

THE TRANSCEIVER FACILITY PLATFORM INDEPENDENT MODEL (PIM): DEFINITION, CONTENT AND USAGE

Eric Nicollet (Thales Communications, Paris, France; eric.nicollet@fr.thalesgroup.com)

ABSTRACT

Efficient reconfigurable radios require powerful abstraction of the Transceiver sub-systems (up/down conversion between base-band and low power RF signal), with improved portability of the embedded software implementing the radio related processing (MAC, PHY layers).

A Transceiver Facility is introduced: a modeling toolbox for Transceiver abstractions, to capture waveform-specific requirements towards Transceiver, or to specify the flexible Transceiver part of a reconfigurable radio equipment.

The way to use Transceiver Facility is then presented across the preliminary steps of a reconfigurable product development, with commented illustration in the case of a flexible Base Station.

Content of the Facility is finally described. Besides useful data transmission and reception, control capabilities are offered, such as parameters initial settings, frequency hopping and duplex state control, or adaptive level controls. How the Transceiver Facility PIM is paving the way to advances in standardization, portable waveform and open platforms, certification means and development tools is finally exposed.

1. INTRODUCTION

Efficient Reconfigurable Radio Equipments with portable waveform software applications are still a challenge to be met. This is even more true for the most embedded parts of waveform implementations, such as MAC and PHY layers. The Transceiver sub-system is playing a huge role in this area, since... any Radio equipment comprises by definition a Transceiver sub-system.

Transceiver is defined as the complete set of treatments bringing, on the transmit data path, the complex base-band samples up to the low power RF analogue signal, and reciprocally on the receive data path. Any management and control features necessary for Transceiver usage are part of the Transceiver sub-system. "Sub-system" terminology is added to insist on the fact Transceiver is one of the prime sub-systems intrinsically constitutive of any radio

equipment (such as Power Amplifier, Cosite Mitigation, Antenna, ...)

Figure 1 illustrates this matter of fact:

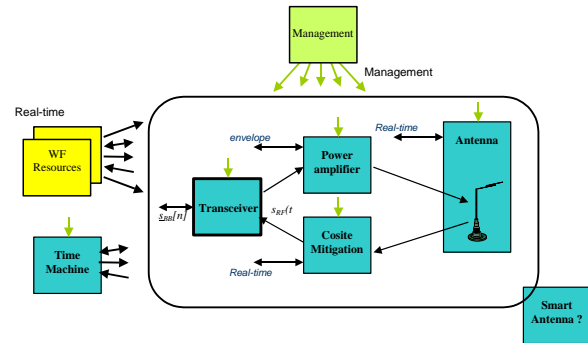


Figure 1 – How the RF part of a radio equipment is made of Radio sub-systems

To satisfy the portability needs of the waveform software applications using them, to enable the development of Reconfigurable Radio Equipments, an abstraction of the Transceiver sub-system is known to be necessary.

Since a Transceiver implementation always comprise an analogue part and a mixed signal conversion stage, coupled to a more or less important digital part. Highly varying implementation choices can be made depending on the target product. It is therefore necessary to maintain the granularity of the implementation at the complete Transceiver level, otherwise implementation assumptions would be taken.

The following figure is depicting how a Transceiver sub-system is used by waveform application resources:

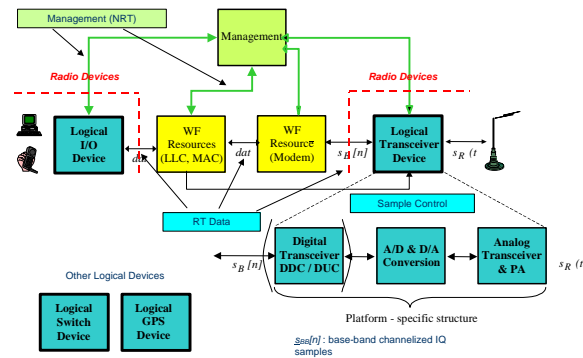


Figure 2 – Transceiver internal implementation and external interfaces

The presented results come from the standardization effort carried out to submit the Revised Submission to the OMG Digital IF request for proposal [1]. It is realized in the scope of the E²R II project [2], as a part of E²R WP4 activities [3]. It is an evolution carried out from SCA change proposal 277 [4] and OMG Digital IF Initial Submission [5]. The content of the two previous references was successfully implemented in the scope of PEA AL program.

As for the initial achievements, a total compliance exists with the OMG Software Radio Specification [6] and the JTRS Software Communications Architecture [7].

Section 2 is presenting Transceiver Facility PIM, section 3 is detailing an effective way to use it, while section 4 and 5 are respectively presenting the Transceiver Facility PIM Features and Common concepts. Conclusion is listing fields of improvements as well as the areas possibly benefiting of the work presented.

2. THE TRANSCEIVER FACILITY PIM

2.2. The Transceiver Facility

The Transceiver Facility is a set of concepts that support requirement specifications towards Transceiver sub-systems. It is intended to become a normative part of the Digital IF Revised Submission. The Facility is organized as a set of Features, supported by a set of Common concepts.

Features are the normative compliance points to the Transceiver Facility. This means any compliant Transceiver sub-system shall implement a certain list of Features, selected as a sub-set of the complete Features lists of the Facility.

This fundamental binding mechanism enables to define on the one hand a consistent and standardized set of Features, in the Transceiver Facility, while enabling on the other hand an easy and flexible exploitation inside specification of particular Transceiver sub-systems.

Each PIM Feature can be composed of any combination of:

- Interfaces, operations and arguments,
- Attributes, Types,
- State machines,
- Non functional requirements (esp. real-time),
- Applicable sequence digrams.

The Features are making a complete and consistent specification baseline, with total abstraction of the Transceiver implementation.

The attributes of the Facility are corresponding to dimensioning parameters of a Transceiver, and are not

assigned numeric values. Defining those values is part of the binding process towards the concrete specifications based on the Facility.

The following figure illustrates the content of a Feature:

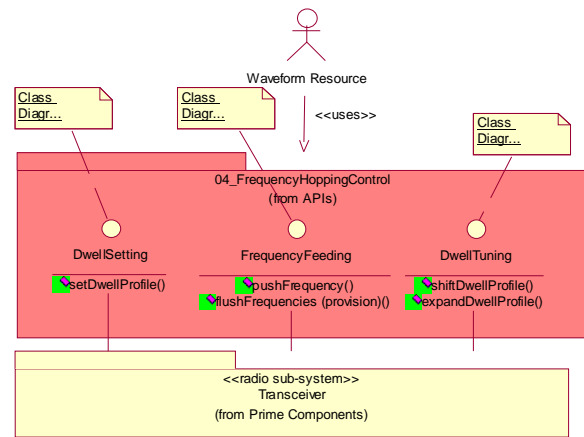


Figure 3 – Illustration of Feature content: FrequencyHoppingControl

This example shows that the feature is made of three interfaces: DwellTuning, FrequencyFeeding and DwellSetting, each being detailed through a specific class diagram.

Section 4 presents the Features of the Transceiver Facility.

Common concepts correspond to concepts shared across different features, and are captured separately from the features themselves. They are the fundamental concepts of the Transceiver sub-system abstraction.

Section 5 presents the Common concepts of the Transceiver Facility.

2.3. The Transceiver Facility PIM

A Model Driven Architecture approach is followed, thus yielding to define the Transceiver facility at PIM (Platform Independent Model) level. This PIM is realized as a formal UML 1.x model, coupled to explanatory figures and technical notes when necessary.

The root logical package is defined as “Transceiver Facility”, with one sub-package “Features” and one sub-package “Common concepts”. Inside sub-package “Features” are as many sub-packages as there are Features, each of them named from the Feature name and stereotyped <<feature>>. Inside sub-package “Common concepts” are as many sub-packages as there are Common concepts, each of them named from the common concepts name and stereotyped <<concept>>.

Each Feature or Common concept package is possibly containing interfaces and classes, state machines or

sequence diagrams. All the interfaces of a Feature are grouped in a sub-package called from the feature name and stereotyped <<API>>.

At previous level of progress [4], [5], the functional perimeter of the Transceiver PIM was the one of the demonstrations realized in PEA AL: two waveforms supported (FM3TR, 4285), running on two platforms (THALES and Raytheon), with few provisions taken to support other waveforms or platforms.

As of today, the Transceiver Facility is not attached any more to a pair of waveforms and a pair of platforms, but has become a generic modeling toolbox..

3. USAGE OF TRANSCEIVER FACILITY PIM

The purpose of this section is to illustrate how the Transceiver Facility is concretely used.

3.1. Waveform functional design

The waveform functional design defines the waveform decomposition in functional resources interacting with functional devices. The Transceiver Sub-system is part of this model, as one of the functional sub-systems participating to waveform implementation.

The Transceiver Facility is an input to this task, that is used to characterize the expectations towards Transceiver.

A binding is realized from the Flexible Transceiver Facility, with namely:

- Selection of the expected features sub-set,
- Values of dimensioning parameters (Feature's classes attributes),
- Which resources are connected to which logical port of the Transceiver,
- Connections types and real-time constraints.

The waveform functional design can be realized as a formal UML PIM, capturing the functional resources and radio-sub systems that make up the considered waveform.

Illustration

The following figure illustrates a waveform functional design in the context of a flexible UMTS-capable base-station:

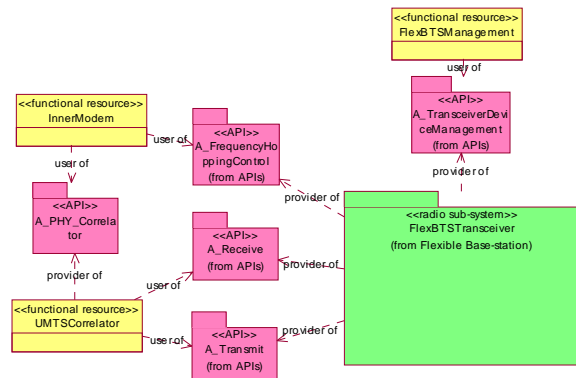


Figure 4 – Example of a waveform functional design with focus on Transceiver sub-system usage

The FlexBTSTransceiver is provider of the following APIs: FrequencyHoppingControl, Receive, Transmit and TransceiverManagement, respectively used by the functional resources InnerModem, UMTSCorrelator and FlexBTSTManagement.

The functional design has defined an additional API PHY_Correlator used by InnerModem and provided by UMTSCorrelator that is not directly impacting the Transceiver.

3.2. Flexible Transceiver functional specification

In this viewpoint, the target is the Flexible Transceiver sub-system, itself part of an industrial Reconfigurable Radio Equipment product.

The Transceiver Facility is used to directly specify requirements of the Flexible Transceiver:

- The Features it supports,
- The values of dimensioning parameters it supports (attributes of the Feature classes), with fixed values, set of values or intervals.

Such direct specifications can be efficiently completed in referencing the Waveform requirements towards Transceiver identified in pre-existing Waveform functional designs.

Those tasks can be realized as a formal PIM in UML, capturing all the capabilities of the specified Transceiver, while still abstracting its implementation choices.

3.3. Flexible Transceiver implementation design

The Flexible Transceiver implementation is assumed compliant with the functional specification realized in § 3.2. Design assumptions are completed by the implementation specifications of the product. From this complete set of requirements, the design process takes place.

Internal architecture of Flexible Transceiver is defined: transposition technologies, digitization method, digital up and down conversion technique, oscillators, etc. are defined.

Computational resources are defined, with their development languages, tools and operating environments. A computational resource may host both Flexible transceiver treatments and Resources implementations. The software connectivity layers between the computational resources are defined.

The logical APIs of the Flexible Transceiver identified at functional specification stage are mapped accordingly.

Software implementation of the logical ports are realized using a formal UML PSM (Platform Specific Model). The used technical services (scheduling and connectivity) are modeled, the logical interfaces are translated into the implementation languages, detailed specification of the implementation ports are done.

3.4. Waveform implementation design

From previous step, the ports available to connect a waveform resources to the Flexible Transceiver are known, with the associated syntax and semantics, the development environments, the resource sharing constraints, etc. These assumptions enable to carry on the implementation design of the waveform resources that will be connected to the Transceiver sub-system.

Illustration

Figure 4 presented a UMTS waveform functional design extract, where three functional resources were interacting through standardized APIs with an abstract Transceiver sub-system.

The outcome of application of the steps described in § 3.2, 3.3 and 3.4 from Figure 4 assumptions is summarized in figure 5.

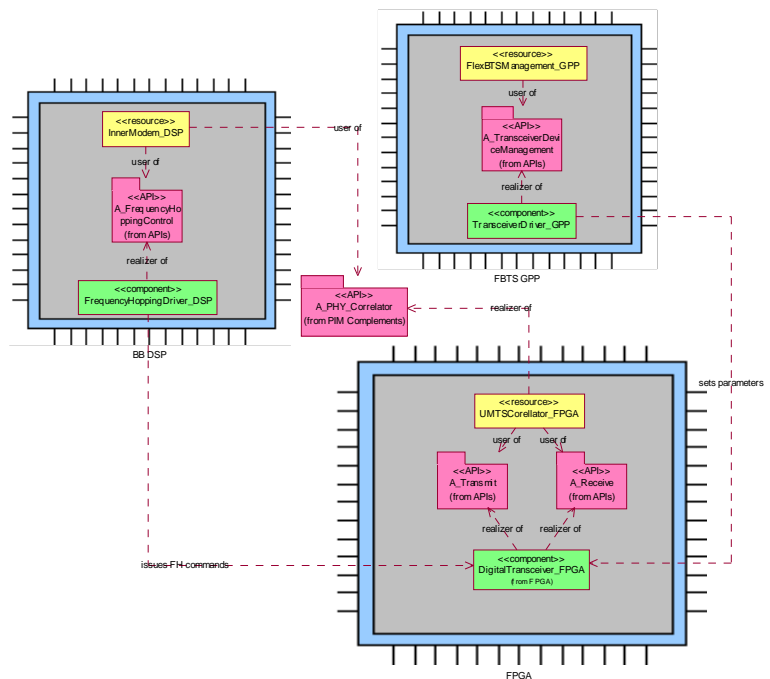


Figure 5 – Example of Waveform implementation design

In this example, the implementation choices introduced three computational resources: a DSP named “BB DSP” (base-band DSP), a FPGA named “FPGA” and a GPP named “FBTSGPP” (flexible BTS GPP).

The PIM logical resources InnerModem, UMTSCorrelator, and FlexBTSMManagement were respectively mapped into resources InnerModem_DSP, UMTSCorrelator_FPGA and FlexBTSMManagement_GPP, and deployed on the corresponding computational resources.

The Flexible Transceiver design identified three components impacting the selected APIs: DigitalTransceiver_FPGA, making signal processing treatments in the FPGA, complemented by FrequencyHoppingDriver_DSP and TransceiverDriver_GPP, which are handling the interactions between the resources and the FPGA, providing the corresponding control APIs at GPP and DSP level.

The PIM APIs are thus to be implemented either internally to the computational resources (A_FrequencyHoppingControl, A_TransceiverDeviceManagement, A_Transmit and A_Receive) or in a crossed solution between DSP and FPGA (A_PHY_Correlator).

4. FEATURES

The identified Features are presented in this section, without entering into their detailed specification. The list and detailed content will certainly evolve until the revised submission to Digital IF is realized.

4.1. TransceiverDeviceManagement

This feature enables a Transceiver to be managed in compliance with either OMG SWRadio Spec or SCA. The compliance with the interface Device is mandated, precisions on the stat machine are given.

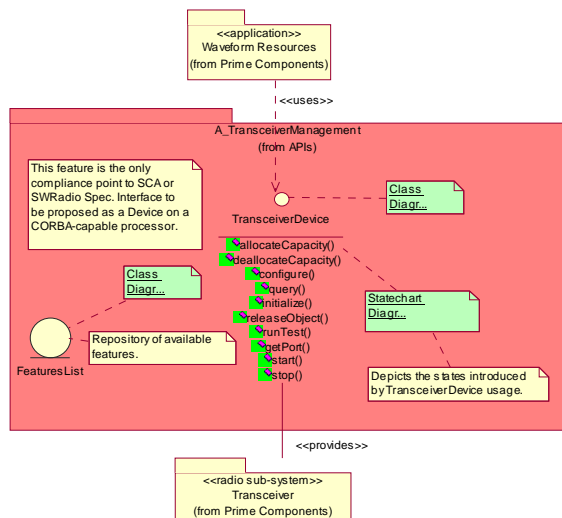


Figure 6 – Content of TransceiverManagement

The allocateCapacity() and configure() operations are used to set attributes identified by the Facility to appropriate values.

The exact property type (allocateable, reconfigurable, queryable, etc.), unit, minimum values range, representation type, etc. of each Facility attribute is set during the binding process to the particular specification.

Such choices will affect the meta-data of the waveform application or the reconfigurable platform.

4.2. Transmit

This feature enables to express the transmission capabilities of the Transceiver.

Namely specifies interface PacketTransmit in case of packet transmission of the base-band samples, and mandates usage of a transmission channelization mask.

4.3. Receive

This feature enables to express the reception capabilities of the Transceiver.

Namely specifies interface PacketReceive in case of packet reception of the base-band samples, and mandates usage of a reception channelization mask.

4.4. HalfDuplexControl

This feature enables to accurately control the duplex state of the Transceiver. The interface DuplexControl and associated state machine is namely specified in this feature.

4.5. FrequencyHoppingControl

This feature enables to control the frequency hopping of the Transceiver. The interfaces DwellSetting, DwellTuning and FrequencyFeeding are namely defined in this feature. Figure 3 is illustrating the content of the Feature.

4.6. AsynchronousTxFlowControl

This feature enables to have an asynchronous exchange mode in transmission between a transmitting physical layer resource and the Transceiver. It namely defines the interface flowSignalling that enables the high / low watermark information to be given by Transceiver to the transmitting resource.

4.7. RxAdaptiveGainControl

This feature enables to specify the expected Adaptive Gain Control behavior of the Transceiver. It namely defines the associated classes and optional real-time control interfaces. Content of current version of the Facility is not mature yet.

4.8. TxAdaptiveLevelControl

This feature enables to specify the expected Adaptive Level Control behavior of the Transceiver. It namely defines the associated classes and necessary real-time control interfaces.

4.9. ErrorsManagement

This feature specifies how a compliant Transceiver issues warning and errors information to the resources having subscribed to the defined interfaces.

5. COMMON CONCEPTS

The identified Common concepts are presented in this section, without entering into their detailed specification. Some examples are given for illustrative purpose.

The list and detailed content will certainly evolve until the revised submission to Digital IF is realized.

5.1. BaseBandSamples

Characterization of the base-band sample flow carried at the data path interface level. The classes BBSamplesStream

(stream general features), BBSamples (a buffer of consecutive samples) and BBSample (the elementary sample) are namely defined in this concept.

5.2. Channel

Characterization of the expected signal channelization, using a transfer function mask. The classes ChannelMask, GroupDelayMask and SpectrumMask are namely defined in this concept.

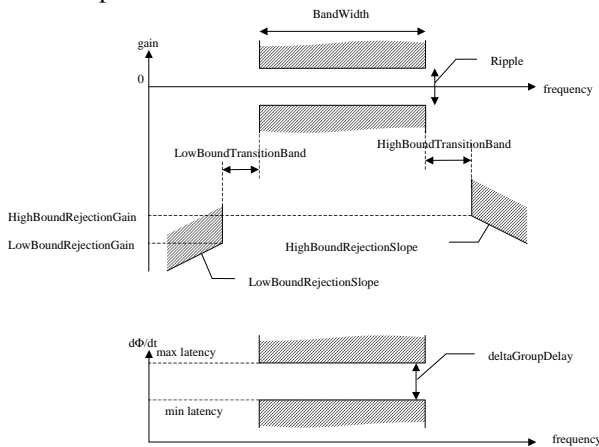


Figure 7 –ChannelMask definition

5.3. Frequency Hopping

Characterization of Frequency Hopping real-time control mechanisms. The classes DwellProfile, FrequencySet and LevelSet are namely defined in this package.

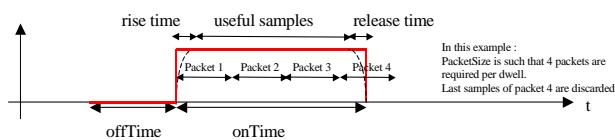


Figure 8 – DwellProfile definition

5.4. InteractionTypes

Characterization of Interaction Types possibly attached to any logical port or interface connecting the Transceiver to an external entity. The classes AsynchronousInteraction, SynchronousInteraction and BalkingInteraction are namely defined in this concept.

Once finalized, the content should be moved to the OMG SWRadio Specification UML Profile, since not dedicated to Transceiver.

6. CONCLUSIONS

This paper has presented the nature, usage and content of the Transceiver Facility PIM developed as the basis of forthcoming OMG standard Digital IF. Several developments can be imagined from this basis.

On the Transceiver Facility itself, the structure (e.g. introduction of a “Binding support” package) and the detailed content (Features and Concepts) need improvement. Extensions through introduction of features related to Spectrum Aware Cognitive Radio or Smart Radios can be realized. The qualification or certification of Facility compliant implementations is an important matter. The addition of Transceiver Facility toolboxes in development tools would facilitate implementations.

Utilization of the Facility for specific waveforms can become normative complements to the Digital IF standard. Similarly, bindings for domain-specific Flexible Transceivers can be added as normative complements to the Digital IF specification.

PSM mappings of the Features are necessary normative complements, to be structured across main possible development languages, and completed by standardized ports implementations for the most popular component models (e.g. the one of OMG Software Radio Specification [REF]).

Last, the paper paves the way to definition under the same methodological approach of other critical Radio Sub-systems (e.g. Power Amplifier, Time Machine, ...).

7. ACKNOWLEDGMENT

This work has been performed in the framework of the EU funded project E2R II. The authors would like to acknowledge the contributions of their colleagues from E2R II consortium.

8. REFERENCES

- [1] OMG Software Based Communications DTF, “Request for Proposal PIM and PSM for Digital Intermediate Frequency Interface”, *OMG Document*, sbc/2004-08-15, 2004.
- [2] “End-to-End Reconfigurability Phase II Integrated Project”, <http://www.e2r2.motlabs.com/>.
- [3] E. Nicollet, “E²R commercial Reconfigurable Radio Equipments: from advanced research to proof-of-concept and Standardization”, *SDR Forum Technical Conference*, 2006.
- [4] E. Nicollet and C. Serra, “SCA 3.1: Transceiver Sub-system APIs”, *SCA Change Proposal*, CP 277, January 2005.
- [5] T. Demirbilek and E. Nicollet, “Joint Digital IF Initial Submission”, *OMG Document*, sbc/05-03-01, March 21 2005.
- [6] OMG DTC, “Platform Independent Model (PIM) & Platform Specific Model (PSM) for Software Radio Components”, *OMG Specification*, dtc/2005-09-05, September 2005.
- [7] JPEO JTRS, “Software Communications Architecture Specification”, *version 2.2.2 / FINAL*, 15 May 2006.