THE APPLICABILITY OF SCA IN THE DESIGN OF HIGH DATA RATE (HDR) JTRS RADIOS AND WAVEFORMS

Davis, W., Comtech EFData, Tempe, AZ, USA; wdavis@comtechefdata.com;
Aslam-Mir S., DataSoft Corp., Scottsdale, AZ, USA; shahzad.aslam-mir@datasoft.com

ABSTRACT

The JTRS SCA architecture is designed to promote reuse of software elements that are primarily GPP and in limited instances, DSP bound. The SCA provides a uniform harmonizing foundational platform upon which waveform software element reuse becomes more attractive.

The development of new, or, the updating/hosting of existing high-bandwidth waveforms to enable them to operate on next generation high data rate SCA modem platforms, creates challenges in the radio design arena. The use of SCA for the design of high data rate modems to support high bandwidth waveforms (e.g. those operating above 2Ghz terminals) is a relatively new area of endeavor. Such platforms are designed to furnish substantially higher data rates necessitating large-scale use of FPGA based (as opposed to GPP and DSP) based hardware. The use of SCA in such cases is limited therefore owing to its limited coverage on metrics and guidelines on how to develop reusable SCA compliant waveforms. Recent efforts such as the USAF's High Data Rate - Radio Frequency (HDR-RF) Modem program have served to highlight how the SCA specification needs enhancement and to provide guidance to the designers of such platforms. In essence SCA needs to bridge the gap from what has been a traditional general-purpose processor reuse strategy to a more FPGA centric approach so as to meet the challenges designers face. This paper presents a brief discussion on the new potential approaches and benefits with a view to promoting greater waveform reuse, portability, shorter porting schedule and ultimately reduced cost.

1. INTRODUCTION

In classical approaches to SCA design it is the norm to have SCA devices modeled to represent either unique distinct pieces of hardware, or, parts of some larger element that may be both hardware and software in some form of hybrid configuration. At a more canonical level these devices are CORBA peer programs executing in their own process spaces in a GPP. The GPP runs a COTS stack composed of BSP, OS, ORB and an SCA compliant Core Framework (CF) as shown in Figure 1. When the processor is something other than a GPP such as for example a DSP or an FPGA the synthesis is more involved and may involve the use of addendum specifications from the SCA. Examples of these are items such as the MHAL specification or a modicum of other proposals to the SCA that have not actually become part of SCA in their own right. We shall discuss these later.

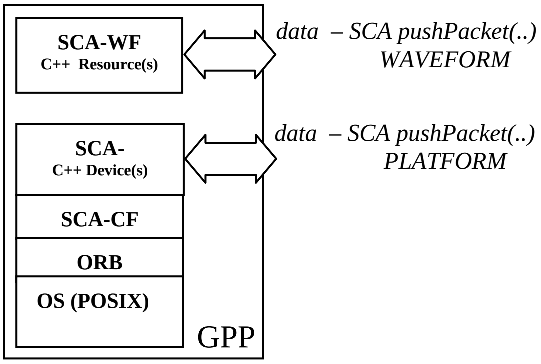
SCA-WF
C++ Resource(s)
SCA-
C++ Device(s)
SCA-CF
ORB
OS (POSIX)
GPP
data – SCA pushPacket(..) WAVEFORM
data – SCA pushPacket(..) PLATFORM

Figure 1: ***Classical SCA synthesis to realize a JTRS SDR***.

The model of Figure 1 is a good one, and has been extended to allow for the design to accommodate DSPs and FPGAs. The basic assumption however in this approach is still that it is composed of a reasonably heterogeneous processor landscape. However for HDR platforms this model breaks down and strictly speaking does not really hold true for High bandwidth, High Throughput (HBHT) SATCOM modem platforms.

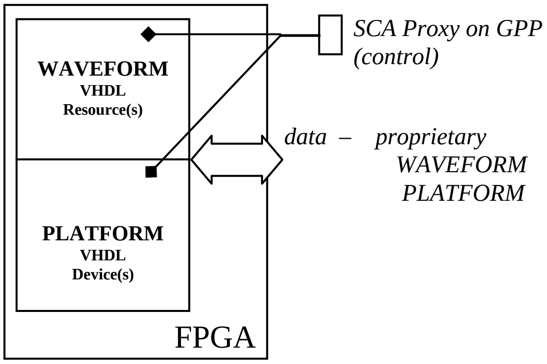
WAVEFORM
VHDL
Resource(s)
PLATFORM
VHDL
Device(s)
FPGA
SCA Proxy on GPP (control)
data – proprietary WAVEFORM PLATFORM

Figure 2: ***HBHT Modem SCA synthesis to realize a JTRS SDR***.

HBHT platforms are composed of significantly more computing engine power that is typically implemented in GPP or DSP based ones. This means that the classical data traffic elements that constitute C++ SCA Resources and

Device executables typically hosted on the GPP processing elements are not portable or reusable on the FPGA. This is shown in the model in Figure 2. This does not mean that SCA cannot be hosted here, nor does it mean that the reuse benefits SCA brings cannot be realized. In this paper we discuss a potential implementation of SCA in a FPGA-centric environment focused on reuse and one that tries to separate platform specific interfaces from waveforms specific interfaces so as to promote the greater potential for waveform material reuse.

2. STATEMENT OF PROBLEM

The problem can be stated quite simply – ‘*implement an SCA-compliant SATCOM modem platform composed almost solely of FPGAs which shall be capable of high data rates supporting a variety of SCA compliant waveforms hosted on it’*. While this is a simple statement, its implementation is a much more challenging. There are a number of hurdles to overcome, starting with the demands on the digital section of the modem platform. It is typical to see multiple FPGAs woven together with some form of control interface and signal path interconnect. As of yet there are no industry standard interfaces between these devices that are widely accepted, indeed the closest are those that are used in SoC design. The two main challenges using an SoC approach are – firstly, there is currently no standardized FPGA device interface in JTRS circles – no agreed upon standard. Nor is there a vendor -neutral set of metrics that can accurately predict or report on a waveforms resource utilization of gate count, RAM block and logic slices. Secondly, the practicality of the port; in a multi-FPGA device environment, will the functional mapping be the same or will it require additional porting effort to remap? As an example at higher data rates, parallel processing of core functionality (such as the FEC) is required. How will this parallelism be mapped across different FPGA based hardware platforms?
Another question that forms part of the problem is when porting multiple FPGA type waveforms into a single hardware platform what should be the focus? one school of thought suggests attention on the hardware abstraction layer to enable the designer to match the waveform implementation to the hardware design using a set of agreed upon interfaces.

3. KEY CHALLENGE HOSTING SCA ON FPGA CENTRIC MILSATCOM PLATFORMS

The real challenge is to get industry and customer buy-in on a *single unified approach to a radio that has a mixed GPP-FPGA infrastructure* for synthesizing this type of radio. This is problematic at this time because there are a number of camps that have used a variety of methods and achieved success to varying degrees. In addition we lack at present any empirical data, or comparative studies that subjectively analyze the various approaches proposed by industry today. Lastly, there has been no *benchmark* test that can be performed on these HDR platforms to aid in the evaluation of the platform on an equal basis until recently, and so measures of reuse cannot be quoted meaningfully until these 3 factors reach some consensus amongst practitioners. Thus, there arises a real need for a universal SCA Component Interface Description that can be implemented at low overhead, latency and cost; is location transparent (like CORBA), and provides interoperability, portability, and reuse in an FPGA environment. We refer to this interface as the Universally Addressable Interface (UAI).

4. HOSTING A CORE FRAMEWORK ONTO AN FPGA CENTRIC RADIO PLATFORM

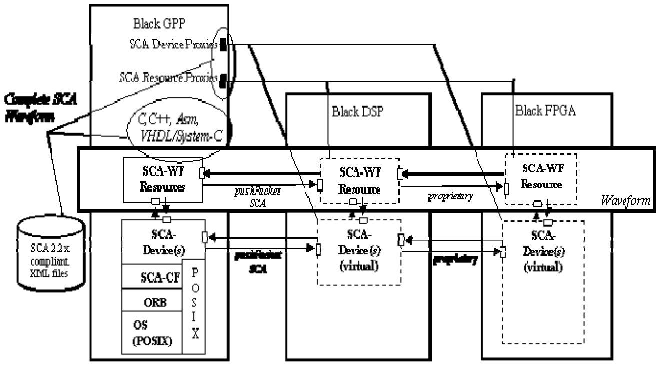

Figure 3: *Classical use of SCA with heterogeneous processing element Note waveform uniformly distributed across elements.*

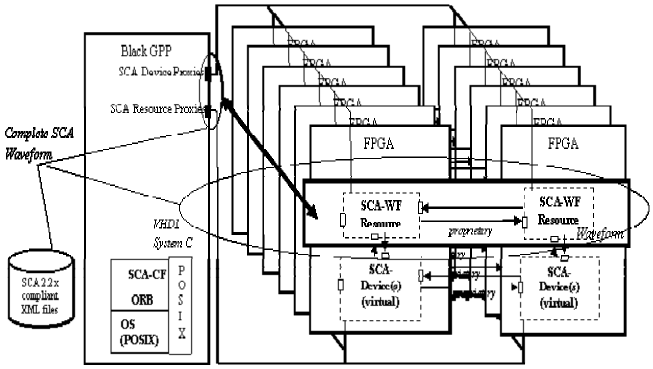

Figure 4: *Synthesizing SCA on HDR FPGA dominant processing platform. Note waveform concentrated in FPGAs*

Hosting an SCA compliant Core Framework on an HDR is a relatively simple task as its runtime can be supported on a GPP in the radio. This of course makes the assumption that a GPP shall be furnished in the radio with sufficient capability with the correct and necessary OE composition. However, a GPP is not necessarily the answer. Another approach is to run an SCA infrastructure on a core in one of the FPGAs in the radio and not use a GPP at all. This is a real possibility and requires a CPU core be co-hosted in one of the FPGAs so as to be able to run an SCA OE. Both approaches have merit; the final choice usually being a function of the customer's requirements which typically are cost and complexity. Once an SCA radio is launched using the certified processes, there remains the question of how and where the Human Control Interface or *Radio Services* (RS) stack is to be hosted. Radio Services are currently not explicitly part of the SCA and though a vital component of any production SDR are not tested in any JTAP or other certification test suite. It is usual for RS to interface with the SCA core framework so as to allow interaction with the radio. An important reusability concern arises when the interaction of RS with the waveform is considered. And so there arises another architectural force that demands a separation of platform specifics from waveform specifics yet stay loosely coupled to some extent in the guise of radio services interacting with, configuring and controlling an SCA waveform that is active and using the core framework. The use of a model driven approaches to this are beneficial. Waveform modeling can provide specific XML driven user interfaces to minimize the differences between waveforms, which lowers porting risk among different waveforms and nicely abstracts out commonality for UI design.

5. WAVEFORM MODELLING AND HOSTING

FPGA parts of an HDR waveform modeling and design may take place in common tools such as Matlab-Simulink™ or some other model simulation package and then transition to hardware after VHDL or System-C code generation. Usually the data-paths between the blocks on the Simulink™ design palette conceptually are connected together using some proprietary approach such that a single bin file results with the waveform components glued together in one image. It is here that the SCA needs a precise and exacting standard approach. These components need to be standard so as to be able to reuse and port this FPGA design. For SCA to solve this problem there needs to be adoption of *a portable FPGA realization with open standard interfaces that can be called from anywhere*. This problem is not new – it has been attempted before by the engineers working on the SCA in the form of the Specialized Hardware Supplement (SHS) – we discuss some of the approaches that have been proposed

5.1. Logical Device facade with driver connected to bus.

This approach [9] brings an FPGA modeled as an executable or loadable device that is abstracted via a logical device interface on the GPP. The logical device provides all the management and control APIs needed to communicate (in SCA fashion) with the FPGA, but not use any data pushPacket calls. Often this is implemented via a device driver embedded within the logical device that accesses the FPGAs via firmware drivers. The APIs here are usually only the management and control APIs. There is little or no capability to make data passing pushPacket types of calls. This is a common and successful model and has the advantage of insulating the SCA domain from back end hardware modifications and redesigns as the SCA façade only relies on the availability of the device driver. As far as waveform portability is concerned – there are obvious limitations as the FPGA bin file is highly specific to the particular vendors FPGA and any associated cores used in the FPGA code design. Thus there is little capability for a separation of platform logic in the FPGA from waveform specific logic that needs to be decoupled and portable to FPGAs on another vendor's radio that may use FPGAs from a different manufacturer.

5.2 MHAL

The JTRS Modem Abstraction Layer (API) is a fairly recent proposal that was designed specifically to enhance portability to waveform components operating on DSPs or FPGA or a combination of both. MHAL uses a message passing protocol to encapsulate control from a GPP to the DSPs and FPGAs. Waveforms ported to this API can only be ported to JTRS sets that support this API.

5.3 Open Core Protocol (OCP)

The OCP[17] is a new standard that provides a bus-independent, high performance and configurable interface. Once an OCP profile is defined, any device can be interconnected to another as long as they conform to the right OCP profiles. Literature [1] has provided a detailed approach with metrics and reuse capabilities based upon OCP and simulation.

5.4. CP289 – a variation with OCP Profiles

CP289 was proposed as part of the Specialized Hardware Supplement of SCA 3.0 and 3.1. It separates waveform and platform concerns in several distinct and well conceived stages to achieve component portability. The proposal classifies waveform elements into GPP or specialized hardware (SHP) components. SHPs can be either RCC (resource constrained components) or RPL (RTL programmable logic) components such as FPGAs. The part of the proposal that deals with FPGAs recommends using OCP interfaces to guarantee portability. The approach purports therefore that waveform IP in the FPGA is developed separate to that of the platform – so enhancing portability. The management of the waveforms elements internal to the FPGA are to be an OCP thread and the waveform inter-component interfaces should be defined in terms of OMG IDL based SCA ports [7]. These are then mapped into OCP ports. CP289 suggested 3 possible useful profiles for SDR waveforms:

(i) Block Dataflow – principally advanced signaling and data.
(ii) Worker Control Interface – used for command and control.
(iii) Streaming – used for constant time series streams or sensor data as examples.

5.6. CORBA on the FPGA

At least two vendors have developed technologies that allow the FPGAs to run CORBA ORB like entities in the FPGA[6] and allow waveform components to be exported from the FPGA with OMG IDL interfaces for management and control purposes. The efficiency of these technologies with respect to data passing and to help achieve waveform portability is currently still a work in progress. The work has great promise. It will be necessary to see some comparative test data via some industry studies or reports to be able to judge the relative merits of each of these product ranges at present.

6. SCA AND WAVEFORM MODELLING FOR HDR

It is unclear how the designer will ensure that HDR waveform elements in an FPGA conform to SCA compliance given that there is little guidance on what to do in the FPGA. Ultimately, passing SCA certification depends on satisfying the tests of JTels Waveform Test Tools (WTT) execution against the proposed solution. But can WTT test a highly FPGA centric solution and give an accurate and meaningful estimate of compliance and porting risk for the waveform implemented? The answer to this question is not clear, and warrants more work on the JTels WTT to garner metrics data like that in alluded to in [1].

7. DEGRESS OF REUSE

Waveform reuse can be viewed as a continuous scale, and will evolve in time for the FPGA platforms. OCP offers a first step in this approach. It helps resolve the problem of generically defining a portion of the FPGA logic that can simulate the SCA APIs in the FPGA in a similar fashion to those on the GPP. A waveform independent, platform specific hardware abstraction layer needs to be defined as a standard also, much like POSIX is an OS/OE standard is a GPP platform specific standard.

In order to achieve the high data rates and maximize waveform flexibility, very large and often times expensive FPGAs must be utilized resulting in expensive to build and deploy platforms. Fortunately, over time as the SCA community gains exposure to FPGA centric waveforms, the platform requirements can be "right sized" and economies of scale will continue to reduce the FPGA device cost [2][3][4][5]. Given the lack of a standard FPGA device, GPP to FPGA interconnect and FPGA logical function mapping, effort estimation and therefore cost estimation is difficult and variable, but there are new metrics for right sizing the FPGAs and calculating OCP overhead per waveform component[1]. Additionally, most of the existing waveforms are written in software and therefore are not reusable for a FPGA centric, high data-rate application. OCP based design requires care because waveform components must be implemented to a specific OCP profile. The waveform then can only be ported to a radio platform that can support that particular OCP profile(s). This OCP interface then is what can be made the universally addressable interface (UAI) for the FPGA platform as it is defined initially in IDL and starts life in the same form as the other traditional SCA element interfaces, but is mapped to the target platform as an OCP element. Thus reuse is available at the OCP level as an implementation and portability is possible because of a common (albeit started in IDL) interface.

8. RADIO SERVICES

Practitioners in the field of JTRS SCA have touched upon the importance of the Radio Set View versus Waveform Views [11] with respect to Radio Services. Radio Services are really 'open unencumbered APIs' used to interact with the radio and should be the goal of SCA, and measured in SCA certification. These are practical GPP side APIs connected to the waveform in the radio. The acceptance of a *Radio Set view* for API definition is essential to waveform portability since a radio can and will support many waveforms now and into the future. Hindering this is the lack of any accepted definition of what a Radio Service or Device is or what Radio Services and Devices are required for an SDR with which humans will interact. As a result, HDR programs which are going to guarantee waveform portability must provide a new characterization of a Radio Set View to support its class of HDR waveforms.

SCA compliance is usually proven using the industry tool-suites that are custom built for that purpose in that US Governments tool repository. These of course are the Jtel's JTAP and WTT or WFT tools. These provide measures of whether or not your platform and waveform hosted on it pass SCA requirements or not. However this certification is only for the GPP side of SCA. Unfortunately as humans have to interact with the radio at all times – there is almost always an application that falls outside of the SCA testing domain – that of radio services which is necessary but not tested. These can be classically modeled and incorporated as an SCA application or a service. However, they do not get tested by any certification process. While many may argue this is not necessary – HDR platforms that will see a variety of different data-rate waveforms into the future will need some standardization of the waveforms interface with Radio Services. This means that as design of HDR platforms mature we will want/need a common interface on the FPGA platforms to interface our Radio Services stack to and hence achieve a Radio Set View to interface to the waveform view [11].

9. CONTAINERS, AND THE UNIVERSALLY ADDRESSABLE INTERFACE (UAI)

Containers typically have been a way to manage and control complexity when it comes to collections of objects. Containers manage object lifecycles, associated resource usage and the complex interrelationships that can be formed among collections of objects. These collections can provide a service or use other services to provide a service to form a waveform on an SDR[10].

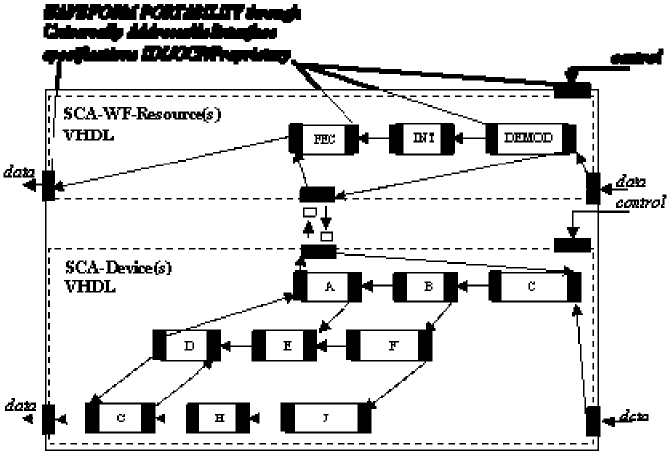
WAVEFORM PORTABILITY through
Universally Addressable Interface
specifications IDL/OCP/Proprietary
control
SCA-WF-Resource(s)
VHDL
FEC
INI
DEMOD
data
data
control
SCA-Device(s)
VHDL
A
B
C
D
E
F
data
G
H
J
data

Figure 5: *Achieving Waveform portability with the notion of universally addressable interfaces to define the portability boundary. The definition of these interfaces could be IDL, or OCP.*

The embedded system development cycle where both new hardware and COTS software to be hosted on it are being undertaken from the beginning means that hardware and software teams have to march in sync with sync-points in the project to perform checks. The complexity of the software must not necessarily reflect implementation details of the underlying hardware that is the modem manufacturers IP. As a result of this unifying force plus the desire for control across the modem from a series of single points of entry (the SCA foci) lead to the inevitable use of software 'containers'. This aspect or pattern was first introduced in the Java community during the J2EE enterprise software period of the 1990's. SDR today however is not a server centric architecture like J2EE, but can benefit from lightweight container model to manage complexity of the many resources of a waveform application. However what SCA is still missing is the notion of a lifecycle container that can manage larger assemblies of elements – not just lifecycle which is what the core SCA APIs provides. In the arena of HDR platforms there needs to be more clarity and greater control over how to isolate elements of complex aggregate devices to get finer grained control and provide that control to a user of the SCA compliant radio through Radio Services HCI. From the waveforms view, SCA architectural aspects also lead us to consider an SCA aggregate resource just like we have an aggregate device. An aggregate resource has value in the component world when we compose waveforms as assemblies of a multitude of resources and devices connected together. Today waveforms are called SCA applications. In the age of net-centric applications higher level application assemblies can be synthesized on an SCA radio by creating applications upon applications to build a net-centric node, here the container paradigm controls complexity on the SDR.

10. JTEL – JTAP, WTT AND CERTIFICATION IN THE CONTEXT OF HDR-RF

SCA portability and reuse metrics for HDR waveforms warrant some review as discussed earlier given that the waveform is almost entirely VHDL. It is unclear if JTAP and WTT certification is meaningful as a measure of SCA compliance and if it is – what exactly does that certification mean as it does not test FPGA reuse metrics. The JPO should provide guidance on metrics and introduce them into the SCA if indeed OCP emerges as an FPGA standard for JTRS. This would mean a new JTAP and WTT for HDR classes of platforms and waveforms which would then realistically measure and report SCA conformance, reuse and portability metrics.

10. CONCLUSIONS

It is possible using OCP to synthesize an HDR platform and develop portable, reusable classes of waveforms. There remain under-specified areas in the SCA that *must* deal with the FPGA dominant platform compliance needs if JTRS SCA is to succeed in the HDR domain. There also needs to be greater SCA specification of Radio Services and how they interact with an HDR waveform. There is an acute need for a definitive study that publishes a comparison and characterization of the performance and feature comparisons for porting and reuse among OCP, CP289, and CORBA-hybrids for the FPGA. The CP289 approach appears to create greater overhead than a pure OCP approach. Finally, concerning the applicability of SCA to HDR platforms and waveforms, here is one humble point of view. If we consider the platform – the abundance of FPGAs means that one cannot apply the classical vanilla SCA with its pushPacket() data passing approach. OCP offers one approach to augmenting the SCA standardization process and thus define waveform portability interfaces in the FPGA usefully – our proposed UAI. If we consider the waveform – it is not much different to the platform in the case of HDR. It is also still defined and implemented in VHDL or equivalent predominantly. The same concept of OCP profiles and IDL too is useful and using OCP to separate platform from waveform. The standardization of Radio Services servicing a waveform also needs to occur with specification APIs. The only unresolved question that remains is if JTAP and WTT are indeed sufficient as a tool to measure compliance of these new HDR platforms and waveforms from the perspective of waveform portability? There needs to be a way to effectively measure and report on the complexity metrics of porting an HDR waveform from one such FPGA platform to another that uses OCP profiles.

10. ACKNOWLEDGEMENTS

The authors would like to acknowledge Comtech EFData and DataSoft for the kind support and permission to discuss the HDR project. In addition we acknowledge the valuable contributions of all team members in the Comtech-DataSoft team. Finally, last but not least, we would like to thank the US Air Force for their continued support.

10. REFERENCES

[1] K. Skey, J. Bradley, and K. Wagner – "A Reuse approach for FPGA-based SDR Waveforms", *In Proc. SDR Forum. MITRE Corporation.*

[2] D. Boschen, M. Ubnoske, and P. Winkler – "Efforts to Implement an SCA Compliant Wideband MILSATCOM Waveform", *MITRE Corporation.*

[3] C. M. Scardelletti et al. 2007. – "Software Defined Radio Architecture for NASA's Space Communiactions". *In High Frequency Electronics.*

[4] J. Neel, C. Aguayo, and J. Reed. – 2006, "Automated waveform partitioning and optimization for SCA waveforms", *In Proc SDR Forum Tech Conf. 2006.*

[5] M. Hermeling. – 2006, "Component-based support for FPGA and DSP",*In Proc SDR Forum Tech Conf. 2006.*

[6] Multiple Authors: 2007 – "Method and apparatus for accelerating generic inter-orb protocol for a corba orb", US patent filing: *USPTO Application #: 20070130570.*

[7] J. Noseworthy and J. Kulp, "Code Generation for SCA Components Running on FPGAs," *SDRForum Conference paper.*

[8] J. Hogg, and F. Bordeleau, "SCA Deployment Management: Bridging the Gap in SCA Development". *Whitepaper from www.zeligsoft.com.*

[9] L. Land, "SCA Framework Puts a CORBA Face on FPGAs", *COTS Journal* October 2004.

[10] F. Pilhofer, 2004, "Supporting SCA Applications in a Lightweight CCM Environment", *Mercury Computer Systems Presentation*, 2004.

[11] E. Christensen, and David Dohse, 2004, "Software Defined Radio Services and Device APIs", *in Proc. SDR Forum 2004 Technical Conference.*

[12] PIM and PSM Software Radio Components, Object Management Group.

[13] JTRS JPO – "Application Program Interface Supplement to the Software Communications Architecture Specification", JTRS-5000API V3.0, August 27, 2004.

[14] JTRS JPO – "The Modem Hardware Abstraction Layer (MHAL) Application Program Interface" – May 2007, Version 2.11.1.

[15] JTRS JPO – "The Software Communications Architecture (SCA) Specifications", - *Version 3.0, The Specialized Hardware Supplement.*

[16] JTRS JPO – The Software Communications Architecture (SCA) Specifications, - May 2006, *Version 2.2.2.*

[17] The OCP International Partnership – *The Open Core Protocol (OCP) Specification, Rel 2.2* – www.ocpip.org.