OMG Software Radio Specification and the SCA

Jerry Bickle
Raytheon
+1-260-429-6280
Gerald_L_Bickle(at)Raytheon.com

Kevin Richardson
MITRE
+1-703-883-3300
Kwrich(at)mitre.org

Contents

- OMG SWRadio Specification Overview
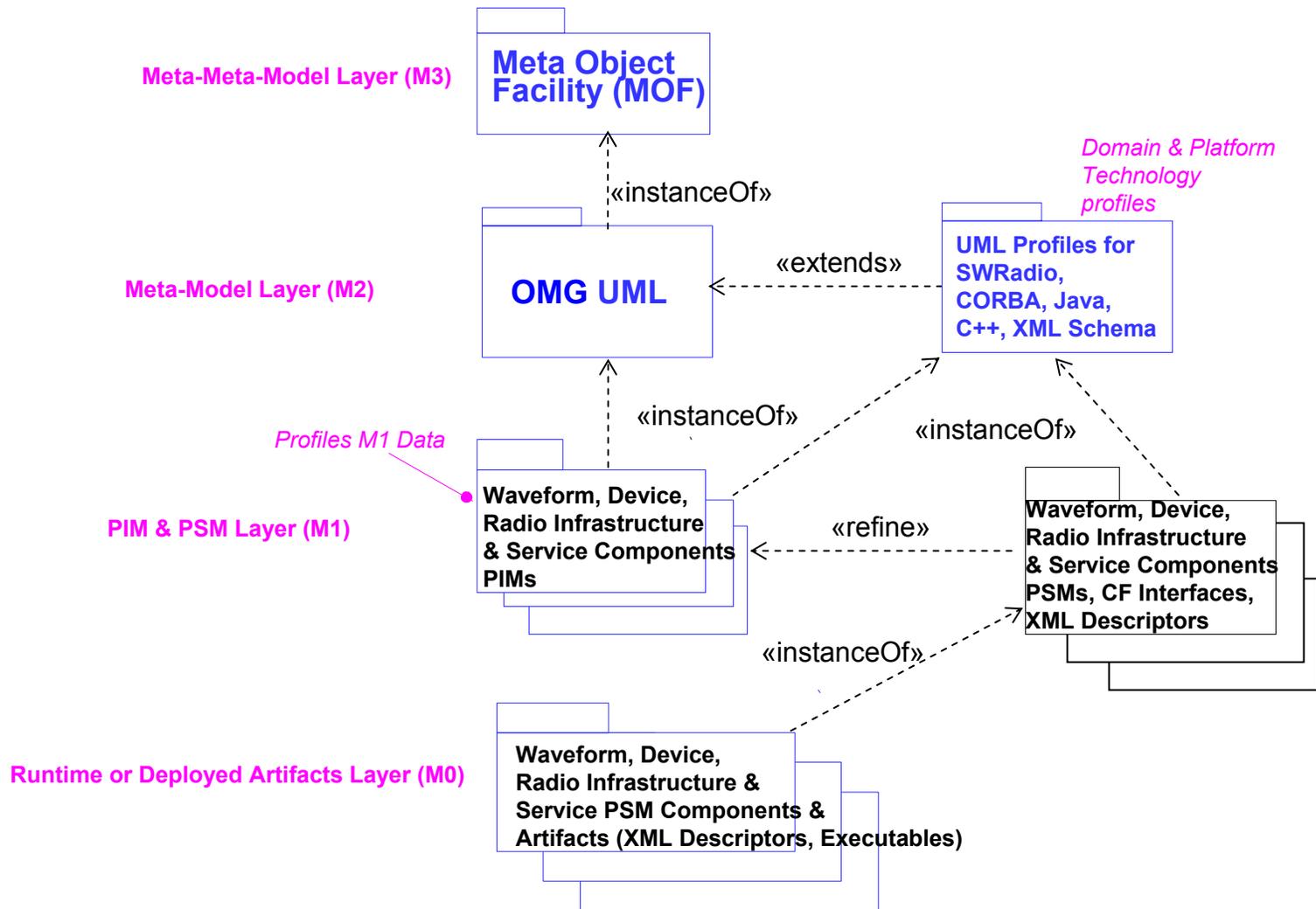- SCA 2.2.1 Comparison

OMG SWRadio Specification Overview

- SWRadio MDA Principles

- SWRadio Development Viewpoints

- UML Profile for SWRadio

- SWRadio PIM Facilities

- SWRadio PSM

SWRadio MDA Principles

- UML Profile for SWRadio extends UML for SWRadio tool support: validation, system engineering, and SWRadio component development

- PIM has been primarily structured as a set of facilities each addressing a key aspect of SWRadio

- Well-defined set of modeling conventions
  - Naming conventions
  - Modeling conventions
  - Subset of UML notation
  - Specific semantics of this notation in the context of this PIM

- Conforms to MDA
  - PIM can be transformed to different component platforms
    - CORBA-PSM, Java-PSM, etc.

- Compatible with existing OMG standards
  - MOF
  - UML

SWRadio MDA Principles, cont'd

Meta-Meta-Model Layer (M3)

Meta Object Facility (MOF)

Domain & Platform Technology profiles

«instanceOf»

Meta-Model Layer (M2)

OMG UML

«extends»

UML Profiles for SWRadio, CORBA, Java, C++, XML Schema

«instanceOf»

«instanceOf»

Profiles M1 Data

PIM & PSM Layer (M1)

Waveform, Device, Radio Infrastructure & Service Components PIMs

«refine»

Waveform, Device, Radio Infrastructure & Service Components PSMs, CF Interfaces, XML Descriptors

«instanceOf»

Runtime or Deployed Artifacts Layer (M0)

Waveform, Device, Radio Infrastructure & Service PSM Components & Artifacts (XML Descriptors, Executables)

OMG SBC Workshop, Sept 2004
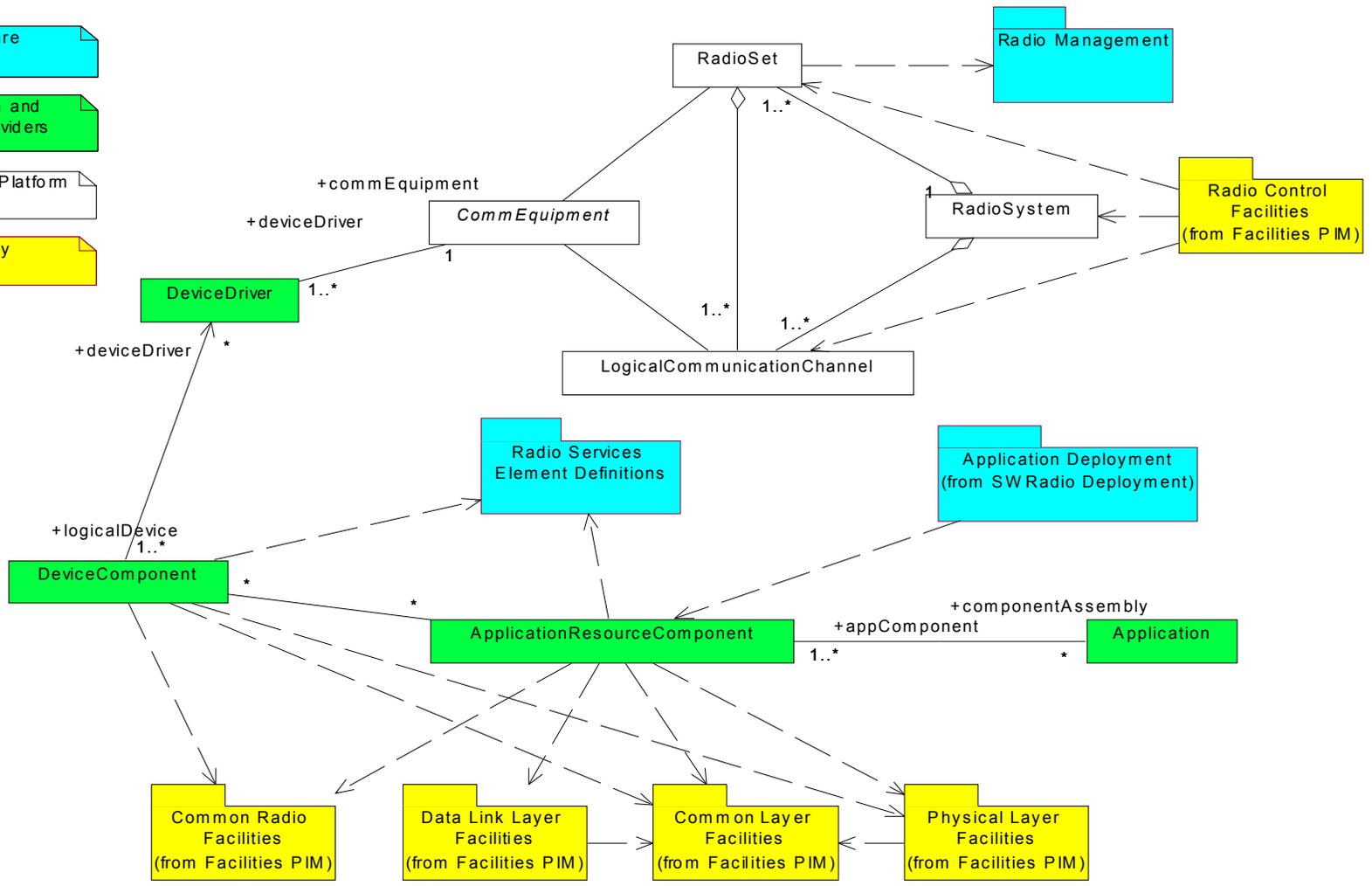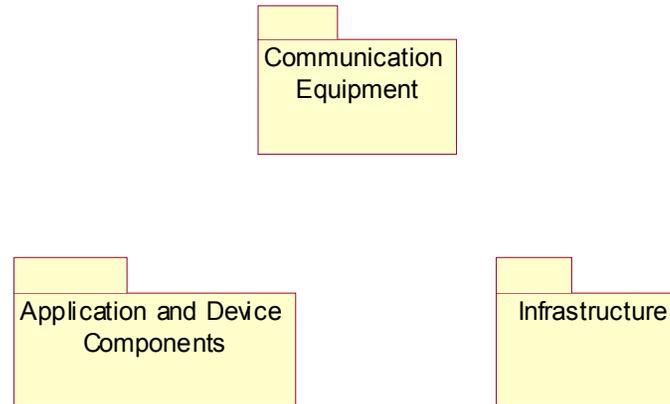
SWRadio Development Viewpoints

- To address the issues of the different actors involved in SWRadio product developments, the current profile was 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.

7

UML Profile for SWRadio

Communication Equipment
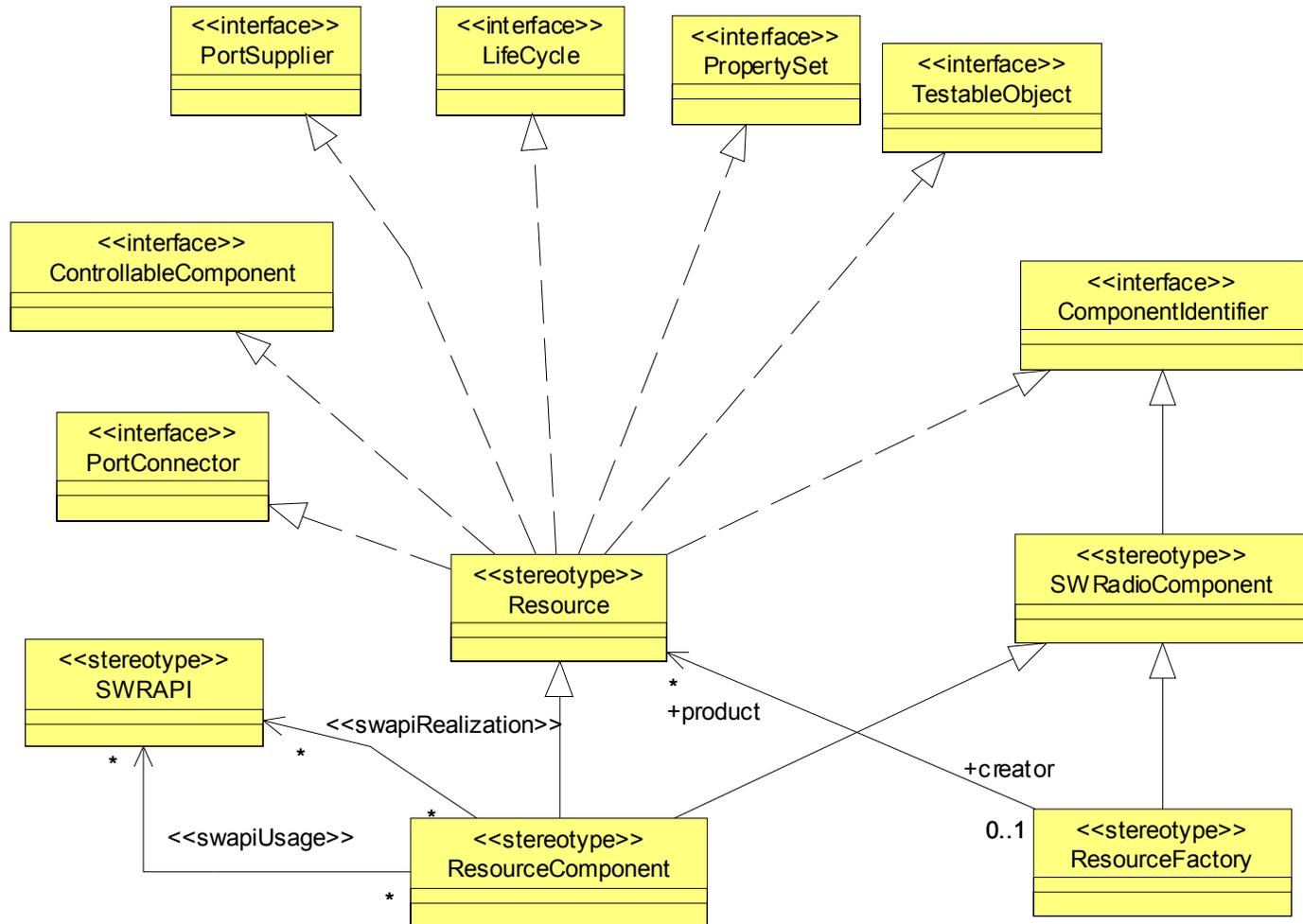
Application and Device Components

Infrastructure

- To be consistent with the three development viewpoints, the UML Profile for SWRadio is partitioned in three main packages:
  - the Applications and Devices Components,
  - the Infrastructure, and
  - the Communication Equipment package.
- Each package defines the set of concepts and UML stereotypes required to perform a specific role in the development of an SWRadio product.

8

Software-Based Communications

- Application Components
  - Contains the component stereotypes for application developers
  - Application, ApplicationResourceComponent, LayerResource (Data Link, MAC, Physical)
- Base Types
  - Contains the common types for defining SWRadio components.
- Interface & Port Types
  - Contains the port and interface stereotypes for SWRadio interfaces and components
- Device Components
  - Contains the component stereotypes for device developers
  - Logical Device, Loadable and Executable
- Properties
  - Contains property stereotypes for SWRadio components
  - Configure, Query, Characteristic, Capacity
- Resource Components
  - Contains the interface and component stereotypes for waveform and device developers
  - ControllableComponent, LifeCycle, PropertySet, ResourceComponent, etc.

9

Software-Based Communications

MODEL DRIVEN ARCHITECTURE

<<interface>>
PortSupplier

<<interface>>
LifeCycle

<<interface>>
PropertySet

<<interface>>
TestableObject

<<interface>>
ControllableComponent

<<interface>>
ComponentIdentifier

<<interface>>
PortConnector

<<stereotype>>
Resource

<<stereotype>>
SWRadioComponent

<<stereotype>>
SWRAPI

<<swapiRealization>>

*
+product

<<swapiUsage>>

+creator

<<stereotype>>
ResourceComponent

0..1

<<stereotype>>
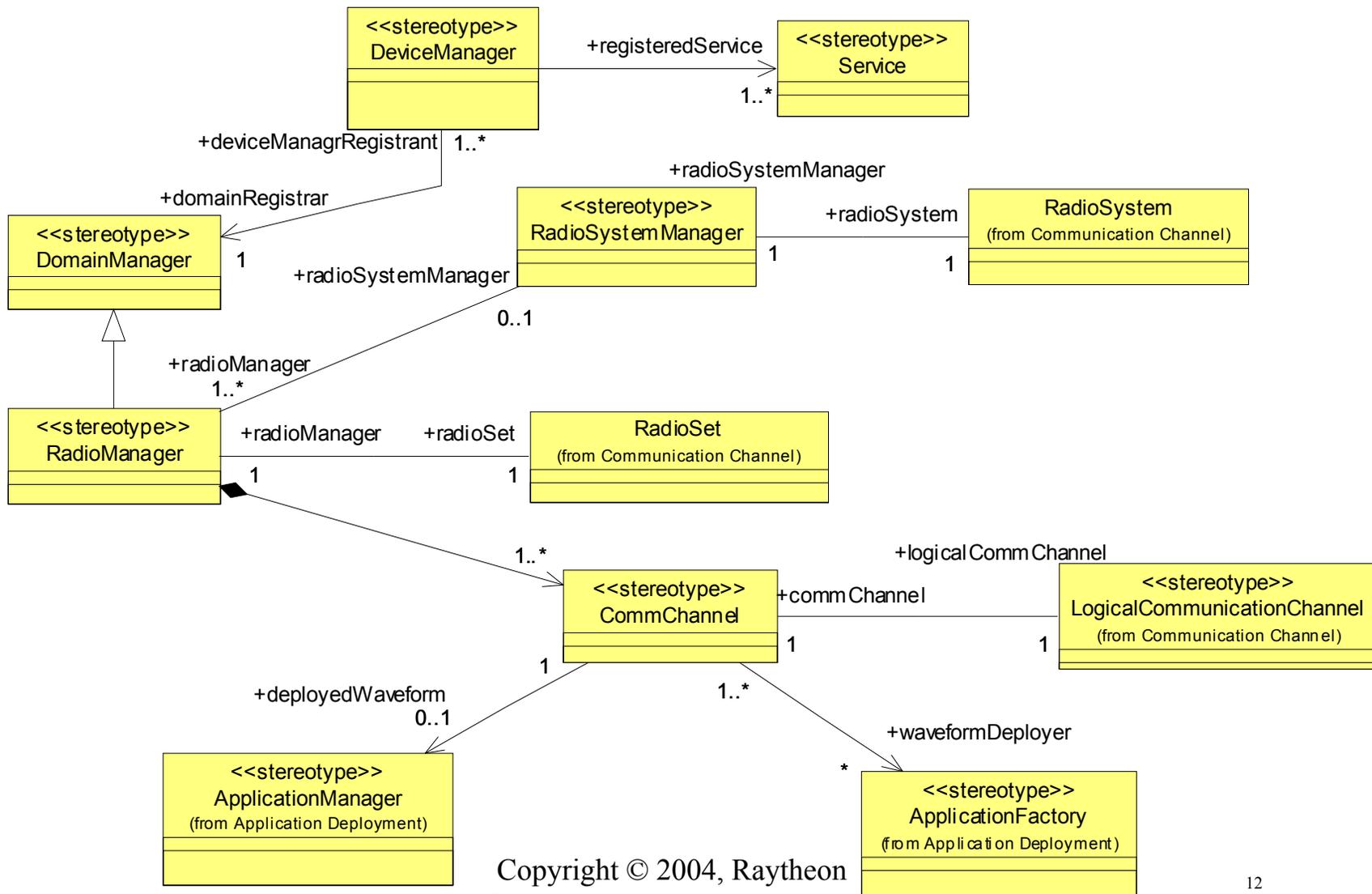ResourceFactory

*

*

*

*

10

UML Profile for SWRadio Infrastructure

- **Radio Services**
  - Common services within the radio platform that are utilized by applications
  - Managed component service
- **Radio Management**
  - RadioSet, RadioSystem, and Device Management
- **Communication Channel**
  - Physical, IO, Security, and Processing Channel
  - Captures the relationships between channels and SWRadio devices
- **Application Deployment**
  - Components and Artifacts stereotypes for the deployment of:
    - Waveforms on communication channel's distributed devices
    - Radio Services within the Radio Set

12

UML Profile for SWRadio Communication Equipment

- Stereotypes for SWRadio devices
- Communication Equipment describes the relationships and attributes that are appropriate for radio devices.
  - Crypto Device - performs encryption and decryption on asset of data.
  - I/O Device - describes the relationships and attributes that are appropriate for I/O devices
    - Antenna, Amplifier, Filter, Frequency Converter, audio, serial, etc.
  - Power Supply - provides electrical power to other devices.
  - Processor Device - processes digital or analog data.
- Port Types
  - Analog & Digital
- Property Types
  - Characteristic & Configure

UML Profile for SWRadio Communication Equipment, cont'd

Copyright © 2004, Raytheon Company. All Rights Reserved
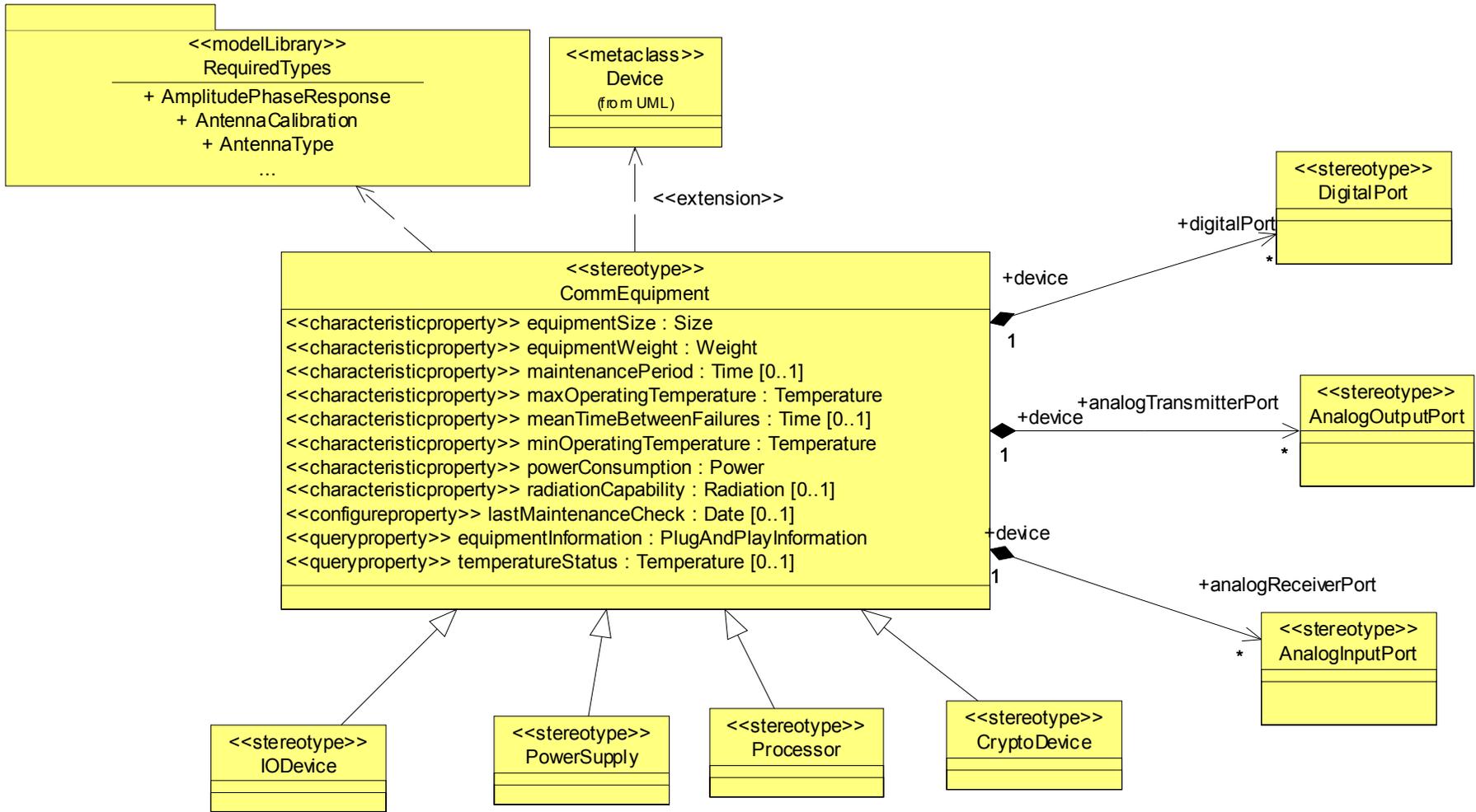
SWRadio PIM Facilities

- ## Common Radio Facilities
  - Provides common service definitions that are applicable for all applications (waveforms or radio control)
  - File Services, OMG Lightweight Services (log, event, naming, etc.)

- ## Common Layer Facilities
  - Provides interfaces that cross cut through facilities that correlate to layers. These interfaces can be viewed as building blocks for SWRadio components that realize multiple interfaces.
  - Protocol Data Unit, Error Control, Flow Control, Measurement, Quality of Service, and Stream Facilities

SWRadio PIM Facilities, cont'd

- Data Link Facilities
  - Link Layer Control (LLC) facilities. LLC layer provides facilities to upper layers, for management of communication links between two or more radio sets.
  - Data Link Layer (Connectionless, ConnectionLess Ack, Connection), and Medium Access Control Facilities
- I/O Facilities
  - Defines the configuration properties for Audio and Serial Facilities

- **Physical Layer Facilities**
  - Modem Facilities
    - The modem facilities include all digital signal processing elements required to convert bits into symbols and vice versa.
  - RF/IF Facilities
    - The RF/IF Facilities is used to configure and control the basic devices of the physical channel. The granularity at which these interfaces are implemented is not specified.

- **Radio Control Facilities**
  - Provides for interfaces for radio and channel management.

SWRadio PSM

- Automatic PSM generation from PIM and profile definitions
  - Transformation rule set specified in the specification
- Platform Specific Model
  - CORBA Modules
    - CF
      - StandardEvent, PortTypes
    - DfSWRadio
      - CommonLayer, DataLinkLayer, CommonRadio, PhysicalLayer, RadioControl
    - DSFileServices
  - XML Schema
    - Properties
    - Communication Channel
    - Physical Layer Properties
  - POSIX
- Other PSMs could be defined

SCA 2.2.1 Comparison

- Core Framework IDL Differences

- Properties

- SCA API Supplement vs SWRadio Facilities

- Requirements

SCA 2.2.1 Comparison – Core Frame IDL Differences

- Resource Interface Changes

- Device Interface Changes

- DeviceManager Interface Changes

- DomainManager Interface Changes

Resource Interface Changes – SCA Similarities

- New Interfaces Derived from Existing Resource Interface
  - ComponentIdentifier Interface
    - Readonly Identifier Attribute
  - ControllableComponent
    - Start and Stop from Resource interface

- Existing Interface
  - LifeCycle

Resource Interface Changes - SCA Differences

- New Interface
  - ControllableComponent
    - <u>Added</u> readonly <u>started</u> attribute

- PropertySet
  - Modified PartialConfiguration exception to return <u>reasons</u> for the configure exception.

- TestableObject
  - TestId parameter changed to a string

Software-Based Communications

- Port Behavior
  - Change name of Port interface to PortConnector interface
    - Connections managed at the Resource level instead of at the required/uses port level but an implementation can still manage connections at the required port level.
    - ConnectPort Operation
      - added <u>requiredPortName</u> input parameter.
      - Added another error code for InvalidPort exception to indicate the Required Port name does not exist for this component.
  - PortSupplier
    - getPort operation behavior change to only return provided port object references

23

Resource Interface Changes
- Illustration

<<interface>>
LifeCycle

initialize()
releaseObject()

<<interface>>
PropertySet

configure()
query()

<<interface>>
TestableObject

runTest(**testid : in string**, testValues : inout Properties) : void

<<interface>>
PortSupplier

getPort()

<<interface>>
ControllableComponent

<<readonly>> started : Boolean

start()
stop()

<<stereotype>>
Resource

<<interface>>
ComponentIdentifier

<<readonly>> identifier : String

<<interface>>
PortConnector

connectPort(requiredPortName : in String, connection : in Object, connectionId : in String) : void
disconnectPort(connectionId : in String) : void

New interface but no impact to existing implementations

Impact to existing implementations

Device Interface Changes – SCA Similarities

- Readonly Attributes
  - Identifier, CompositeDevice, Label, SoftwareProfile, Admin, Operational, Usage

- Capacity Operations
  - allocateCapacity and deallocateCapacity

Device Interface Changes
- SCA Differences

- New Interface
  - StateManagement
    - The setAdminState operation changed to use a new AdminRequestType and UnsupportedRequest exception.
    - <u>Added</u> readonly states attribute to return all states at once
    - <u>Added</u> readonly admin characteristic supported attribute
    - Optional admin behavior at the profile level
    - Moved the state attributes and types from Device
      - Readonly Attributes: Admin, Operational, Usage
- Device inherits StateMangement interface
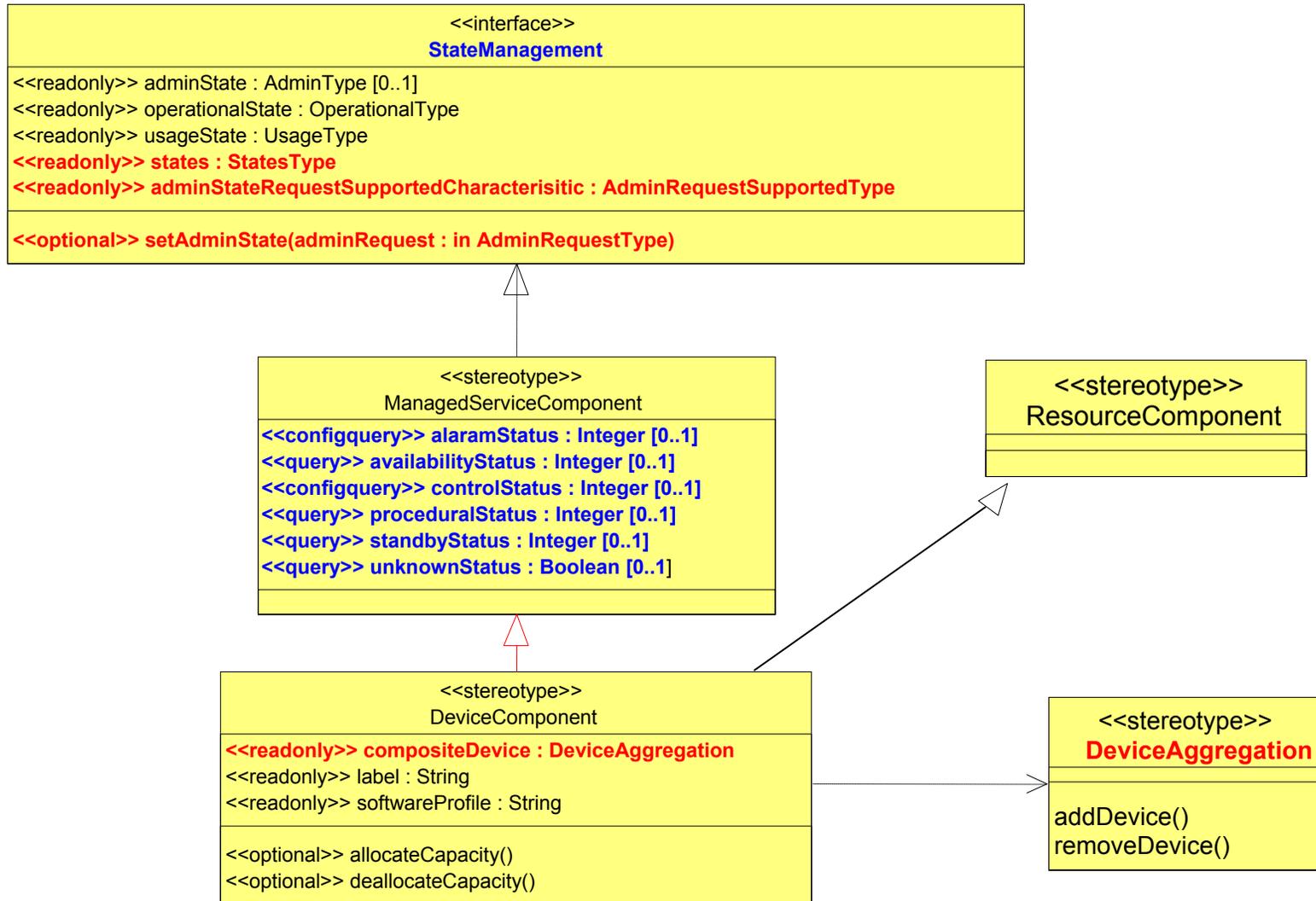- DeviceAggregation
  - Rename AggregateDevice interface

Device Interface Changes
- SCA Differences, cont'd

- New <u>Optional</u> Configure and Query Properties taken from ISO/IEC International Standard 10164-2.

  - Alarm Status

  - Availability Status

  - Control Status

  - Procedural Status

  - Standby Status

  - Unknown Status

Device Interface Changes - Illustration

```
+-----------------------------------------------------------------------+
|                        <<interface>>                                  |
|                      StateManagement                                  |
+-----------------------------------------------------------------------+
| <<readonly>> adminState : AdminType [0..1]                            |
| <<readonly>> operationalState : OperationalType                       |
| <<readonly>> usageState : UsageType                                   |
| <<readonly>> states : StatesType                                      |
| <<readonly>> adminStateRequestSupportedCharacterisitic :             |
|              AdminRequestSupportedType                                |
|                                                                       |
| <<optional>> setAdminState(adminRequest : in AdminRequestType)        |
+-----------------------------------------------------------------------+
```

```
+--------------------------------------------+      +---------------------------+
|             <<stereotype>>                 |      |      <<stereotype>>       |
|         ManagedServiceComponent            |      |    ResourceComponent      |
+--------------------------------------------+      +---------------------------+
| <<configquery>> alaramStatus : Integer [0..1]|    |                           |
| <<query>> availabilityStatus : Integer [0..1]|    +---------------------------+
| <<configquery>> controlStatus : Integer [0..1]|
| <<query>> proceduralStatus : Integer [0..1]|
| <<query>> standbyStatus : Integer [0..1]   |
| <<query>> unknownStatus : Boolean [0..1]   |
+--------------------------------------------+
|                                            |
+--------------------------------------------+
```

```
+--------------------------------------------+      +---------------------------+
|             <<stereotype>>                 |      |      <<stereotype>>       |
|            DeviceComponent                 |      |     DeviceAggregation     |
+--------------------------------------------+      +---------------------------+
| <<readonly>> compositeDevice : DeviceAggregation| |                           |
| <<readonly>> label : String                |      +---------------------------+
| <<readonly>> softwareProfile : String      |      | addDevice()               |
+--------------------------------------------+      | removeDevice()            |
| <<optional>> allocateCapacity()            |      +---------------------------+
| <<optional>> deallocateCapacity()          |
+--------------------------------------------+
```

LoadableDevice Interface Changes

- Impacted by StateManagement Interface & DeviceAggregation Name Change

- Added Characteristic Properties

  - Loadkind

  - Name and Version Types
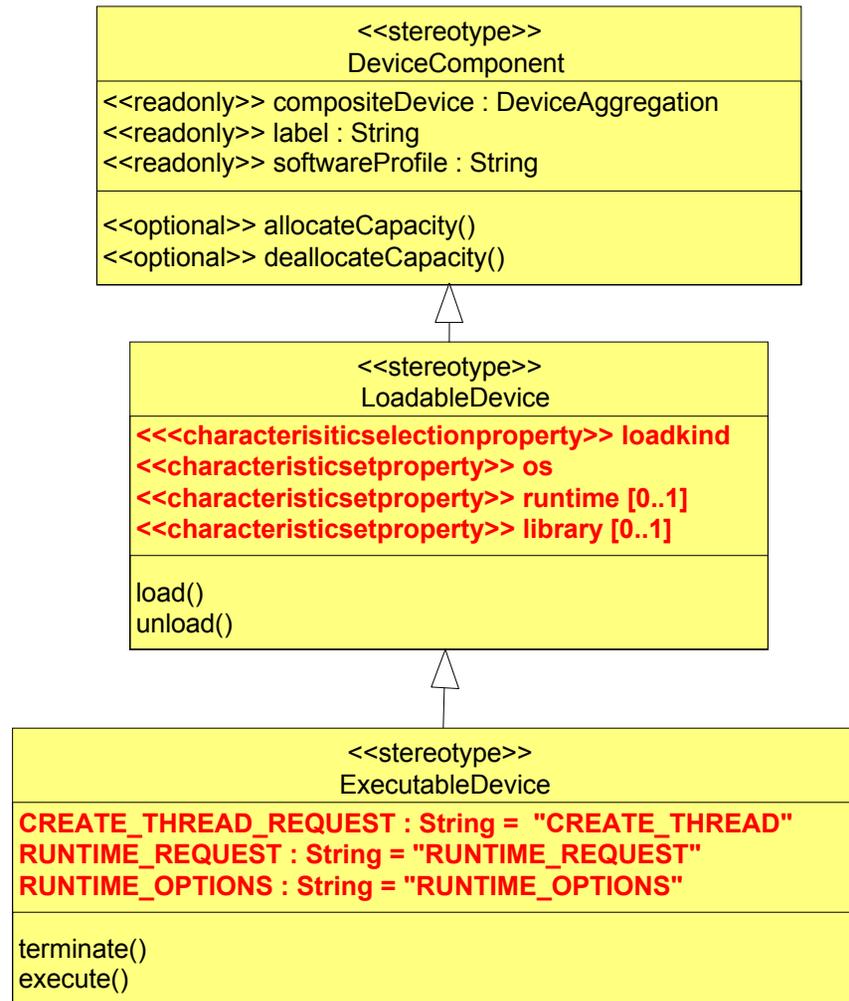
    - OS

    - Runtime

    - Library

ExecutableDevice Interface Changes

- Impacted by StateManagement Interface & DeviceAggregation Name Change

- Additional Execute behavior for Options parameter

  – Added new Options:

    - Runtime Request & Options

      – Creates a runtime environment

    - Thread Create Request

      – Creates a thread in a process

<<stereotype>>
DeviceComponent

<<readonly>> compositeDevice : DeviceAggregation
<<readonly>> label : String
<<readonly>> softwareProfile : String

<<optional>> allocateCapacity()
<<optional>> deallocateCapacity()

<<stereotype>>
LoadableDevice

<<<characterisiticselectionproperty>> loadkind
<<characteristicsetproperty>> os
<<characteristicsetproperty>> runtime [0..1]
<<characteristicsetproperty>> library [0..1]

load()
unload()

<<stereotype>>
ExecutableDevice

CREATE_THREAD_REQUEST : String = "CREATE_THREAD"
RUNTIME_REQUEST : String = "RUNTIME_REQUEST"
RUNTIME_OPTIONS : String = "RUNTIME_OPTIONS"

terminate()
execute()

DeviceManager Interface Changes - SCA Similarities

- Still supports
  - Attributes
    - Identifier
    - Label
    - Device Configuration Profile
    - File System
    - registeredServices
  - Operations
    - Shutdown
    - getComponentImplementation

DeviceManager Interface Changes - SCA Differences

- New Interface broken out from DeviceManager
  - RegisteredService interface
    - Un/register service operations removed from DeviceManager interface
    - Given out by the DeviceManager to services that are started by the DeviceManager through its configuration profile.
- Added new Operation
  - portExists
    - Use to determine functionality that is supported
- Removed Device un/register operations and registeredDevices attribute
  - Collapsed with service operations behavior
- Port Interface changes as described previously for Resource Interface Changes
  - PortConnector – added at DeviceManager level
  - PortSupplier - Use to to give out functionality that is supported
  - PropertySet
- FileSystem attribute – FileSystem interface different CORBA Module

DeviceManager Interface Changes - Illustration

Software-Based Communications

<<interface>>
PropertySet

<<stereotype>>
SWRadioComponent

<<interface>>
PortConnector

<<interface>>
PortSupplier

<<stereotype>>
DeviceManager

<<readonly>> registeredServices : ServiceType [*]
<<readonly>> deviceConfigurationProfile : String
<<readonly>> fileSys : **FileSystem**
<<readonly>> label : String

shutdown()
getComponentImplementationId()
portExists()

<<stereotype>>
ServiceRegistration

registerService()
unregisterService()
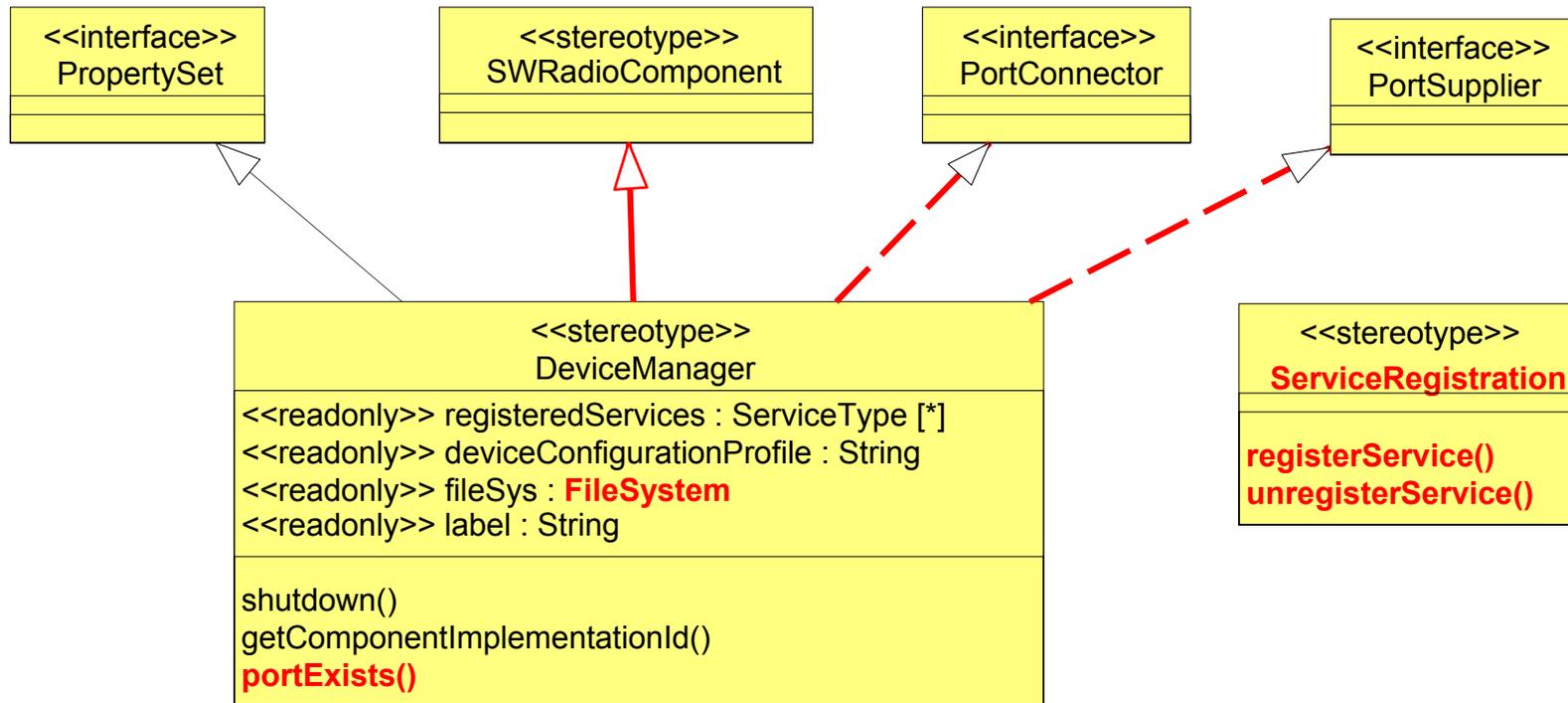
DomainManager Interface Changes - SCA Similarities

- Attributes
  - Applications
  - Application Factories
  - Device Managers
  - Domain Manager Profile
  - File Manager

Copyright © 2004, Raytheon
Company. All Rights Reserved

DomainManager Interface Changes - SCA Differences

- New Interfaces broken out from DomainManager
  - Domain Event Channels
    - Domain Manager's event channel operations
  - Device Manager Registration
    - Domain Manager's device manager and service registration operations
  - Domain Installation
    - Domain Manager's application installation operations
- Removal of Device registration operations
  - Behavior now part of un/registerService operations
- Added Capability
  - Port Supplier
    - Use to to give out functionality that is supported (e.g., optional interfaces that have been broken out from the DomainManager)
  - PortExists
    - Use to determine functionality that is supported. Well-Defined port names for functionality broken out from DomainManager interface
- FileMgr attribute – FileManager interface different CORBA Module

<<interface>>
PropertySet

<<stereotype>>
SWRadioComponent

<<interface>>
PortSupplier

<<stereotype>>
DomainManager

<<readonly>> applications : ApplicationManager[*]
<<readonly>> ApplicationFactories : ApplicationFactory [*]
<<readonly>> deviceManagers : DeviceManager [*]
<<readonly>> domainManagerProfile : String
<<readonly>> fileMgr : **FileManager**

portExists()

<<stereotype>>
DomainInstallation

installApplication()
uninstallApplication()

<<stereotype>>
DomainEventChannels

registerWithEventChannel()
unregisterFromEventChannel()

<<stereotype>>
DeviceManagerRegistration

registerDeviceManager()
unregisterDeviceManager()
registerService()
unregisterService()

SCA Service Interfaces

- File Services
  - Own separate module

- PortTypes
  - Module within CF

- StandardEvent
  - Module within CF

Properties

- SCA Property Similarities
  - Executable
  - Simple
  - Structure
  - Simple Sequence
  - Structure Sequence
  - Test

- SCA Property Differences
  - Service Property
    - Capacity and Characteristic Property Types
  - Properties PSM is XML Schema

Properties – Service Properties
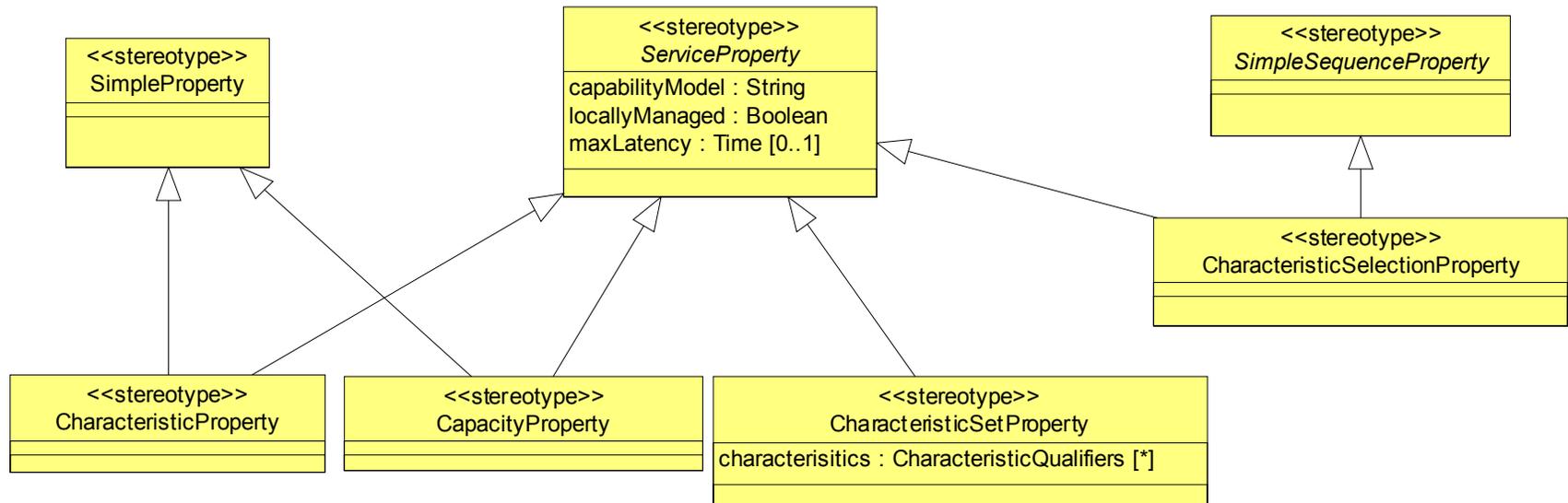
- CharacteristicProperty – static characteristic with one value
- CapacityProperty – dynamic capacity
- CharacteristicSelectionProperty – A list of single values supported by a characteristic (e.g., load kind)
- CharacteristicSetProperty – A list of values of the same characteristic (e.g., library, os, runtime), where each value is a list of one or more name/value pairs (e.g., name, version).

40

- Common Layer Facilities maps to SCA API Supplement Packet Building Block Appendix C maps to

  - Equivalent in functionality but interface name, operation name and parameter type changes along with additions

  - PDU maps to Packet interfaces

    - ISimplePDU maps to SimplePacket

    - IPriorityPDU maps to Packet

  - PacketSignals maps to IFlowControlSignaling

  - StatusSignal maps to SignalError

  - Additional PDU Template and Concrete Interfaces in SWRadio

SWRadio Data Link Layer Facilities vs SCA API Supplement

- Data Link Layer Facilities maps to MAC and Link Layer Control Building Blocks (SCA API Supplement Appendix F and G)
  - Expanded to support other specifications: DLPI specification, OSI reference model, X.200e, IEEE 802 series, 3GPP UMTS and GSM specifications
  - Equivalent in functionality but interface name, operation name and parameter type changes.
    - Local Link Management
    - Connectionless Link
    - Acknowledged Connectionless
      - No user-side specific operations
    - MAC
      - No Transec, MACAddressing, RxTerm interfaces
  - Connection Link – not in the SCA

42

SWRadio Physical Layer Facilities vs SCA API Supplement

- Physical Layer Facilities maps to Physical Layer Building Blocks (SCA API Supplement Appendix D and E)
    - Equivalent and expanded functionality with well defined properties and more granularity of interfaces
        - Pseudo Noise (PN) Sequence Generator, Mapper, Interleaver types, transform
        - Average Power, FrequencyConverter, Polarization, FrequencyResponse, RadiationPattern, SampleRate, Switch
    - Based upon PropertySet interface
        - Provides for flexibility that can easily provide for reprogrammable waveform needs since filters, source coding, channel coding, etc. are specific to waveforms.
    - No real-time headers for packets are defined or template interface with generic get and set operations with template type.

SWRadio I/O Facilities vs SCA API Supplement I/O

- Physical Layer Facilities maps to SCA API Supplement I/O Building Blocks Appendix H

  – Well defined properties for serial and audio

    • No template I/O Configuration interface with generic get and set operations with template configuration type.

  – Relies on Common Layer Facilities (PDU, Flow Control, etc.)

  - Equivalent functionality for I/O control (RTS and PTT) interfaces

  - No Audio Audible Alarms and Alerts Service in SWRadio

44

Resource Interface – SWRadio Requirements Modifications

- The Resource Component has the responsibility to support its port operations

- Component identifier is no longer required to be unique (to be modified)

- Valid properties for the configure operation are no longer specified (to be modified)

- The stop operation no longer has the responsibility of disabling all current operations (pending issue)

- Explicit references for create resource to return CORBA references have been removed (due to platform independent nature of the spec)

Device Interface – SWRadio Requirements Modifications

- New Managed Service Component aspect of the Device introduces specific behaviors for the Device (state model actions revised)

- New required behaviors for runtime creation and explicit thread creation (introduction of runtime and thread options)

- Relationship between specification and specific event channels has been severed (publishing an event to an IDM/ODM to be modified)

- The ability to have either a DTD file references or an inline descriptor reference has been removed (in favor of only file references)

- SetAdminState disabled when release object is invoked

- DeallocateCapacity no longer has to set the adminState attribute to locked

DeviceManager Interface – SWRadio Requirements Modifications

- Formalizes behavior of which items are terminated by shutdown and when the termination occurs

- Introduces a role-based capability and associated behavior to assist in the DeviceManager decomposition and separation of responsibilities (DeviceManager can host Ports)

- Formalizes responsibility of DeviceManager to interact with the Device capacity model (e.g. allocation)

- Devices and Services have been logically grouped with one another (all are Services)

- Identification of specific descriptor files has been removed (DCD is not explicitly mentioned)

- Relationship between the DeviceManager and specific Logs has been severed (registerDevice – service, no longer writes to DomainManager's log)

- FileSystem mount names need to be unique within the DeviceManager

DomainManager Interface – SWRadio Requirements Modifications

- Introduces a role-based capability and associated behavior to assist in the DomainManager decomposition and separation of responsibilities (DomainManager can host ports)

- Relationship between the DomainManager and specific Logs has been severed (no requirements exist to write specific Log records)

- Identification of specific descriptor files has been removed (DMD is not explicitly mentioned)

- DomainManager no longer has responsibility to create IDM and ODM event channels

- Devices and Services have been logically grouped with one another

- registerService no longer has to validate object references

Requirements - Summary

- SCA Core
  - 509 Requirements
- OMG SWRadio Specification
  - UML Profile interface decomposition has introduced "new" requirements into the specifications
    - OMG property related requirements extend and formalize expected behaviors present in the SCA Appendix D text.
  - Platform independence in the OMG SWRadio specification has eliminated the need for a number of the SCA requirements – however it is likely that if an "SCA PSM" were created from the OMG spec it would be appropriate to reintroduce a number of the requirements.

  - 449 Requirements
    - 181 new requirements
      - Many of which describe features which exceed SCA capabilities
    - 268 Identical or very similar requirements
    - 241 SCA requirements deleted
      - 73 related to event / log message publishing
      - 53 related to Logical Device / application / Security / descriptors
    - Includes facilities and AEP