Software Radio Technology Overview And Recent Progress
Digital Communications Conference

Joseph Mitola III
The MITRE Corporation, McLean, VA 22102
(jmitola@mitre.org)

Abstract

This paper summarizes software radio technology emphasizing recent progress, including the first software radio workshop of the European Community and progress of the MMITS (open architecture software radio) Forum.. The software radio is an emerging technology for rapidly building flexible, modular, multiband multimode radio systems. It allows one to create radio infrastructure that can be programmed for new standards and dynamically updated with new software personalities. These personalities include air interfaces that may be downloaded to software radios "over the air", reducing the need to purchase new hardware for new services. The technology has been proven in the field, but there are technical, economic and institutional challenges remaining before the benefits of this technology are fully available at low cost. This paper highlights key technical challenges and opportunities.
Keywords: Software Radio. Digital RF., DSP., Architecture

1.0 Introduction

With a *true* ***software*** *radio*, the user can upload new air interfaces and protocol personalities as software radio "applets". This flexibility is a key reason for the rising significance of software radio technology: the software radio's multiband multimode waveforms are software-defined and this allows users and service providers (military, civil and commercial alike) next-generation flexibility in the design and evolution of wireless networks, infrastructure and services. Although the technology has been demonstrated in military radio research programs such as the Defense Advanced Research Projects Agency's (DARPA's) SPEAKeasy program, most of the potential has not yet been realized in the commercial sector. But since the technology's costs are declining with improvements in Digital Signal Processing (DSP) technology, it bears watching for applications in amateur radio. In fact, an early progenitor of the software radio is Standard Marine AB's HF multimode radio[1]. Although this product had only a 28 kilo sample per second Analog to Digital Converter (ADC), it implemented half a dozen HF radio modes in software. With contemporary technology, the digitally accessible bandwidths are now about 12 MHz with an 80 dB full-band dynamic range or l MHz with a 90 dB dynamic range. Such wideband IF access provides new opportunities for HF, line of sight and satellite amateur radio communications products including new air interfaces.

1.1 Disclaimer

The author is employed by The MITRE Corporation, a not-for-profit company that operates Federally Funded Research and Development Centers (FFRDCs) for the US Department of Defense (DOD). At present, he is on loan to the DoD as director of modeling and simulation of telecommunications systems. Remarks presented here are his alone and are not necessarily the views or policy of The MITRE Corporation or the US Government.

1.2 Perspective

Software radio technology has its roots in the Electronic Warfare (EW) programs of the 1980's which accessed the entire electromagnetic spectrum to monitor threats and to inject jamming signals. Thus, the military radio and EW technology is available to build wideband antennas; wideband radio frequency (RF) modules; high sampling rate Analog to Digital Converters (ADCs) and DACs; and the high performance digital signal processors needed for software radios - but at what cost? Commercial applications, especially amateur radio applications, and large scale civilian radio markets are very cost sensitive. So the focus of this paper is on technologies for improved cost/benefit and broader applications of software radio technology.

The software radio was identified as a key enabling technology of the "future-proof' infrastructure needed by wireless service providers in BellSouth's December 1995 Request for Information for the Software-Defined Radio [2]. The US Federal Aviation Administration (FAA) also requested industrial participation in inserting this technology into future avionics and ground based radio infrastructure [3]. A cross-section of over 100 government and commercial players created the Modular Multifunction Information Transfer System (MMITS) Forum in March of 1996 to promulgate the benefits of open architecture plug-and-play software radio and digital communications to an expanding marketplace [4]. MMITS has included Alcatel (France), Nokia and Ericsson (Sweden), Orange (UK), Samsung (Korea), and Raphael (Israel), and Computer Aided Design (CAD) software vendors, among others, including numerous US commercial and military radio product suppliers. In addition, the first European Workshop on Software Radios, held in Brussels on May 29 brought together over 160 European telecommunications systems professionals to deliberate the European approach to this technology. With so many players across the global landscape engaged in an emerging technology it is essential to first agree on a few definitions.

Table 1 Software Radio Functional Definition

General Properties	Universal air interfaces (source coding, channel coding, error control and protocols), regardless of **multi-technology (FDMA,** TDMA, CDMA or hybrids), **multi-band and multi-standard environments.**
Services*	Seamless internetworking of **AM, FM,** cellular **(analog,** TDMA, CDMA), PCS, mobile data and paging; **seamless bridging of multiple bands and modes**
Standards*	**HF ALE, VHF/UHF** voice/data; **privacy;** GSM, PCS and **Frequency Hop (FH).**
Technical Flexibility*	**Flexible RF, Channel,** Time Slot, Power, Bit rate, Equalization, **Channel Coding** and Error Correction.
Supports Advances	Adaptive networks, **transparent bridging,** innovative signaling and improved quality. Over the air downloading of radio personalities
Growth Path	Velcro radio (multiple hardware personalities) -> **DSP-enabled radio** -> **Multi-personality radio -> Variable personality software radio.**

Lists are illustrative, not exhaustive **[SPEAKeasy Demonstrates the Capabilities Shown In Bold]**

2. Software Radio Definitions

2.1 Functional Definition

The service providers - military, civilian and commercial - are most interested in the functional capabilities and cost of the software radio and less interested in the technical details. The functional definition of Table 1 captures the significant functional dimensions of the software radio. SPEAKeasy, the first widely published military software radio, demonstrated the functions shown in bold in the table [5, 6]. The BellSouth software-defined radio growth path envisions a shift from multiple chip sets, one for each air interface (the "velcro" radio) to the Digital Signal Processing (DSP) - enabled radio. DSP includes not just DSP chips such as Texas Instruments' TMS320 and the ADSP SHARC, but also Field Programmable Gate Arrays (FPGAs) and general purpose processors such as Intel's Pentium/ MMX. As technology evolves software will also control RF analog processing through programmable Micro Electra-Mechanical Devices (MEMS).

2.2 Architecture Definition

The architecture of the software radio is described in detail elsewhere [7, 8]. Fundamental to the definition of the software radio is the use of wideband ADCs, DACs and high performance DSPs to define waveforms in software at Intermediate Frequencies (IF), improving on our recent ability to define waveforms at baseband. Soon technology will emerge for defining waveforms digitally at Radio Frequencies (RF). The other fundamental concept is to host all the software (DSP and otherwise) on general purpose programmable processors. All aspects of the air interface including the channel waveforms would then be defined in software and implemented in real time through isochronous software and/or firmware (versus dedicated digital hardware as in "digital" radios). These fundamentals cannot be achieved without wideband antennas[1], wideband RF amplifiers and RF distribution and (for now at least) wideband IF ADCs and DACs.

The software radio functional architecture block diagram of Figure 1 shows how the wideband ADC provides simultaneous access to a large population of channels, e.g. for subscribers in Personal Communications System (PCS) base stations. Software-based isochronous signal streams support the air interface. The environment characterization near-real-time stream provides parallel access to all the channels, e.g. for adaptive channel assignment algorithms. software radios also have a role in R&D where advanced services like joint source-channel coded variable data rate air interfaces are defined in software and targeted for software, firmware and/or hardware implementations via C and VHDL.

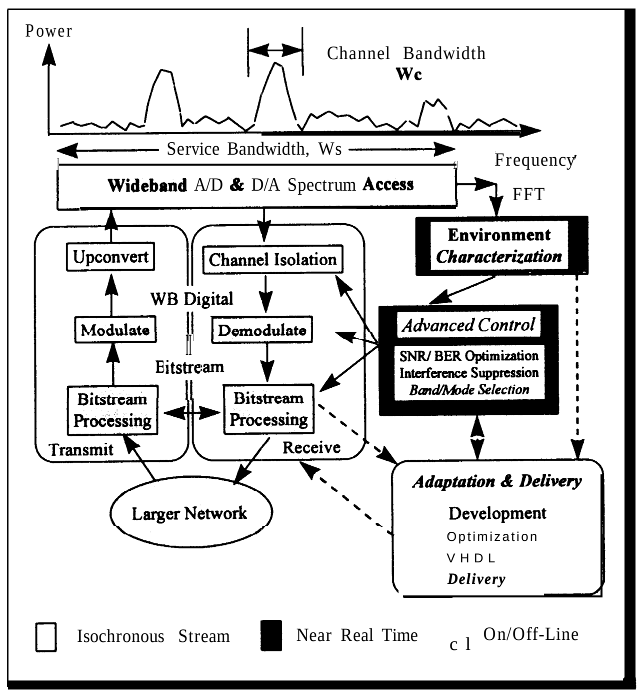


Figure 1 Software Radio Architecture

2.3 The Phase Space Definition

Digital radios and software radios may also be defined in the "phase space" of Figure 2. [Phase

---

[1] The SPEAKeasy I military software radio, for example, used three antenna bands between 2 MHz and 2 GHz, approximately 2-30, 30-300 and **300-2000** MHz.

spaces are used by physicists to represent changes in the states of matter as a function of key external parameters like temperature and pressure; we borrow the terminology to reflect the states of radio devices with respect to the key parameters of maximum frequency accessed digitally and degree of programmability.] The "ideal" software radio accesses RF directly via super-wideband ADC/DACs and accomplishes all processing using general purpose computer chips (see the circle marked X in the upper right corner of the figure).

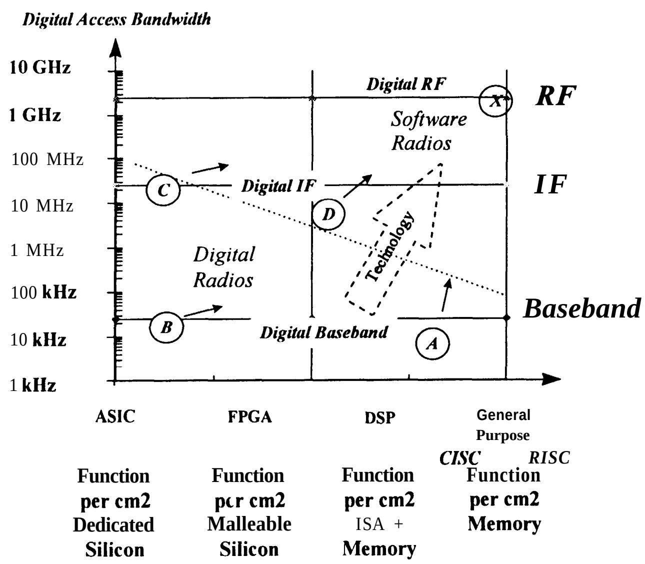


Figure 2 Software Radio Phase Space

Such software radios cannot be built economically yet. The pure digital radio, on the other hand, accomplishes most functions in Applications-Specific Integrated Circuits (ASICs) (circle C in the figure). The software radio maximizes flexibility and therefore truly future-proofs the infrastructure against new standards. The digital radio, conversely, maximizes hardware efficiency and therefore minimizes the size, weight and battery drain critical for handset applications.

Neither approach is a panacea: the key question is the degree of programmability required for the intended market. Contemporary radio designs therefore vary across the dotted line in the phase space that represents the technology frontier, comprising a mix of ASIC, FPGA, DSP and general purpose processors using ADCs and DACs at baseband or IF. The circle marked A in the figure shows the design point for the commercially successful Standard Marine HF amateur radio product [1], for example. Advancing microelectronics technology moves all implementations upward and to the right over time.

Handsets favor ASICs and/or FPGAs with chip level integration. Emerging cellular base stations, on the other hand, favor larger granularity hardware modules. Some use block up and down conversion with 12.5 to 25 MHz bandwidth ADCs/DACs (30 to 70 M samples per second) to accommodate 1 00-plus subscribers. This software radio approach reduces the hardware complexity of a cell site from several racks of discrete single-channel radios to one or two shelves of open architecture PCI, VME or other low cost DSP hardware. Military radio, avionics and amateur radio markets fall between handsets and base stations. Typically two dozen channel waveforms ("modes") are needed from HF to SHF for military radio interoperability, but no more than a half dozen modes are in use at any one time per platform (e.g. tank, aircraft, etc.). Amateur radio applications might employ two or three simultaneous modes out of a couple of dozen available. Simultaneous voice, automatic digital QSL, packet satellite and monitoring favorite channels could all be supported in an amateur software radio.

2.4 The Vector Space Definition

Two years ago, nobody had a "software radio" and now almost every radio vendor on the planet claims to have one. This confuses buyers, program managers and investors. We may clarify things somewhat by defining the degree to which a product meets four key technical and economic criteria that bridge the gap from technologies to applications. The criteria are:

1. The number of air interface channels simultaneously supported (N),
2. Programmable Digital Access (PDA),
3. Hardware Modularity **(HM),** and
4. Software Flexibility and Affordability (**SFA**)

Table 2 defines these criteria. It is useful to group (N), the number of air interface channels, into four types: single channel; dual channel; multiple channel (i.e. less than 6); and full access (i.e. the full number of subscribers in an allocated RF band) to define four levels of N. Multiple channel nodes are typical of military, civil and amateur radio building blocks while the full band modes are typical of cell site infrastructure.

Table 2 Definitions of Four Key Software Radio Dimensions (N, PDA, HM, SFA)

N: Number of Channels:

n: 1,2 or a few (<6) simultaneous air interface channels;

NN: The full number of subscribers in the RF band.

PDA: Programmable Digital Access: None (0), Baseband (1), IF (2), RF (3);

Baseband bandwidth is defined by single subscriber service (e.g. voice, data modem, video)

IF is defined as that bandwidth which simultaneously supports all NN subscribers in the allocated

RF service band (e.g. 12.5 MHz analog FDMA)

HM: Hardware Modularity:

None (0), Receiver/Exciter/INFOSEC/Network Modules (1), COTS DSP Modules (2), Second Level Modules (ADCs, FPGAs, Receiver Chips, etc.) (3);

SFA: Software Flexibility and Affordability:

No Air-interface-defining software (0), Single-supplier software (1), Multiple supplier but single host platform (2), Multiple supplier multi-platform software (3).

The level of Programmable Digital Access **(PDA)** is the level of the conversion to digital at which the radio is functionally programmable in the software radio phase space. The types are: none (totally analog or fixed function digital radio); baseband programmability; IF programmability; and RF programmability. A privacy mode that hops over 2 MHz but that cannot be programmed for any other waveform may have a wideband IF, but it would not have a ***programmable digital access at*** *IF*, which is the criteria specified in this definition.

The Hardware Modularity **(HM)** criterion recognizes the differences in upgrade path between relatively coarse grain (possibly programmable) modules such as receivers and exciters; versus other types of coarse grain modules (e.g. COTS ADC and DSP boards); versus finer grain modules such as FPGA, ADC and DSP chips. The hardware modularity value is not prejudicial: The key is to explicitly decide what type of modularity is called for by the life cycle of the application and to match that type in the implementation. Amateur radio applications favor PC board level modules.

The Software Modularity (SFA) dimension characterizes the service provider's ability to buy plug and play software modules based on the vitality of the marketplace. Software that runs on just one platform and is available from only the original manufacturer tends to box the user into single-source (sometimes very expensive) maintenance. If the functionality of the unit will not change over its life cycle, then this may be a perfectly acceptable path. This would be a rare occurrence in today's fast changing marketplace. Software that runs on many platforms (e.g. JAVA) and is available from multiple vendors generally gives the buyer a better software product with more flexibility and at a lower cost over the life cycle.

We may think of these criteria as a feature space, yielding a characteristic vector for a given radio product:

Radio X: (Nx, DAx, HMx, SFAx)

Each vector element varies between 0 and 3 per the assigned type, so the vectors range from (0000), the unprogrammable analog radio to (3333), the totally programmable software radio. Since there are four levels of capability for each of these four features, there are a total of 256 points in this feature space. These 256 points cluster into four groups that may be called capability levels.

2.5 Software Radio Capability Levels

The clusters of software radio feature vectors consist of the four aggregate "software radio capability levels" shown in Table 3. Radios at level zero have fixed functionality and cannot be programmed by the user. Level one radios have programmable digital basebands while level two radios have programmable digital IF's. Level three, programmable digital RF, is the ideal software radio which with today's technology is unaffordable. Level zero is not necessarily bad or low-technology. Level zero includes digital baseband dedicated function chip sets that can have very high levels of' device technology (e.g. GSM handsets). For example, simple dual mode handsets need no air interface programmability, so non-programmable ASIC chips are generally most cost effective (e.g. for GSM/ CDMA handsets).

Table 3 Definition of Significant Software Radio Capability Levels

Software Radio Level	Characteristics [Examples with level rating (N, PDA, HM, SFA)]
Zero **Fixed Functionality**	**Analog Radios** [Walkie Talkie (1 ,0,0,0), FM FDM (NN,0,0,0)] **Digital Readout** [Pager (1, I ,0,0) Direct Conversion Handset (1, I ,0,0)]
One **Programmable Digital (Baseband) Radios**	**"Narrowband" reprogrammable, modular (*,1 ,>0,>0)** [Contemporary programmable digital radios Closed architecture (1 -n, 1 ,0, 1), modular (1 -n, 1,1,1) "Open Architecture " (I -n, 1,2,1-2), Goal (1 -n, 1,1-3,3)]
Two **Programmable Digital-IF Radios**	**"Narrowband" programmable , some wideband hardware (*,2,>0,>0)** [SPEAKeasy Class (1 -n,2,2/3,1 or 2?) A Few New Cell Site Products (NN,2,1/2,1)
Three **The Software Radio**	**Programmable Digital RF ["Ideal" Software Radio (NN,3,1-3,3)]** **-** Not affordable yet, but useful as a migration challenge

The aggregate software radio level does indicate the degree of air-interface programmability by the user. Contemporary programmable digital radios meet level one criteria, with narrowband programmability through baseband digital signal processing. From software radio feature vectors, it is easy to see the differences among level one implementations with closed architectures and no real hardware modularity or growth path (1-n, 1 ,0, 1), modular hardware (l-n, 1 ,1 , 1), some degree of modular hardware and software for nominally "open architecture" (1 -n, 1,2,1-2); and the highly modular widely supported open architecture goal (1-n,1,1-3,3).

Contemporary programmable digital radios seek to achieve level two, programmable digital IF, in the near to mid- term. The US SPEAKeasy program, for example, reaches toward level two with medium bandwidth ADCs and DACs and nearly 1 GFLOP of DSP. The need for simultaneous channels and flexibility in the field provided by software radio technology must be balanced against other competing demands of the market segment as illustrated in Figure 3. These market segment drivers create a set of engineering, design and regulatory challenges and opportunities.

Market Segment	Simultaneous Channels/Modes	Architecture Drivers	Standards
Handset	1-2+GPS	Mfg Volume "Velcro"	Chip Level Interfaces
Manpack/ Avionics	4-20	Size, Weight, Power	PC1 PCMCIA
Law Enforcement	20-1 oo+	Cost	?
PBX, WLL	20-1oo+	Call Quality	?
Base Station/ & Mobile Bases	>100	Future-Proof DSP Leverage	VME-like +Wideband Bus

Figure 3 Market Segment Drivers

3 Key Challenges

In order to move as an industry from level one to level two of software radio flexibility and affordability, we need higher quality RF access and better partitioning of the systems and software for modular plug-and-play services and support as highlighted in Figure 4. None of these challenges has an easy, inexpensive and near-term solution, but all are being addressed by technology investments currently under way.

- **High Quality RF Access**
  - Increased Useful Wideband Dynamic Range with Improved Noise Immunity
  - Lower Cost of Broad RF Access
  - Reduced Cosite Interference
- **Partitioning for Plug-and-Play and Reuse**
  - MMITS "API" Approach
  - Real Time CORBA
  - Z. 100 Communications Language

Figure 4 Key Software Radio Challenges

3.1 High Quality, Low Cost RF Access

Software radios depend on high quality low cost access to broad ranges of the RF spectrum. Although the ADC plays a key role, the useful dynamic range is defined by two-tone spurious-free dynamic range (SFDR) established by the multiplicative effects of RF conversion, ADC and subsequent digital filtering. As noise and interference from the environment aliases into the ADC passband, it reduces the sensitivity of the overall system. The key challenge in the short term is to reduce such noise and interference through technical advances in antennas, RF analog filters, ADCs and digital filters.

The costs of RF access are driven by mechanical RF structures (antennas, waveguide, coax and other "plumbing") which are large and expensive because of the relatively large number of discrete parts and the high labor content of assembly and installation. The recurring costs of software radio nodes may include upwards of 60% for wideband antennas, RF distribution, RF conversion and IF processing but as little as 10% for Commercial Off The Shelf (COTS) DSP hardware. As digital hardware moves towards the antenna, it brings the advantages of rapidly advancing microelectronics technology including reduced production costs. There are standard RF packages, waveguide and connectors but many RF assemblies are virtually hand crafted on the “production” line. The matching of voltage standing wave ratios, trimming of capacitance, and other touch labor increases costs. Digital technology, on the other hand, reduces or eliminates most such manufacturing steps. The key challenge for the long term is to move more from analog RF to digital RF.

Finally, cosite interference is mostly our own fault. In the early days of radio, the size and expense of the radio equipment made Frequency Domain Duplexing (FDD) economically infeasible for most applications. As a result, many bands have a legacy of Time Domain Duplexed (TDD) air interfaces in the HF, VHF/ UHF, satcorn and other bands. Any software radio that attempts to service all the users in a TDD band with a single (low cost) RF/IF/ADC channel has the problem of “screaming in its own ears” as multiple TDD signals are transmitted and received at the same time. There are a few technical approaches to mitigate such interference, but mitigation beyond 20 to 30 dB requires a research breakthrough. One could dramatically reduce cosite interference in the far term through a spectrum use paradigm shift from TDD to FDD.

3.2 Plug-and-Play and Reuse

Plug-and-pIay radio would bring the benefits of the open architecture desktop to most amateur radio applications. Reuse can be accomplished on at least two levels: software reuse and waveform reuse. Both plug-and-play and reuse depend on a workable partitioning of the software into modular functions with clearly defined and broadly accepted interfaces. The MMITS forum is pursuing a partitioning based on an Applications Programming Interface (API). The Object Management Group (OMG) recently requested proposals for real-time multimedia support for its Common Object Request Broker Architecture (CORBA)[9]. Multimedia support should add isochronous channels to a growing repertoire of CORBA capabilities relevant to plug-and-play and reuse in software radios. CORBA has little domain-specific (radio voice and data) representation ability. This is both a strength and a weakness. The strength is that CORBA is widely applicable. The weakness is that one must define one’s own radio domain representations. As a result, the representation of radio engineering interfaces among software objects such as state machines must be found elsewhere. The Z.100 Recommendations of the ITU-T [10] describe the Specification and Description Language (SDL) that provides a rich set of expressions of telecommunications behavior including call processing; maintenance and fault treatment; system control; data communications; and some telecommunications services. SDL is compilable with a wide base of European users. The key challenge is the integration of CORBA and SDL in an API such as that contemplated by the MMITS forum for commercial plug-and-play software radios.

4.0 Technology-Enabled Opportunities

These are significant challenges, but many have related enabling technologies to which we can look for continued progress through investments focused as suggested in Table 4.

Table 4 Enabling Technologies

Technology	Representative COTS Performance	Investment Focus
Multiband Multibeam Antennas	Decade (e.g. 0.4 to 3 GHz)	Gain, physical size, alignment, cost
Wideband RF	Octave to decade	Automatic mechanical tuning, MEMS
Wideband ADC	70 MHz x 12 bits	Dynamic Range x Bandwidth
HTSC Filters and Amplifiers	30-40 dB More Out of Band Rejection	Product Integration
High Performance interconnect	140 to 1000 Mbytes per second	User-accessible throughput
Digital Signal Processors	25 MFLOPS to 2000 MIPS	Throughput, power consumption
Real Time Object Software	COTS Radio / Telephony Functions Real Time CORBA for Multimedia ITU-T Z. 100 SDL Products	Quantified Real-Time Performance Object Oriented packages Integration of APIs

4.1 Sensitivity and Dynamic Range

The Commercial Off The Shelf (COTS) multiband multibeam antennas, wideband RF and wideband ADC products of Table 4 may be enhanced for software radio applications through the technology investment focus areas shown. High Temperature Super-Conductive (HTSC) filters now entering the market suppress adjacent channel noise and interference by 30 to 40 dB [11]. Otherwise, this interference would alias into a software radio's wideband programmable IF. Conductus, Inc. (Sunnyvale, CA) and Illinois Superconductor Corporation (Mt. Prospect, IL) are among several companies to offer HTSC filters and/or low noise amplifiers. Ericsson's presentation at the European Conference[12] emphasized the relaxation of test requirements for GSM so that today's ADC and filter technology could be applied immediately for multimode base stations.

In the Proceedings of the GaAs IC Symposium [13], Walden reported ADCs with 6 bits of resolution at 6 GHz, allowing one to sample a 2.5 GHz RF waveform above the Nyquist sampling criterion. The two tone spurious free dynamic range (SFDR) of this ADC may approach 30 dB, but most radio applications require 70 to 90 dB or more dynamic range. The ADC seems inadequate until oversampling is considered. The oversampling gain in dynamic range (DNR) is approximately the ratio of the sampling rate to the Nyquist rate as shown in Table 5, provided sampling clocks have the aperture uncertainty and stability necessary for coherent integration. Vetterli has described projection filters that optimize dynamic range recovery [14]. The preservation of SFDR from GHz sampling rates to narrowband modulated channels has not yet been achieved in practice but is an active research area. Current research emphasizes higher speed and dynamic range ADCs and DACs [15] with demultiplexers to reduce data handling clock speeds.

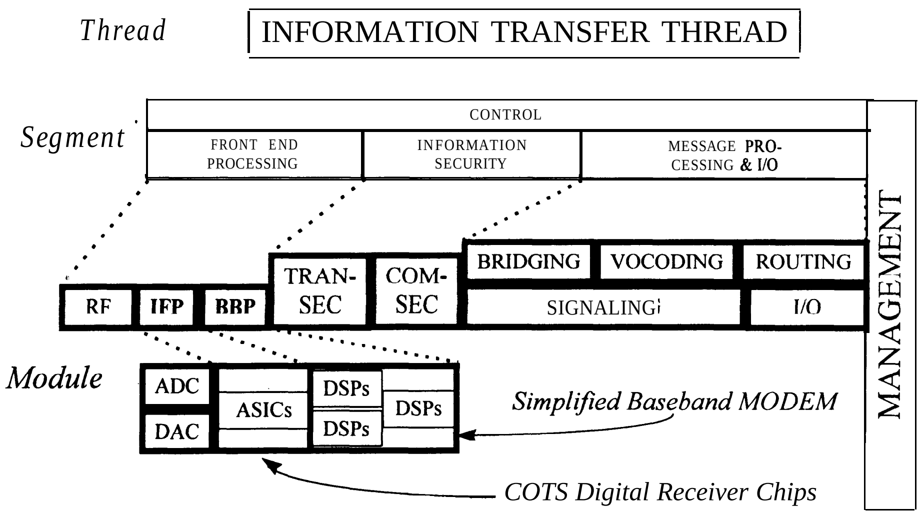
4.2 Digital RF'

Table 5 Dynamic Range Improves With Oversampling

(Over)Sampling Rate MHz	ADC's SFDR	Cell Site Bandwidth MHz	Gain In DNR	Cell DNR	Subscriber Bandwidth kHz	'Subscriber DNR
6000	30	25	21	51	30	80

5.0 Plug-and-Play and Reuse
5.1 The MMITS API Concept

The MMITS Forum has begun to define open architecture standards for plug and play communications embracing digital and software radios, offering the "Information Transfer Thread," a refinement of which is shown in Figure 5.



IFP = IF Processing; BBP = Baseband Processing

Figure 5 Refined Information Transfer Thread Reference Model

The heavy lines in the figure represent Applications Program Interfaces (APIs) mediated by a common bus (e.g. the red or black bus). The dotted lines link successive levels of expansion of the hierarchical decomposition of the Information Transfer (IT) Thread. At the top level, the IT Thread consists of a front-end; an information security (INFOSEC) element; and a back-end for message processing ("inter-networking"), and input/output to user(s) and possibly to wireline interfaces. The functional interfaces at this level provide a reference model for building radio applications via a standard API.

5.2 Real Time Software Objects

Non-recurring engineering often includes greater than 50% software development costs. The software is generally on the critical path with system integration and test as custom software, COTS hardware and operating systems and signal processing libraries are fused into a viable system. Object oriented languages and design have entered the mainstream of real-time applications [16,171, but the impact of Common Object Request Broker Architectures (CORBA and related standards[18]) has not yet reached the front end of the architecture. That is, a user or systems developer cannot yet substitute one RF modem software component for another without relatively labor-intensive customized coding and system tuning.

This is in part due to the lack of standard applications level software to software APIs and in part due to a lack of quantification of requirements for memory, buffer space and processing resources. Orange Communications, Nokia and others at the

First European Conference [12] emphasized the need to develop such API's. It is difficult to tell if one *could* meet throughput, response time and other critical requirements when attempting to reuse code from one's own libraries or from a third party. Quantified software objects specify applications layer interfaces and related memory and processing resource requirements for real-time performance. As such, they reduce the time required and risk of buying software that will not work effectively in one's PC or laptop environment. A reusable object could use CORBA interfaces for plug and play, but this does not guarantee performance. In fact, the CORBA overhead detracts from throughput and response time. So a quantified object characterizes the processing demand that must be supported for real-time performance.

6.0 Conclusions

As digital radios make the transition to software radios, industry is striving for economic efficiency through the adoption of open architecture standards and through technology insertion. Workshops such as the first European Workshop on Software Radios certainly enhance that process through stimulating critical technical and business dialogs. By adopting standard terminology to describe different kinds of software radios, we can enhance our ability to communicate technically and with customers and investors. We also need a accurate communications between buyers and sellers of amateur radio digital hardware and software products. The software radio feature space is offered as a useful characterization of product technology and robustness. As suppliers offer more software radio products, working in a standard API such as the MMITS API and quantified so that procurement risk is low, we will see much broader market acceptance of the software radio.

References

1 Standard Marine AB Product Description, 1991

2 The Software Defined Radio, BellSouth, Dec 1995

3 Broad Agency Announcement, The Federal Aviation Administration, Commerce Business Daily (12 Sept 96)

4 MMITS Forum Charter (Internet www.rl.af.mil/MMITS; April 95)

5 Lackey and Upmal, "SPEAKeasy: The Military Software Radio," IEEE Communications Magazine (IEEE: NY), May 95

6 Fette, B. "SPEAKeasy II Program Overview", Minutes of the MMITS Forum (Reference 3) June and March 96.

7 Mitola, "The Software Radio: Architecture and Prognosis, IEEE National Telesystems Conference 92 (IEEE, NY)

8 Mitola, "The Software Radio Architecture", IEEE Communications Magazine (IEEE, NY; May 95).

9 Object Oriented Technology, IEEE Communications Magazine (IEEE, NY), Feb 97.

10 CCITT Specification and Description Language, Recommendation Z. 100 (ITU, Geneva, Switzerland), 1993.

11 Hogan, "Superconductor products poised for market", *The Industrial Physicist* (Woodbury, NY), March 1997.

12 First European Conference on Software Radios, European Commission, Brussels, May 29 1997

13 Walden, "ADC Integrated Circuits", Proceedings of the IEEE GaAs IC Symposium (IEEE Press, NY) Dec 95.

14 Thao&Vetterli, "Optimal MSE Signal Recovery in Over-sampled A/D Conversion Using Convexity", IC-ASSP-92.

15 Lemnios, Zachary, DARPA ADC Program Review, (DARPA, Arlington, VA), March 96.

16 Ellison, Karen S., Developing Real-Time Embedded Software-In A Market-Driven Company, (Wiley, NY) 96

17 Ellis, J. Objectifying Real-Time Systems, (SIGs Books, NY) 1995.

18 The Common Object Request Broker: Architecture and Specification, OMG 9 1.12.1 (DEC, Maynard MA) 1993