated to improve performance, and future-proof radios can be built.

Like most other software architectures, the SCA allows for the maximum possible reuse of software components. The components support plug-and-play behavior with applications being portable from one radio platform to the next.JTRS radios support legacy network protocols, for the purpose of seamless integration. The architecture supports wideband networking capabilities for voice, data and video.

The SCA defines an Operating Environment (OE) that will be used by JTRS radios. It also specifies the services and interfaces that the applications use from the environment. The interfaces are defined in CORBA IDL, and graphically represented in UML. The OE consists of a Core Framework (CF), a CORBA middleware, and a POSIX-based Operating System (OS). The OS running the SCA must provide services and interfaces that are defined as mandatory in the Application Environment Profile (AEP) of the SCA. The CF describes the interfaces, their purposes, and their operations. It provides an abstraction of the underlying software and hardware layers for software application developers. An SCA compatible system must implement these interfaces. The interfaces are grouped as Base Application Interfaces, Framework Control Interfaces, and Framework Services Interfaces.

The CF uses a Domain Profile to describe the component metadata in the system. The Domain Profile is a set of XML files that describe the identity, capabilities, properties, inter-dependencies, and location of the hardware devices and software components that make up the system.

7.3 Model Driven Architecture

The OMG Model Driven Architecture (MDA) defines a model-based development approach to software development. The main objective of MDA is to enable the portability/reuse of models across different technology platforms. MDA focuses on the definition of Platform Independent Model (PIM), Platform Specific Model (PSM), and Model Mappings that allows moving from one model to another in a systematic manner. One goal of MDA is to define a set of Model

Mappings between standard technologies that can be reused in different contexts. One of the main benefits of MDA is that models can be defined independently of specific implementation platforms and mapped to different platforms using predefined mappings.

The current SWRadio specification fully endorses the MDA approach. It defines a UML Profile for SWRadio, a PIM, and a CORBA/XML PSM for SWRadio components. The UML Profile for SWRadio, which is defined as an extension of the UML 2.0 specification, defines a set of standard stereotypes that can be used for the development of SWRadio applications, infrastructure, and deployment platforms. This profile is used in the current specification to define the PIM and PSM. The Platform Independent Model (PIM) formally defines a set of standard facilities for SWRadio without the technical details of any specific implementation. The Platform Specific Model (PSM) defines a version of the specification that is based on CORBA and XML specific technologies. The mapping between the PIM and the CORBA/XML PSM is captured in PIM-to-PSM mappings. This PIM-to-PSM mapping can be automated, which would allow automatically updating the PSM to reflect changes made in the PIM to maintain a complete consistency between the two models.

7.4 SWRadio Platform and Applications

This specification supports a SWRadio Platform/Application approach where:

- the SWRadio Platform provides a **standardized** yet **extensible** set of software services that abstracts hardware dependencies and support waveforms as well as other application types such as management applications. This specification defines a set of Platform-Independent Interfaces and does not make any assumptions on how those interfaces are supported.
- Applications can be developed and cross ported onto various Platform implementations.

Such a SWRadio Platform/Application approach opens the way to an open market where applications providers can be independent of platform providers.

The SWRadio Platform concept used here refers to a composite infrastructure that is intended to support a set of applications to build various dedicated configurations such as radio nodes, radio terminals, and/or other radio gateways. As a matter of fact, the SWRadio Platform defines a basis for a product line approach.

The SWRadio Platform concept extends the Platform concept used within MDA in the way that SWRadio Platform not only refers to a software API but also includes a set of hardware and software components.

A Radioset based on a SWRadio Platform can be seen as made of several layers. From bottom to top, layers are:

- Hardware layer: set of heterogeneous hardware resources including general purpose devices as well as specialized ones,
- Operating Environment layer: basically provides operating system and (distributed) middleware services,
- Facilities layer: provides sets of services to the application developer,
- Application layer: figures a standalone capability provided by the radioset.

Those layers are figured below:

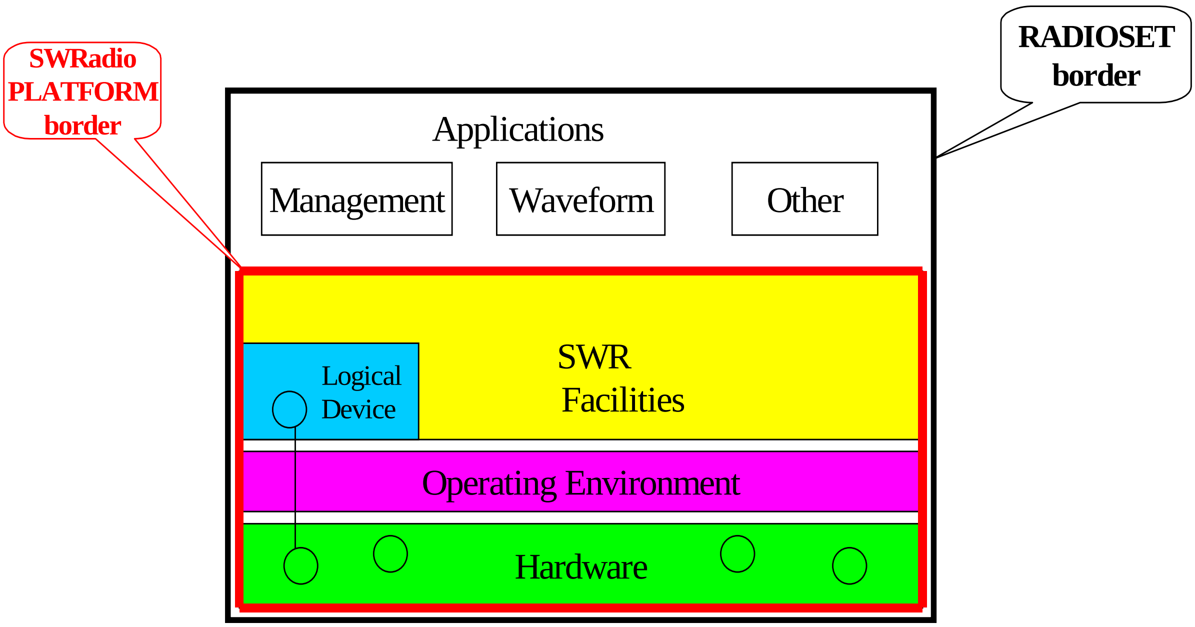

Figure 7.1 - Software Radio Layered View

Applications supported by SWRadio Platform can be dispatched into 3 categories:

- Waveform Applications that are the main focus of SWRadio and figure the waveform-specific application functions that noticeably coordinate the underlying SWRadio Platform functions to achieve the end-to-end waveform processing. These Applications also support general purpose Management Applications with waveform-specific management functions.
- Management Applications that figure general purpose, waveform-independent applications that enable to manage and control the Radioset and its embedded applications. Management Applications act as managers as defined in the OSI management framework (see IS 7498) and see Waveform Applications as agents to relay their requests. SWRadio Platform provides the management services excluding the presentation HMI.
- Other Applications figure all other kind of applications that can be provided inside a radioset such as:
  - Network applications that mainly support routing, security, directory, or QoS functions.
  - End user applications such as Situation Awareness and/or other 3rd party applications.

Applications are supported by SWRadio Facilities inside which Logical Devices abstract some of the actual hardware devices of the SWRadio Platform. Logical Devices are defined for management purpose and their properties are designed to support management functions such as (re)configuration, performance, or fault management.

Logical Devices should not be used directly by applications. Instead, applications should use higher level SWRadio Facilities.

7.5 SWRadio Architecture

The SWRadio architecture consists of two main concepts: services, as well as applications and layering. Services concept depends on the interfaces provided and the usage of those interfaces. Application layering provides a logical grouping of functionality based on current commercial practice.

Through realization relationships in the PIM, a component can offer one or more services. A SWRadio vendor may choose to provide certain services that are required for their platform, and likewise acquire extra services from third party vendors. The services that can be provided by different actors that use this specification are detailed in Chapter 2 - Conformance.

- For a logical grouping of functionality, this specification follows the Open System Interconnection (OSI) Model elaborated and promoted by the International Standard Organization (ISO).

A full description of the OSI model can be found inside the ISO IS 7498The OSI model assumes that the structure of the communications functions located on a network node should be structured into a stack of 7 Layers where:

- a layer talks with its counterpart located on another radio set,
- the communication between peer layers is ruled by a Protocol which exchanges Protocol Data Units (PDU),
- a layer supplies Services to upper layers through Service Access Points (SAP),
- a layer acts as a PDU consumer and/or provider for its upper and lower layers.

Note that some communication links do not always require all layers and that some layers may be empty inside a communications stack. The core of the OSI could be modeled as in Figure 7.2. In this figure, a sublayer is a subdivision of a layer.

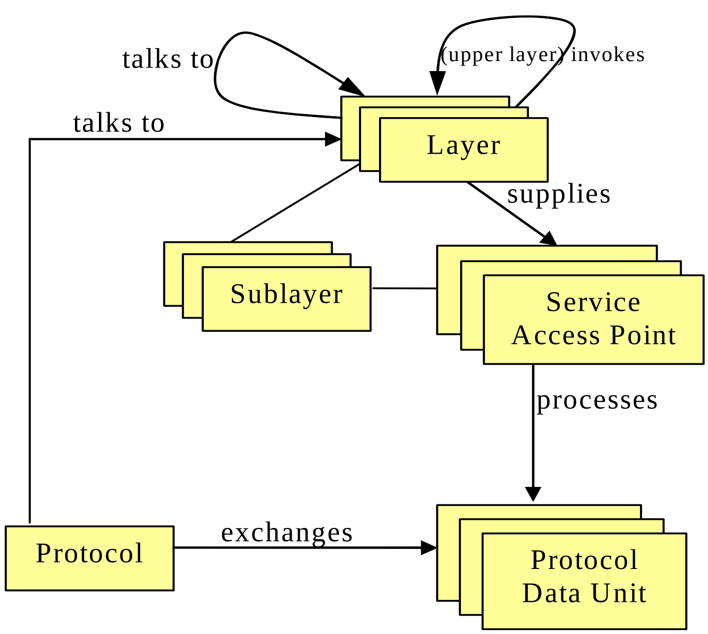

Figure 7.2 - Abstract OSI Model Core

Ironically, and despite its importance, the OSI model has primarily been a conceptual model that was implemented in hybrid manners in practice for performance reasons. Moreover, with the worldwide adoption of IP, the “old” layered model on which IP is built superseded the OSI model. Within the SDR context, the OSI model still proves to be a good design technique since it allows separation of concerns by making use of layering. This approach promotes the usage of interoperable and reconfigurable components through standard interfaces and well-defined packaging.

This specification acknowledges that the OSI communication model is a good reference design for any communication system, but conformance with this model is not mandatory for any radio set due to design and performance constraints. The proposed architecture supports not only the OSI model, but also other in-use or next-to-come models.

The OSI concepts described in this specification apply to the Extended OSI model, which allows Management and QoS interfaces to cut through the waveform layer stack and communicate with any layer. Furthermore, this specification only focuses on physical and link (link layer control and medium access control) layers of the OSI stack.

8 UML Profile for Software Radio

This non-normative section defines the UML Profile for SWRadio. This profile is an integral part of the “PIM and PSM for SWRADIO Components” as described in reference 3.1.3.4. The set of stereotypes defined in this profile constitutes the core language for the definition of the SWRadio PIM and PSM. The current UML Profile for SWRadio extends the UML 2.0 meta-language, with emphasis on extensions to. It mainly extends the Components package and Deployments package of UML 2.0.

The goal of the UML Profile for SWRadio is to enable the development of UML tools to support the development of SWRadio applications and systems. The objectives are not only to facilitate the modeling of SWRadio applications and systems, but also to enable the automatic generation of descriptor files (e.g., XML descriptor files) and code (or code skeletons) from UML models, to enable validation at design time, and to enable the development of simulation environment for SWRadio.

To address the issues of the different actors involved in SWRadio product developments, the current profile has been developed with three main viewpoints in mind: the viewpoint of application and device developers, the viewpoint of infrastructure/middleware providers, and the viewpoint of SWRadio platforms providers. These three viewpoints define distinct sets of concepts (and stereotypes) that are required in different contexts.

The UML Profile for SWRadio is made up of other profiles, which are the UML Profile for Component Framework (as described in reference 3.1.1.4) and UML Profile for Communication Channel (as described in reference 3.1.1.5). To be consistent with the three viewpoints introduced above, the UML Profile for SWRadio is partitioned in four main packages: the UML Profile for Component Framework’s Applications and Devices package, the UML Profile for Component Framework’s Infrastructure package, and the UML Profile for Communication Channel’s Communication Channel and Equipment packages. Each package defines the set of concepts and UML stereotypes required to perform a specific role in the development of an SWRadio product.

The UML Profile for Component Framework’s Applications and Devices package defines the set of concepts that are required to develop SWRadio applications and devices. This package mainly contains a set of stereotypes that extends the UML 2.0 meta-classes Component and Interface. This set of stereotypes includes Resource, Device, and Application components.

One of the main objectives of this profile is to standardize interfaces and components to enable Commercial-off-the-Shelf (COTS) component SWRadio application development.

The UML Profile for Component Framework’s Infrastructure package defines the concepts that are required to develop software components deploy services and applications (e.g., waveforms) within a radio infrastructure for SWRadio applications, and to manage the radio's domain, services, and devices. This package mainly contains a set of stereotypes that extends the UML 2.0 meta-classes Component and Interface. This set of stereotypes includes RadioManager, DeviceManager, Application, and ApplicationFactory components.

The Communication Equipment package (as described in reference 3.1.1.5) defines the concepts that are required to model SWRadio equipment. This package defines stereotypes for the different types of hardware devices used in SWRadio. This package mainly contains a set of stereotypes that extends the UML 2.0 meta-class Device. This set of stereotypes includes RF Device, I/O Device, Security Device, Antenna, Amplifier, Frequency Converter, etc. For each Device stereotype specific characteristics are defined that are required by a waveform component for deployment behavior.

The UML Profile for SWRadio uses the concept of a Port as defined in the UML 2.0 specification by extending the Port definition for two different purposes. In UML Profile for Component Framework, port stereotypes such as ServicePort, StreamPort are defined as software component ports which enable their users to access the associated software interfaces. In UML Profile for Communication Channel, the concept of a hardware port in a RadioSet environment is introduced, and Port specializations such as AnalogInputPort and DigitalPort are specified.

9 Platform Independent Model

The SWRadio PIM specified in this section represents a set of interfaces that may be appropriate for building SWRadio components or used by SWRadio components. The SWRadio Facilities may be realized using many technologies (e.g., CORBA, java, etc.).

The SWRADIO PIM Facilities are made of:

- Component Framework Facilities – The set of interfaces that all components (regardless of any layering) within the radio can realize as described in reference 3.2.5. Examples of these types of interfaces are waveform, device, and platform management interfaces.
- Common Layer Facilities – The set of interfaces that all components (regardless of any layering) within the radio can realize as described in reference 3.2.1. Examples of these types of interfaces are flow control, packet, and stream interfaces.
- Common Radio Facilities - The set of interfaces that all components within the radio can use. Examples of these types of facilities are log, naming, and event service as described in reference 3.2.1.
- Data Link Layer Facilities – The set of interfaces that compose the Link Layer Control (LLC) and Media Access Control (MAC) layer functionality for communication needs as described in reference 3.2.1.
- Physical Layer Facilities – The set of interfaces that define the functionality to convert the digitized signal into a propagating RF wave, and conversely, to convert a propagating RF wave into a digitized signal for processing as described in reference 3.2.4. The facilities also include frequency tuning, filters, interface cancellation, analog digital conversion, up/down conversion, gain control, synthesizer etc., functionality. Physical layer facilities also include functionality for baseband I/O such as serial and audio devices.
- Radio Control Facilities – The set of interfaces that define the functionality to manage the radio domain and channels within the radio as described in reference 3.2.4.

10 Platform Specific Model

The SWRadio PSMs are CORBA interfaces and XML. The PIM to CORBA interfaces are described in the Communication Channel and Equipment, Commponent Framework, and Data Link and Physical Layer Facilities volumes. Basically the same rule set is used for the PIM transformations to CORBA interface but different CORBA modules captures the transformation results. The PIM to XML is described in the Component Framework and Component Document Type Definitions volumes.

INDEX