USING CORBA ON DSPS, FPGAS AND MICROCONTROLLERS – SYNTHESIZING A UBIQUITOUS “SCA MACHINE” IN SOFT-RADIO DEVICES

Shahzad Aslam-Mir: PrismTech, San Diego, CA., USA.,
Mark J. Glenn: PrismTech, Burlington, MA., USA

ABSTRACT

Software Defined Radio (SDR) application developers increasingly want to exploit the performance and power of different combinations of modern Digital Signal Processors (DSPs), FPGAs and general purpose processors (GPPs). As such, radios are being realized with different physical topologies, with different combinations and numbers of such processing elements in them today. This paper talks about the consequent motivation for the use of an ‘SCA-machine’ that stretches over all three classes of processor and normalizes the hardware from the perspective of the waveform: with the aim of maximizing hardware independence and providing a common API, supporting waveform “plugability” and portability, etc.

1. INTRODUCTION

Current debate concerning the use of hardware abstraction layers on processors in the radio currently perceived to be non-CORBA-enabled, has given rise to two schools of thought. Thinking is currently split between standards groups, either defining a HAL specification, or using something like the OMG’s software defined radio PSM/PIM model to define these components (and the interfaces they host) in CORBA IDL. The SCA today is one of the more powerful models being used to define the canonical structure of the soft-radio device. However in terms of implementation, SCA makes little stipulation about the use of CORBA on the more exotic elements and platforms in the radio’s signal-processing chain – viz. the DSP and, even more controversial from the ORB point of view, the FPGA.

This paper will detail the advantages of the latest advances in leading-edge high-performance real-time ORB technology (based on MIPS-based normalized design methodologies) which make it possible to extend CORBA support to hitherto non-CORBA-enabled platforms; resulting in benefits not only directly to the radio platform, but also to the waveforms developer.

Perhaps the most active area in the use of the SCA (as required by the US Government in its DoD communications systems) is in the Joint Tactical Radio Systems (JTRS) effort, as run by the JTRS Joint Program Office (JPO). The SCA model purports to use a generic core framework in software, which hosts the running of several SCA-compliant waveforms. This model works well to promote portability in several dimensions, however the motivation for this paper is to solve the duality problem, which stated simply is

(a) how to keep the software architecture portable over a wide range of hardware devices, at level P-h (Figure 1) and also
(b) then keep the overlaid soft waveform independent of the underlying core framework by means of standards, and also by means of suitable generic abstractions so the waveform may be hosted over multiple core frameworks at level P-w (see Figure 1).

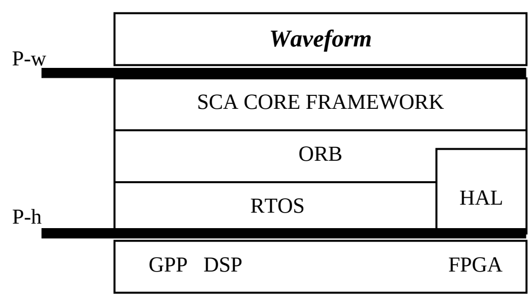

Figure 1: Illustrating the porting ‘duality’ problem

The duality of this problem is what drives the motivation for an “SCA machine” that abstracts away from the waveform designer nearly all knowledge of the underlying hardware, and thus allows these individuals to focus on pure waveform synthesis.

2. THE SCA MACHINE

There is much debate as to whether the use of a HAL/MHAL (or some other form of HAL) is more suitable than the use of say, CORBA on DSP or traditional non-CORBA-enabled elements.

It is the author’s assertion that both are needed at the right levels or layers. A radio component HAL is useful at the lowest levels of the stack, where algorithms and signal-processing elements may need to call any componentized intrinsic of the hardware: say a Viterbi machine on a TI DSP. However, this material need not be apparent at the waveform layers as illustrated in Figure 1, rather it should be part of some SCA element that is assembled by the SCA factories into a waveform's canonical signal processing chain. In addition, the waveform developers should be shielded from specifics of the radio hardware at the waveform level, so as to maximize the portability of said waveform being developed. Thus, the concept emerges of an "SCA machine" which promotes such a notion.

This concept defines a "SCA machine" into which we load only SCA-compliant waveform elemental components via SCA-compliant interfaces. The way in which this concept differs is that the loading of all elements on all targets is done via SCA and no independent hardware abstraction layers (HALs/ and modem or audio HALs) exist anywhere in the radio. There is thus complete SCA capability on the DSP (albeit with a reduced POSIX profile) and also on the FPGA. Furthermore there are no software adapters being used between the DSPs and FPGAs. In order to make such an architecture efficient, the "SCA machine" internally has a *radio-component HAL* which is used to normalize physical differences between radios from different vendors.

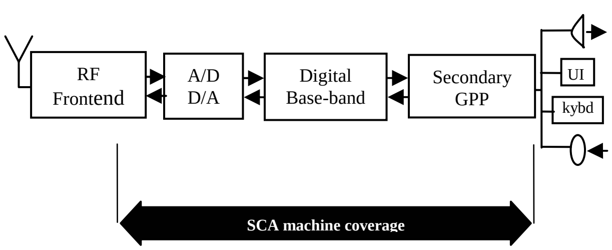

Figure 2: Possible extent of coverage of an SCA machine

3. SCA PENETRATION

In JTRS circles, where the use of SCA is mandatory, the key question under debate today is where SCA capability should, and should not, be in the radio. The focus seems to be mostly around the processor types – e.g. should there be SCA/ORB capability on DSP and/or on any FPGAs that form part of the physical hardware of the radio. In addition should there be SCA capability in the RF-IF areas. Further, with the advent of digital RF chips now becoming available should, or rather could, SCA control be stretched closer to the antenna? It is our assertion that the provision of a uniform set of SCA APIs over every area, by wrapping all the hardware in lower level HAL APIs that are hidden under the SCA layers, be proposed.

4. CURRENT STATE OF SCA-RADIO ART

The current SCA specification has recently been added to in the form of a "Hardware Supplement", which is the beginning of a move to normalize the hardware heterogeneity and harmonize the SCA APIs - to maximize their value. As SCA is derived from the IEEE's POSIX, and the OMG's CORBA specifications it is in some sense a living document. This is beneficial owing to the fact that the OMG members are constantly feeding back both implementation and usage experience that helps the specifications to evolve. In this users group is, of course, the entire JTRS community, which is actively involved in building SCA-compliant radios, core frameworks and, of course, waveforms.

In parallel to the very active JTRS community, the member companies of the OMG (who are also part of that JTRS community) participating in the OMG's software based communications (SBC) group have produced a comprehensive and detailed canonical model of a soft radio device. In the spirit of the OMG's model-driven architecture, this model is referred to as the Platform Independent Model (PIM) [12]. From this are taken mappings which many or may not map the entire PIM into particular technologies domains. Future versions of SCA may be thought of as mappings of the PIM. The problem in the current state of the practice tends to be the use of HALs and MHALs at the SCA layer which has the effect of tightly-coupling waveform code to the physical topology of the radio, making hardware and waveform independence more difficult.

5. DERIVATION OF A CANONICAL RADIO SCA-MACHINE

The strength in a model such as the OMG's SBC PIM lies in the fact that the PIM may be mapped even into non-CORBA domains, as the mapping is really to a 'technology domain'. In light of such an architectural paradigm, an "SCA machine" can be realized as a mapping of part of the PIM adapted to suit particular domains of soft-radio usage. These may be partial mappings for say, particular physical types of device (e.g. handheld device, as opposed to a device that may be rack mounted). We provide as a partial list some of the many benefits that can be realized if such an "SCA machine" architecture is realized:

Greater degree of SCA compliance and thus shorter verifiability and certification risk – Use of a pure “SCA machine” means that nothing can be loaded into the radio other than pure SCA media, i.e. since the only way into the radio is via the “SCA machine”, nothing non-compliant can be loaded and run. In addition, if SCA modules only can be loaded, they can be ‘securely verified’ (much like a Java VM verifies byte-code and treats it in the sandbox model). Further, if SCA modules are loaded and aligned in the signal-processing chain via the “SCA machine”, the separation of red-black and crypto modules can be more effectively controlled as well (perhaps even without the need for extensive and complicated infrastructure such as MILS/MLS), as the “SCA machine” is acting as the portal entry for any and all media into the radio. Such strict control of SCA also means that a lot of compliance issues, that would traditionally only surface at verification and certification time, would have been addressed inherently.

Shortening certification, acceptance testing and final time-to-market of future SDR implementations is a benefit realized by using an “SCA machine”. This is due to the fact that it should be easier to reconfigure the device quickly, given the greater degree of fine-grained re-distributability of objects in the signal-processing chain of the radio. This is because the components can be reconfigured more simply as they all operate on one level – the SCA level as depicted at the layer P-w in Figure 1.

Higher degrees of waveform portability – This is possibly one of the most important aspects of SCA. Typically the waveform has elements that run on the GPP and others that run on the modem and DSP. In addition it may utilize elements on any FPGAs in the radio. The mapping of the waveform’s components to different parts of the core framework must usually be done in a manner that permits the waveform to remain decoupled from the core-framework and thus from having to have any a-priori knowledge of the physical structure of the radio. Traditionally this has not been the case, with waveform code having extensive details of the physical componentry of the radio written in or hardwired into the waveform code. This then means that the waveform code is no longer portable between any two radios of different structure. A ubiquitous “SCA machine” that hides HALS and MHAL details from the waveform, yet permits them to be used effectively, offers benefit in terms of simplicity of development of the waveform and thus reduced development risk.

Intelligent and fine-grained radio resource management – An “SCA machine” that makes extensive use of real-time CORBA semantics internally (e.g. control of memory, concurrency, and CPU use) and is capable of controlling an end-to-end set of transformations on a bit stream, like that in digital base-band, offers tremendous potential to not only experiment with many different permutations of waveform, but also permits a more effective wide-reaching tuning and reconfiguration capability throughout the device. This “SCA machine”, capable of running on GPP, FPGA and DSP alike, offers the waveform developer greater freedom to optimally and seamlessly, position waveforms elements over different parts of the radio; to thus get the best performing configurations. This is especially possible today, now that there are advanced techniques to synthesize pluggable, dynamically-loadable solutions (for Xilinx FPGAs for instance). In addition, reduced minimal CORBA Services, coupled with intelligent radio resource management (with feedback and control loops implemented via ORB) offer a powerful solution. Intelligent radio resource management is one solution which uses the SCA resource manager and adds to it the notion of power and CPU MIPS control.

Independence from waveform language of implementation – Of key concern is the supportability of legacy waveforms on new JTRS radios. More importantly, some of these need to be converted to be SCA compliant. As an “SCA machine” only loads waveform elements at the P-w boundary (see Figure 1 again) which has defined IDL interfaces, the language of implementation (e.g. ADA) of the waveform can be different and still be able to operate over the “SCA machine” owing to IDL’s language neutrality.

Freedom to be able to make the most appropriate choice of CORBA language mappings in the right environment – An “SCA machine” hides from the waveform developer the complexities of the radio’s internal APIs, whilst allowing, for example: the use of low-footprint languages (such as C or wrapped assembler modules) on Digital Signal Processors (DSP); C-wrapped portable VHDL on FPGA platforms; C++ on General Purpose Processor (GPP) platforms, and Java on the HMI (man-machine) interfaces of the devices. Thus, allowing the most effective use of development resources and skill-sets in the right places. So yielding a shorter time-to-market.

Freedom to experiment with alternative mechanisms and topologies when mapping waveform elements to the SCA CF (over different types of platform and chipset) means that traditional design-time decisions may be happily delayed until run-time. Various elements of a radio’s signal-processing chain, e.g. modulators, inter-leavers and suchlike, may be spread or dispersed over the radio’s processing elements on more than one configuration. An “SCA machine” that is ubiquitous means that different modes and configurations can be tried more quickly and an optimal setup reached. In addition dynamic re-configurability becomes easier also.

Power consumption in battery-operated SCA devices can be dynamically tuned and re-optimized (in response to changing conditions) more readily in an environment where tasks run can be quickly re-balanced between the General Purpose Processor (GPP) and the DSP. The DSP often consumes significantly less power to perform the same algorithmic function as would a GPP.

In the long term any device running an "SCA machine" would help in producing a lower-cost radio as design-time decisions that lock radio developers into extremely costly design-cycles (based on object and processing layouts being hardwired) can now be open and allow a re-balancing of the object and processing distribution throughout the radio "on the fly".

6. EVOLUTIONARY TREND TOWARDS AN "SCA MACHINE"

Current industry practice shows that there is a trend to produce some degree of hardware independence by providing many different type of hardware abstraction layer, such as modem-HALs, audio-HALs and suchlike (xHAL). In addition, the emergence of newer DSP and FPGA middleware environments [9] has permitted several alternate approaches that build upon the OMG's PIM model [9]. Figure 3 illustrates that, over time, the importance of the ORB diminishes to that of a middleware enabler; perhaps what becomes more important is the core framework and its extent and range of coverage over the various radios elements. What is critically important is that the portability layers at P-w and P-h remain the same to maintain the duality of portability over different hardware.

It is currently common to make use of adapter technologies on the GPP to proxy over to the DSP and GPP in SCA implementations. On the DSP there may be component HALs and on the FPGA specialized GIOP machines (in gates) that furnish access to special algorithm machines that are used by the waveform. Access to these occurs via multiple adapter transformations in a "to and fro" manner, as the call context leaves and re-enters the SCA world. The DSP and FPGA in this situation are treated as non-SCA-enabled and non-CORBA-enabled elements. This means that waveform code running on these elements is tightly-coupled and highly-dependent on these processors and this waveform will not work for another radio that doesn't absolutely have the same physical topology of processing elements. This is clearly a major obstacle to waveform portability/reuse and hardware independence. The SCA influence (and thus benefits) only resides on the GPP.

What is proposed here is that the HALs and specialist environments of the FPGA be subsumed into the "SCA machine," which then presents only an SCA interface to the waveform elements; now the waveform elements are independent of the physical radio and only dependent on the "SCA machine", which must, by definition, furnish the appropriate APIs to the waveforms and operate and end in the proper specification-prescribed manner.

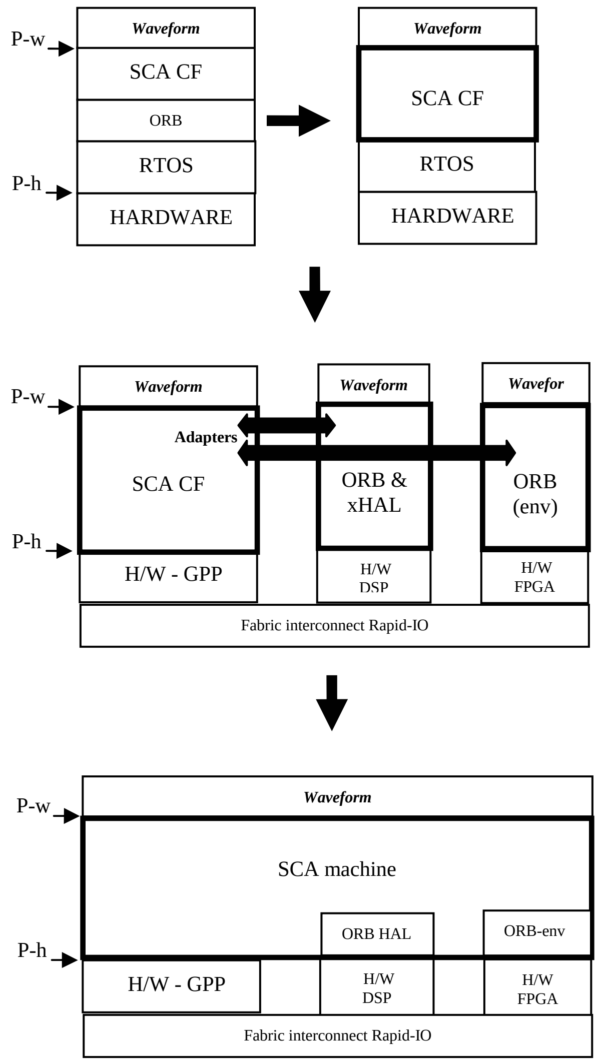

Figure 3: Illustrating the evolutionary path to an SCA machine

7. KEY TO SYNTHESIZING THE SCA-MACHINE

To synthesize the "SCA machine" it is necessary to normalize the hardware in a manner such that a waveform sees nothing but pure SCA interfaces - upon which it must load its components. This means that pure IDL-based CORBA interfaces would appear over every platform and modem as a standard. The single most decisive factor or property of a CORBA implementation to support this realization is quite simple… It is usually expected that a CORBA implementation needs to provide:

- a very small, dynamic and static memory footprint.
- a user-configurable and re-tunable, fine-grained footprint control mechanism and
- a highly deterministic core request dispatch engine

But, these features alone are insufficient (an incomplete measure of the implementation). **It is fundamental to provide these features at the absolute minimal computational cost.** Herein lays the key to making a "SCA-machine" that spans the entire device a reality. The computational cost in this case is measured not in processor speeds, on and off processor memory, or the more traditional CORBA transport and ORB latencies, but rather a more elemental and comprehensible measurand for SDR implementations. This is the notion of the number of CPU instructions per KB of data marshaled, normalized against processor speed. Although there is contention in this statement, as processor architectures may be different, the use of MIPS or MOPS gives us a tangible measure of the cost of achieving something. This single measure encapsulates a concise, yet precise detail of how efficient an ORB implementation can be on a given platform, and thus how efficient the SCA soft-machine will be on the device. In addition, it gives us a measure of how long a battery will last and how heavy and large therefore the device may possibly need to be.

8. CONCLUSIONS

Perhaps the most important issues in SCA development today revolve around standards compliance in order to achieve true waveform portability across all elements of the radio. This paper has shown that the use of a ubiquitous "SCA machine" across the radio unburdens the waveform developer from having to have intricate knowledge of the physical radio, and thus yields substantial cost, time, re-use and flexibility benefits.

9. REFERENCES

[1] Raisanen A.V., Lehto A., – *Radio Engineering*, Artech House 2003.

[2] Noble J, Weir C., – *Small Memory Software*, Wesley 2000.

[3] Proakis J.G., – Digital Communications, 4th Ed. McGraw Hill.

[4] Mitola J., – *Software Radio Architecture – Object oriented Approaches to Wireless Systems Engineering*, Wiley 2000.

[5] Reed J.H., – *Software Radio, A modern approach to Radio Engineering*, Prentice Hall 2002.

[6] Johnson C.R., Sethares W.A., – *Telecommunication Breakdown – Concepts of communication transmitted via software defined radio*. Pearson Prentice Hall 2004.

[7] Dohse D., Bush L., Osborne G., Christensen E., – *"Successfully introducing CORBA into the signal processing chain of a software defined radio"*, COTS Journal, January 2003.

[8] Fette B., LaMacchia M , Christensen E.,– *"High Performance Software Radios"*, April 2004.

[9] PrismTech Corporation, 2003, 2004 – *"Use of high performance DSP middleware for SCA based SDR-OEs."* www.prismtechnologies.com.

[10] Smith S. W., – *Scientist and Engineers Guide to Digital Signal Processing*, California Technical Publishing 1997. www.DSPGuide.com.

[11] Sklaar B., *Digital Communications, Fundamentals and Applications*, Prentice Hall 2000.

[12] Bickle J., Smith J., et al., – *OMG Specification for Software based communications architecture – PIM and PSM for Software Radio Components* Joint Revised Submission, OMG document SBC-04-04-01. http://www.omg.org/cgi-bin/doc?sbc/04-04-01.