EVOLUTION AND STANDARDIZATION OF THE SOFTWARE COMMUNICATIONS ARCHITECTURE

Dominick Paniscotti (VP SDR Products of PrismTech Corp., Burlington, Mass.; dominick.paniscotti@prismtech.com); Jerry Bickle (Chief Scientist SDR Products of PrismTech Corp., Burlington, Mass.; jerry.bickle@prismtech.com)

ABSTRACT

An effort to commercially standardize the Joint Tactical Radio System's (JTRS) Software Communications Architecture (SCA) specification was undertaken by the Object Management Group (OMG) several years ago. In December 2005, a new standard known as the Platform Independent Model (PIM) and Platform Specific Model (PSM) for Software Radio Components was adopted by the OMG to serve as an open, commercial standard for the development of Software Defined Radios (SDR) such as the JTRS. This article discusses how this new commercial specification relates to the existing JTRS SCA specification and details the extensions available as part of this specification (such as communication channels and the Unified Modeling Language (UML) Profile for Software Radio (SR). In addition, the article will also focus on specification conformance.

1. INTRODUCTION

The OMG PIM and PSM for Software Radio (SR) Components specification is based upon the concepts found in the SCA (such as a waveform application consisting of resource components, domain manager component, logical device and device manager components) but represent these concepts in a domain specific language known as the UML Profile for Software Radio. The SCA mandates specific technologies such as Common Object Request Broker Architecture (CORBA) and eXtensible Markup Language (XML) and further describes the use of these technologies to support the definition and development of Software-Defined Radio (SDR) software.

The UML Profile for SR, on the other hand, expresses these concepts in strictly architectural terms (not mandating any particular technologies) using only UML. In this manner, the relationships between the software elements making up the SR architecture can be strictly defined and constrained. This level of "software engineering completeness" further allows the ability to transform models built to this UML Profile directly into executable software that can be deployed on SRs.

The OMG PIM and PSM for SR components consists of a UML Profile for Software Radio, data link and physical layer facilities, and POSIX profiles, which are explained in the following sections.

2. UML PROFILE FOR SOFTWARE RADIO

The UML Profile for SR extends the elements in the UML 2.0 specification (such as artifact, component, device, interface, property) by forming new definitions pertaining to a SR domain specific language. These new definitions extend UML elements via additional attributes, constraints, and semantics. The UML Profile for SR provides the required language to support modeling waveform applications, platform components, and communication channels independent of specific technology choice and then allows one to transform these elements directly into the implementation technologies required. This allows the design of the radio software to be captured independent of technology (e.g. C, CORBA, Java, or XML) and then later mapped to technology as required to meet the developer's needs as illustrated in Figure 1.

For example, waveform application components might initially be implemented in C++ for the Linux operating system running on a Pentium general purpose processor (GPP). Later, possibly due to technology upgrades the same waveform components might be needed in VHDL to support a field programmable gate array (FPGA) or in C to support a digital signal processor (DSP). Since the specification does not force technology choice, the developer can transform the architectural concepts found in the specification to support the VHDL (FPGA) or C (DSP) implementation. Figure 2 serves to illustrate the concept. In all cases, however, the new technology dependent implementations must adhere to design set forth by the original designer of the component and is enforced by the UML profile.

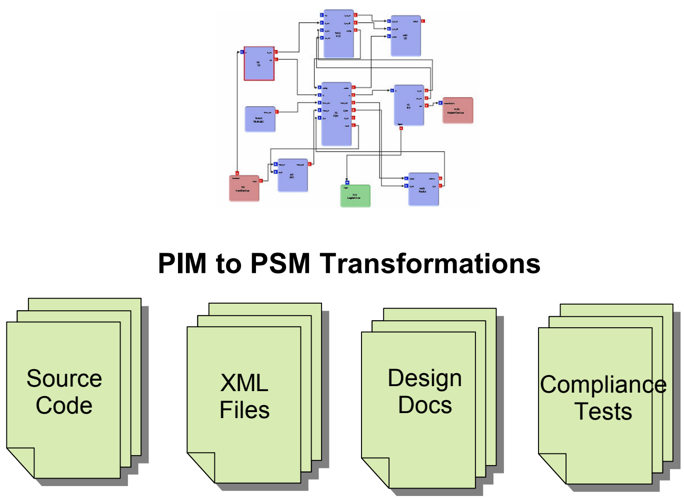

Figure 1. Model Driven Development (MDD) Technology Transformations

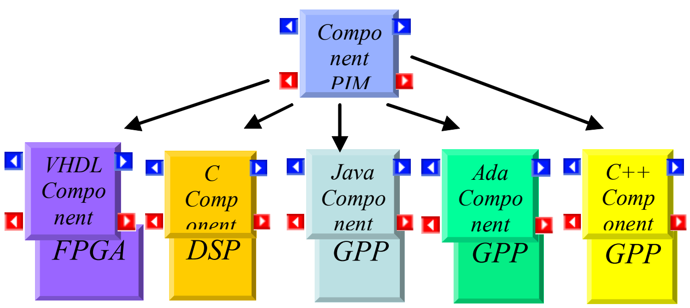

Figure 2. Component PIM to PSM Transformations

Tooling makes this task significantly easier as models of the waveform and radio platform designs can be captured in a technology independent manner then transformed using generative programming approaches into the source code language of choice.

Tools can further capture the designs in a language called XML Metadata Interchange Format (XMI). XMI, another standard managed by the OMG, allows the import and export of model information between tools. The XMI is expressed in terms of the UML Profile for SR. Using XMI allows modeling tools designed for particular aspects of radio development to exchange the information needed to support seamless integration. Tools implementing the SR profile will likely also have domain specific editors and generators as illustrated in Figure 3. These editors and generators will vary across tools, however having a single UML Profile underpinning these tools. And import and export capabilities via XMI ensure the users of these tools are not locked to any particular vendor.

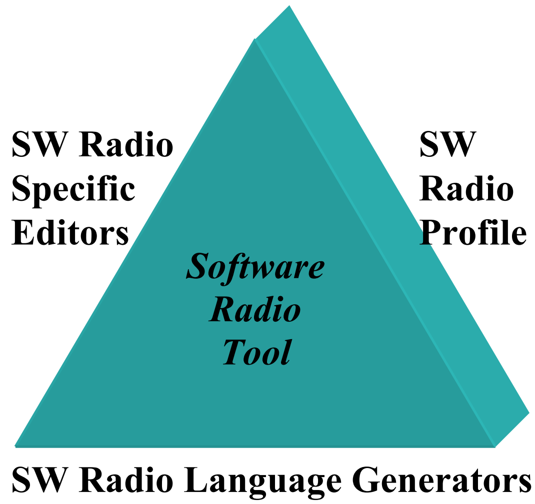

Figure 3. Software Radio Tool Features

The UML Profile for SR is designed to be extended for specific domains such as military radio, space/ground based radios, robotics, commercial handset and base station domains via extension points which allow the incorporation of additional architectural concepts and/or constraints. It comprises two UML profiles: Component Framework Profile and Communication Channel Profile.

2.1. COMPONENT FRAMEWORK PROFILE

The Component Framework Profile is a UML Profile that captures the SCA concepts for waveform components (Resource and Resource Factory) and platform components (DomainManager, DeviceManager, Device and Service). These component definitions are stereotypes that extend the UML 2.0 component and are further constrained to realize specific interfaces and property types. The profile contains additional SCA component concepts such as configure, query, capacity, characteristic, and executable properties, as well as both uses and provides ports to support component connections.

A modeling tool utilizing the Component Framework Profile can capture waveform and platform components using a PIM and then transform those models into the specific technologies required by the SDR developer. The benefits of modeling tools using such a profile are:

1. Increased quality of the end product by left shifting defect detection from integration phases to analysis and design phases.
2. Productivity increases.
3. Test automation – from the model, the generation of test code that validates conformance of the generated and developed source code. Additionally

all artifacts (models, source code, tests) are kept completely up to date and in sync with other.

The Component Framework Profile also contains model library packages that capture UML interfaces for radio component PIM definitions. These interfaces relate to the SCA CORBA interfaces (Resource, Device, DeviceManager, DomainManager, etc.) but are expressed as UML interfaces so they can be transformed into other technologies besides CORBA.

The platform specific technologies specified in the OMG SR specification at this time for the Component Framework Profile are CORBA and XML. The UML interfaces are transformed into CORBA interfaces and the component descriptors are transformed into similar SCA DTDs or in component schema descriptors as defined in the CORBA Component Model (CCM).

2.2. COMMUNICATION CHANNEL PROFILE

The Communication Channel Profile is a UML Profile that captures the concepts of a radio's end-to-end communication data path (i.e. baseband to RF). This extends the concepts currently found in the SCA. The definition of a communication channel aggregates other logical channels (ie. I/O, physical, processing, secure). The channel concept, as shown in Figure 4, is expressed in the profile as an extension of the UML class. Channel types (I/O, physical, processing, secure, etc.) are therefore specializations of a UML class. A channel is associated with one or more communication equipment types. Depending on the channel type determines the set of communication equipments that are associated with a channel.

The communication equipment classifier is an extension of the device classifier, which forms the base definition for all software-based radio hardware abstractions. The specializations (antenna, amplifier, audio, converter, filter, processor, serial device, etc.) of the communication equipment add additional attributes and further constrain the type of ports associated with a device.

In addition to the profile, there is radio control and physical layer facilities defined for managing a radio's communication channel. The radio manager concept is an extension of the SCA domain manager with additional facilities for managing communication channels and state management.

3. SR PIM FACILITIES

The SR PIM facilities define a set of interfaces that can be used to express component definitions. The facilities are expressed as UML interfaces and are broken into following categories:

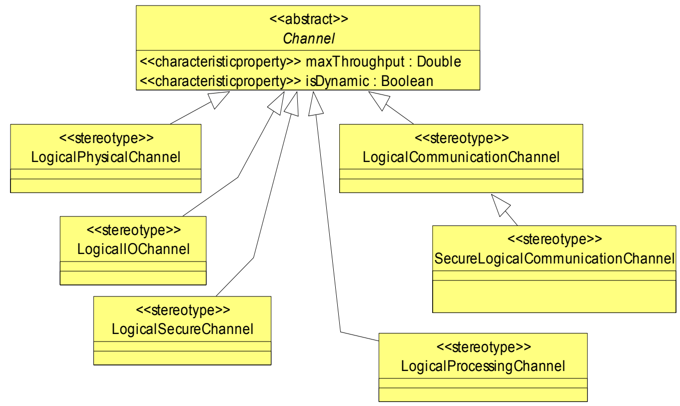

Figure 4. Channel types are specializations of a UML class

- Common layer facilities - This facility defines the set of interfaces that all components (regardless of any layering) within a radio platform can realize. Examples of these types of interfaces are flow control, protocol data unit (PDU), and stream interfaces.
- Data link layer - These facilities define link layer control (LLC) and media access control (MAC) layer functionality for communication needs.
- IO - These facilities define serial and audio component definitions for communication needs.
- Physical layer - These facilities 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. The facilities also include frequency tuning, filters, interference cancellation, analog/digital conversion, up/down conversion, gain control, synthesizer etc., and functionality.

The interfaces in the physical layer facilities at this time continue to be a work in progress. Current activities at the OMG and SDR Forum (SDRF) are focused on the definition of smart antenna and digital IF interfaces.

4. POSIX PROFILES

The OMG specification, much like the SCA, defines an Application Environment Profile for embedded, resource-constrained systems, based on a standardized POSIX based Application Environment Profile (AEP). In addition to this, a lightweight Application Environment Profile (LwAEP) is defined. The LwAEP is constrained version of the AEP and is targeted at environments with limited computing resources. Examples of constrained embedded processors include DSPs, processor cores within FPGAs and microcontrollers.

5. SCA RELATIONSHIPS

The SCA is defined at PSM level using CORBA interfaces and XML DTDs. In contrast, the OMG PIM and PSM for SR components defines UML Profiles for modeling SR concepts and PIM facilities that can be transformed into any technology.

The DTDs defined in the OMG specification are backwards compatible with existing SCA DTDs but have been extended. These extensions allow component implementations to be decomposed into component assemblies and to express collocation semantics for components in an assembly. Also some enumeration types were made string types for greater flexibility for industry usage and implementations.

The Core Framework (CF) and PortTypes CORBA module interface definition language (IDL) in the OMG specification are also broken into multiple files instead of condensed into one monolithic file (as in the SCA). This allows implementations of these interfaces to remain smaller in memory size as only the required interfaces are found in the executable software images.

The OMG specification, based upon lessons learned from several SCA Core Framework implementations, has also introduced changes to further optimize the following areas:

- Deployment optimizations- The meaning of property definitions and values for component definition, implementation and instantiation was simplified thus making installation and deployment much simpler and more understandable.
- Component connections were also simplified.
- Teardown optimization- The process of disconnecting component connections during an application release was simplified. In the SCA, disconnect call is on all application's deployed component and radio services. In the OMG, disconnections are only necessary for radio services not for deployed components since this is now handled by the component release behavior.

To address industry concern regarding the processing and memory overhead introduced by the SCA architecture, several interfaces and components were modified. These modifications have minimal impact on compatibility with the SCA as they continue to support existing SCA implementations seamlessly but allow for lighter–weight component definitions for other SR domains.

A LwAEP was added to the OMG Software Radio specification. This definition is lighter weight than one defined in the SCA 3.0 specification and corresponds to the recommended change proposal against SCA 3.0 that was supposed to make it lighter weight. SCA 2.2.2 did not include the LwAEP in its revisions but did reference a later POSIX specification (2003) for its profile definition.

6. CONFORMANCE

Simply put, conformance is defined at level of component and interface usage. No requirement on what components are required for a radio system. One needs to determine what radio components along with their interfaces are required for a radio being built based upon the radio requirements and level of portability one is striving for.

The OMG Software Radio specification defines three levels of portability like the SCA, which are at the radio domain level, radio's node level and application level.

For the radio domain level this involves the Domain Manager/Radio Manager component and domain management interfaces. This capability provides a benefit of managing a family of radios the same way within a company's radio product lines or across many different vendors' radios. This does not mean the Human Machine Interface/Human Computer Interface looks the same but the underlying implementations are using the domain management interfaces.

For the radio node level this involves the Device Manager, Logical Device, and Service Components along with these interfaces and descriptors. This level of portability is important when third party vendors are supplying hardware and the software to interface to their hardware to radio manufacturers or radio system integrators.

For an application this involves the Resource Component, Resource interfaces, properties, and descriptors. This level of portability is important when third party vendors are supplying waveform applications to radio manufacturers or radio system integrators.

One needs to determine the level of portability they are striving for their radio system in order to determine the components along with their interfaces needed by a radio set or system. This will determine the elements used in the Software Radio Profile when building components for a radio and for a waveform application.

7. SUMMARY

In summary, the OMG PIM and PMS for SR Components allows SDR developers to capture waveform and platform designs of SCA conformant systems in a manner that is independent of implementation technology. This allows SDR manufacturers to leverage existing SCA investments, provides a cost effective mechanism for SCA evolution and enables SR technology insertion. At the same time, the OMG PIM and PSM for SR Components is flexible enough to accommodate commercial SDR domains since conformance is defined at the level of component and interface usage.

