The Software Communications Architecture

Claude Bélisle
Research Manager
Military Satellite Communications
Communication Research Centre
claude.belisle@crc.ca
SCA - A Paradigm shift

• Decouples hardware, software and system integration functions
  – Facilitates acquisition process by eliminating stovepipe systems

• Promotes re-use of signal processing software modules
  – Modulator, demodulator, encoder, interleaver, FFT…
  – Reduces application development cost

• Open framework architecture
  – “Glues” the software and hardware
  – Facilitates application and module portability
SCA Core Framework

- Central radio software piece, the “operating system”
- Provides an abstraction between software and hardware
  - Defines interfaces, behavioural specifications and general rules to support devices and application portability
- Based on commercial standards
  - X.731 ITU/CCITT OSI System State Management
  - CORBA
  - Posix (Portable Operating System Interfaces)
  - CCM (Corba Component Model)
- Designed to meet commercial as well as military application requirements
SCA Design Concept
SCA Core Framework

• The Core Framework consists of:
  – Base Application Interfaces
  – Framework Control Interfaces
  – Framework Service Interfaces
  – Domain Profile

• It specifies a life cycle for the signal processing modules to be downloaded on the hardware:
  – Load
  – Initialize
  – Connect
  – Configure
  – Execute
  – Terminate
  – Unload
  – Release
Building a Reference Implementation

• What is a Reference Implementation?
  – Open-source software
  – Defines the behavior of the specifications
  – Codifies all of its relevant technical aspects

• Benefits of RI
  – Reduces the level of ambiguity of the SCA specifications
  – Increases the potential for interoperability
  – Increases understanding of the architecture through an example
  – Reduces the cost and time-to-market of SDRs
CRC and SCA-RI

- Active member of SDR Forum
  - Participated in the development of the SRA
  - Involved in SCA technical discussions
    - Introduced the concept of Ports to enable true modularity of software components

- Developed a PoC Software Defined Radio
  - FM Line-of-Sight
  - SCAv0.3
  - In C++ on Digital Signal Processors (DSP)

- Realized the need for an Open Source Reference Implementation
  - Proposal to SDR Forum to promote commercial adoption
  - October 2001
SCARI (1)

• Implementation
  – SCA version 2.1
  – Mandatory features
  – Written in Java for portability and ease of comprehension
  – Includes a simple waveform example

• Partners
  – Implemented in collaboration with DRDC – Ottawa
  – Sponsored by the SDR Forum
SCARI (2)

- **Product**
  - 60,000 lines of code, 300 pages of documentation
  - Peer reviewed
  - Available at www.crc.ca/rmsc or www.crc.ca/scari
  - More than 7000 downloads from worldwide organizations
  - 37 000 hits since June 2002

- **By-product**
  - CRC submitted 21 technical change proposals to JTRS / JPO in reference to SCA version 2.2
Impact of SCARI

• **Opened the door to new players**
  – No longer limited to the majors
  – Decoupled Hardware / Software / Waveform development

• **Facilitated the emergence of new markets for SDR concepts**
  – Radar processing
  – Medical imagery
  – Other signal processing intensive applications

• **Transformed the waveform development approach**
  – Modularity at component level rather than applications
Waveform Development Vision

- **Current Approach**
  - Extension of conventional techniques
  - Single monolithic block defining the application
  - In this case, the waveform is the application

Taken from JTRS API Supplement
Waveform Development Vision

• Considering that software cost is:
  – 20% development
  – 80% maintenance

• Development approach strongly promoted by CRC
  – Reduce the granularity of the software components
    • Similar to specialized chip sets in board design
    • Simplifies debug and maintainability
    • Facilitates reuse of components between applications
Waveform Development Vision

- A waveform is composed of many applications
- Each application is composed of many signal processing modules (resources)
Application Example
Digital Audio Broadcast

- Physical Layer of the DAB receiver application containing 12 resources
CRC Waveform Application Builder

DAB Example

1024 pts FFT
D-QPSK decoding
Freq Deinterleave
QPSK Demapping
Block Decoder
Time Deinterleave
Viterbi Decoder
ICS-652 A/D-DDC Device
Mpeg Player Resource
Audio Device
Time & Freq Sync

A/D Converter Device → Time & Freq Sync → 1024 pts FFT → D-QPSK Decoding

Block Deinterleave → Block Decoder → Q-PSK Demapping → Freq Deinterleave

Time Deinterleave → Viterbi Decoder → MPEG player → Audio Device
Connecting Applications

• Connections between applications is however required
  – This is supported by the SCA but mechanism not flexible enough, requires hardcoding

• There is a need to transpose the resource connection mechanism to the application level
  – CRC will submit a change proposal to JTRS/JPO
  – Paper to SDRF conference to be published
Following the RI

- CRC continues to develop software to promote the expansion of the Software Define Radio
  - SDR Development Tools
    - Waveform Application Builder (WAB)
    - Radio Manager
    - Node Boot Builder
  - SCA Core Framework v2.2
    - Java
    - Hybrid
    - C++
CRC SCA Core Framework v2.2

• **Java**
  – Extension of SCARI
  – Low cost
  – Most valuable for training
  – JTEL certification would be important for public release

• **Hybrid**
  – Java for management functions, C++ for signal processing
  – Easy to maintain Domain Manager
  – Allows development of devices and resources in C++
  – Mid-range cost
  – Applicable to embedded platforms with single board computer running Java virtual machine
CRC SCA Core Framework v2.2

- **C++**
  - Full feature implementation of SCA CF v2.2
  - All C++ implementation
  - Applicable to embedded platforms
  - High cost
Thank You

claude.belisle@crc.ca