# Cognitive Radio: From Spectrum Sharing to Adaptive Learning and Reconfiguration

Feng Ge, Qinqin Chen, Ying Wang, and Charles W. Bostian
Virginia Polytechnic Institute and State University
Wireless @ Virginia Tech
Center for Wireless Telecommunications, Virginia Tech,
Blacksburg, VA 24061, USA
540-231-2558
{gef, chenq, ywang06, bostian}@vt.edu
Thomas W. Rondeau
Centre for Telecommunications Value-chain Research (CTVR), Trinity College, Dublin, Republic of Ireland
trondeau@mee.tcd.ie
Bin Le
Cognitive Radio Technologies (CRT), LLC
bin.le@crtwireless.com

*Abstract*—This paper[1][2] introduces important cognitive radio developments like spectrum sharing, learning and adaptation algorithms, and the software and hardware architecture to support these functions. A cognitive radio is defined here as a transceiver that is aware of its environment and can combine this awareness with knowledge of its user's priorities, needs, operational procedures, and governing regulatory rules. It adapts to its environment and configures itself in an appropriate fashion. The radio learns through experience and is capable of generating solutions for communications problems unforeseen by its designers.

Our spectrum sharing cognitive radio is built upon GNU Radio and uses the Universal Software Radio Peripheral (USRP) device as our radio front end platform. We use cyclostationary feature analysis to detect low SNR modulated signals because of its ability to distinguish between modulated signals, interference, and noise in low signal to noise ratios. A parallel algorithm running on a Cell Broadband Engine (Cell BE) is used to attack the associated high computational complexity. A new spectrum sensing scheme, incorporating spectrum monitoring, data transmission, and dynamic channel switching, is designed to fully utilize the idle time of the primary user.

Our work is based on the concept of a Cognitive Engine: an intelligent software package that "reads the meters" and "turns the knobs" of any attached software defined radio (SDR) platform. Using an eclectic combination of artificial intelligence techniques including case-based decision theory, multi-objective genetic algorithms, and neural networks, it implements a system of nested cognition loops. Applied to public safety communications, this technology is the basis of a working prototype Public Safety Cognitive Radio that can scan the public safety spectrum (multiple bands and multiple waveforms, all incompatible) and configure itself to interoperate with any public safety waveform that it finds within 0.1 seconds of determining that a signal is present.

TABLE OF CONTENTS



## 1. INTRODUCTION

Spectrum occupancy measurements reported in 2005 by Shared Spectrum Company (SSC) showed that the average spectrum occupancy over all the radio bands between 30 MHz and 3,000 MHz was 5.2% as observed over multiple typical geographical locations [1]. This is in sharp contrast to the overcrowding of wireless communication in the public safety bands, the WiFi band, and most other unlicensed industrial, scientific and medical (ISM) radio bands, etc. To alleviate this disparity, the Federal Communications Commission (FCC) is exploring the possibility of a spectrum sharing mechanism, by which a secondary user can share the spectrum on conditions of non-interference to the primary users.

Cognitive radio [2] is the most promising technology in spectrum sharing. A cognitive radio can change its transmitting or receiving parameters to communicate efficiently while avoiding interference with licensed or unlicensed users. We define a full cognitive radio as *a transceiver that is aware of its environment and can combine this awareness with knowledge of its user's priorities, needs, operational procedures, and governing regulatory rules. It adapts to its environment and configures itself in an optimal fashion. The radio learns through experience and is capable of generating solutions for communications problems unforeseen by its designers*. This full cognitive radio envisions rapid progress in "software

1

[1] 1-4244-1488-1/08/$25.00 ©2008 IEEE.
[2] IEEEAC paper #1668, Revision 2, updated January 14, 2008.

radio" technology, which has been aided by advances in processors, RF technology, and software since 1991 [3]. Therefore, a cognitive radio is the evolution of a SDR into an automatically reconfigurable communications system that responds to network and user demands.

This paper introduces some important cognitive radio developments which cover both spectrum sharing and some full cognitive radio functions. In Section 2, a Spectrum Sensing Cognitive Radio (SSCR) design based on the SDR software platform GNU Radio is introduced. Specifically, we will analyze a signal detection approach based on cyclostationary feature analysis and use the Cell BE to attack the computational challenge. In Section 3, we extend our SSCR closer to a full cognitive radio. We designed the platform-independent Cognitive Engine to "read the meters" and "turn the knobs" of any specified SDR system. This Cognitive Engine follows our radio learning and adapting software core in response to radio environment and user requirement. The learning and adapting software core builds upon case-based decision theory, multi-objective genetic algorithms, radio and user database, and a policy engine. In Section 4, we apply our Cognitive Engine architecture and equip specific radio application functions on the GNU Radio platform to design the prototype Public Safety Cognitive Radio. It can scan the public safety spectrum (multiple bands and multiple waveforms, all incompatible) and configure itself to interoperate with any public safety waveform that it finds within 0.1 seconds of determining that a signal is present.

## 2. SPECTRUM SHARING COGNITIVE RADIO

A Spectrum Sharing Cognitive Radio is defined by IEEE as *a radio frequency transceiver that is designed to intelligently detect whether a particular segment of the radio spectrum is currently in use, and to jump into (and out of, as necessary) the temporarily-unused spectrum very rapidly without interfering with the transmissions of other authorized users* [22]. The most challenging technology for SSCR is the spectrum sensing detector which should have both low SNR sensitivity and high agility for wide band signal detection. In general, there are two ways to achieve both requirements: one is mainly based on expensive and highly sensitive analog radio component for signal detection, such as DARPA's XG program [4]; the other procedure uses digital domain techniques by moving as close to the antenna as possible and relies on the low-price and fast computing processors to solve radio analog components' inefficiency or shortcomings.

We developed our SSCR following the second way, which is essentially the development trend for SDR [5]. A Software (Defined) Radio is defined by the FCC as *a radio that includes a transmitter in which the operating parameters of the transmitter, including the frequency range, modulation type or maximum radiated or conducted output power can be altered by making a change in software without making any hardware change* [23]. Among a number of commercial platforms and free open source ones, we have chosen GNU Radio as our SDR software architecture and the Universal Software Radio Peripheral (USRP) device as our radio analog platform.

*USRP and GNU Radio*

The Universal Software Radio Peripheral (USRP) is an openly designed low-price SDR hardware platform which implements radio front-end functionality and A/D and D/A conversion currently using the Universal Serial Bus (USB2) to connect to the PC that hosts the device. The current USRP device (Figure 1) consists of a motherboard containing up to four high speed 12-bit 64 Msps analog to digital converters (ADC), four high speed 14-bit 128 Msps digital to analog converters (DAC), an Altera FPGA and a programmable Cypress FX2 USB 2.0 controller. The ADCs, DACs and the FPGA together provide support for IF processing. The FPGA on the board provides four digital up converters (DUC) and four digital down converters (DDC) to shift frequencies from the baseband to the required frequency. The FPGA can be reprogrammed to provide additional functionality. RF front ends are attached in the form of daughter cards which can currently cover all the existing radio bands from 0 Hz to 2.4 GHz.

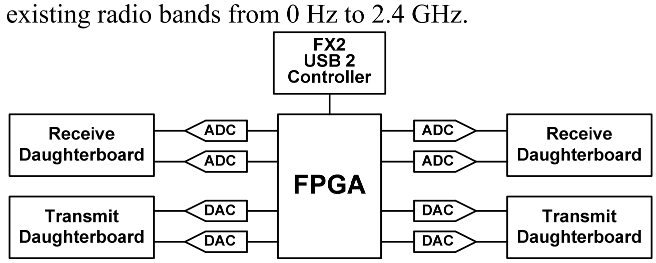



**Figure 1 - Block Diagram of the USRP,** adapted from [6])

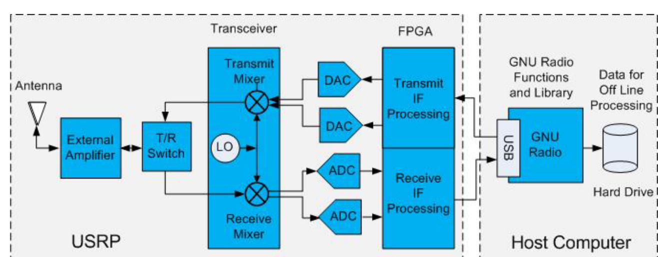



**Figure 2 - A basic SDR system based on GNU Radio and USRP**

GNU Radio is an open source toolkit for building software radios [7]. It was started in early 2000 by Eric Blossom and others and has evolved into a mature software infrastructure used and supported by a large community of developers. It was originally designed to run on General Purpose Processors (GPP), combined with minimal analog radio hardware, and allows software radio development of waveforms, modulations, protocols, signal processing, and other communications functions in the digital domain. The GNU Radio signal processing library includes existing and developing blocks for most signal processing functions, such as waveform modulation and filter creation. It also includes I/O operations like file access. Programming in the GNU Radio platform uses a combination of C++ and Python, a simple, high-level language: the computationally intensive processing blocks are implemented in C++ while the control and coordination of these blocks for applications that sit on top are developed in Python. The USRP is fully supported by the GNU Radio library and a combined system of both is given in Figure 2.

*Spectrum Sharing CR Architecture*

The first challenge for SSCR is to design a signal detector that can quickly search through a wide bandwidth for vacant spectrum in order to establish a new channel. Further, the cognitive radio has to quickly switch to another channel when a primary user appears. We designed a SSCR based on GNU Radio and USRP [8] with the architecture shown in Figure 3.

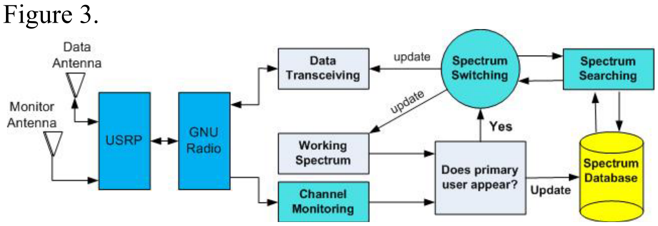



**Figure 3 - Spectrum sharing CR architecture**

The SSCR uses two RF chains: the receiver chain, including the monitor antenna and the channel monitoring, monitors the working spectrum, and the transceiver chain, including the data antenna and the data transceiving, works as the secondary user. In this example, data transceiving is supposed to fully utilize the idle time of a primary channel. The URSP radio analog component control by our SSCR is through GNU Radio functions. The channel monitor will detect any active signals in the specified frequency range and it will also continuously update the spectrum database. When a primary user appears, the data transceiver will switch from its previous working spectrum and search for the available channel from the spectrum database that offers the best QoS to continue the previous communication. This process can guarantee an efficient utilization of the primary signal's idle time.

Currently SSCR's spectrum sharing scheme works in an offline mode from the GNU Radio perspective. It collects radio signals from USRP and digitizes them and then stores the digital samples on the host computer through GNU Radio functions. Next, the samples are analyzed in the computer by the FFT accumulation method (FAM) algorithm. The cyclic frequency result is used to determine spectrum switching following the spectrum scheme shown in Figure 3. We also want to point out that the FAM algorithm for signal detection will be incorporated into the GNU Radio function library on GPPs so that we eliminate memory reading and writing delay and achieve better agility in spectrum sharing.

*Signal Sensing Approach*

A critical requirement for opportunistic spectrum sharing is to sense the spectrum holes quickly and accurately, so that non-interference to privileged users is guaranteed. For individual cognitive radios, cyclostationary feature detection has advantages for spectrum sensing due to its ability to differentiate between modulated signals, interference, and noise in low signal to noise ratios. It is well suited for signal detection and modulation recognition, signal parameter estimation, and the design of communication signals and systems [9].

In this paper, we use the well-known FAM algorithm [10] to estimate the spectral correlation function (SCF) $S$ for a signal process $x(n)$ over $\Delta t$ seconds duration. The discrete $S$ value at each point of the two dimensional cyclic frequency and signal frequency space is shown in equation (1)

$$S_x^{\alpha_i+q\Delta\alpha}(nL,f_i)_{\Delta t} = \sum_{r=0}^{P-1} X_T(rL,f_k)\,X_T^*(rL,f_l)\,g(n-r)e^{-\frac{i2\pi rq}{P}} \qquad (1)$$

where $\alpha_i + q\Delta\alpha$ represents discrete cyclic frequency, $f_i$ is the discrete signal frequency, $g(n)$ is a data tapering window, $L$ is a decimation factor in the frequency domain, $P$ equals ((N - $N'$)/ $L+1$) where $N$ is the total number of samples and $N'$ is the number of samples used to calculate each complex demodulator $X_T(rL,f_k)$. The choice of $N'$ must take into consideration that the time-frequency resolution product $(N/N')$ must satisfy $N/N' \gg 1$ for a statistically reliable measurement [10] and that $N'$ is large enough to obtain the desired frequency resolution. $L$ is usually chosen to be less than or equal to $N'/4$ [11].

We use the crest factor (CF) for signal detection and feature extraction by exploiting cyclic frequency domain profile (CDP) shape [12]. To evaluate any signal's presence, we used the following simple model

$$x(t) = s(t) + \phi(t) \qquad (2)$$

where $x(t)$is the continuous form of the signal processing $x(n)$, $s(t)$ denotes any detected signal, and $\phi(t)$ denotes additive white Gaussian noise (AWGN). A threshold $C_{TH}$ is defined when no signal is present, i.e., when $s(t) = \phi(t)$, for sampled signals:

$$C_{TH} = \max\left(\frac{I'(\alpha)}{\sqrt{\frac{\sum_{\alpha=0}^{N} I'^2(\alpha)}{N}}}\right) \qquad (3)$$

where $I'(\alpha) = \max_f |S_x^{\alpha}(f)|$ for $x(t) = \phi(t)$.

Similarly, we have

$$C_I = \max\left(\frac{I(\alpha)}{\sqrt{\frac{\sum_{\alpha=0}^{N} I^2(\alpha)}{N}}}\right) \qquad (4)$$

where $I(\alpha) = \max_f |S_x^{\alpha}(f)|$ for $x(t) = s(t) + \phi(t)$.

To test signal presence in AWGN, the following binary hypothesis testing is performed.

$$H_0: \qquad x(t) = \phi(t)$$

$$H_1:\ x(t) = s(t) + \phi(t)$$

Based on the threshold $C_{TH}$, we can test the signal's presence as follows:

$$C_I \le C_{TH}: \quad Declare\ H_0$$
$$C_I > C_{TH}: \quad Declare\ H_1$$

The presence of active signal declares $H_1$ and its spectrum frequency is compared to the current secondary working spectrum. If the comparison result indicates the primary signal's presence, spectrum switching will be triggered and the secondary user spectrum is switched to another vacant spectrum. The active signal's spectrum will be always used to update the spectrum database.

*Speedup by Cell BE*

Cyclostationary feature detection has advantages for spectrum sensing. However, a key issue with cyclostationary signal analysis is the high computational complexity arising from the large number of required complex convolution operations. In addition, the computation requirement increases significantly in proportion to the bandwidth to be covered. These factors limit receiver agility and sensitivity.

We have studied the well-known FAM algorithm for cyclostationary feature analysis and have shown that the computation required to cover a single IEEE 802.11g WiFi channel bandwidth at a fine resolution is too high for current GPPs [8]. Future broadband technology will use significantly larger bandwidths, which inevitably demands new computation methods and architectures. We have designed a parallel FAM algorithm on a Cell BE powered PlayStation 3 with six usable Synergistic Processing Elements (SPEs) and one Power Processor Element (PPE). We use two levels of parallelism: the task level, supported by SPEs and one PPE, and the data level parallelism supported by SIMD and Vector Multimedia Extension (VMX) instructions. To fully utilize the Cell BE's capacity, several available speeding techniques are also used such as Direct Memory Access (DMA), loop unrolling, pre-computed sinusoid and cosine arrays for FFT, double-buffering, etc. Further, GNU Radio is being ported into a PlayStation 3, after which we plan to run our SSCR fully in the PlayStation 3.

## 3. THE COGNITIVE ENGINE

In this part, we extend our SSCR closer to a full cognitive radio and introduce the Cognitive Engine [13] which is designed to be independent from any specific SDR hardware and software architecture. It is an open architecture for developing and applying cognitive radio algorithms and deploying cognitive radio functionality. Specifically, the current Cognitive Engine, as shown in Figure 4, contains seven components: the cognitive controller, the sensor, the user interface, the optimizer, the policy verifier, the radio platform, and the decision maker attached with the knowledge base. The outermost layer contains several possible functions that can be deployed in the cognitive engine. Particularly, our SSCR's spectrum switching resides in the decision maker, the spectrum database resides in the knowledge base, and the spectrum searching resides in the optimizer.

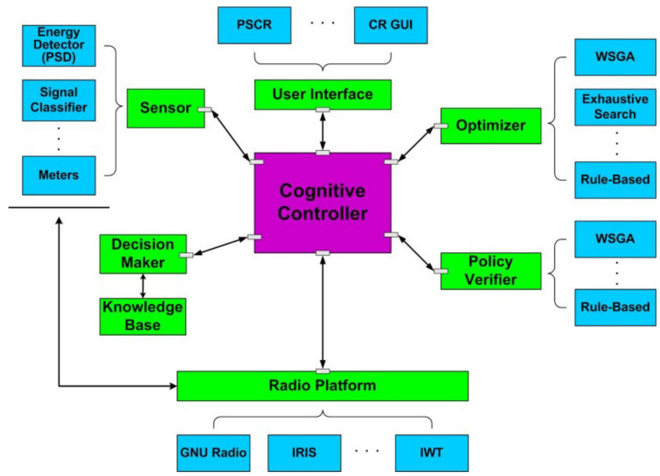



**Figure 4 - Cognitive Engine architecture,** adapted from [13]

*Cognition Loop*

To guide our Cognitive Engine development, we proposed a new version of the cognition loop [14] shown in Figure 5. This new version is based on Mitola's original cognition cycle definition [2] which consists of "observe," "orient," "plan," "decide," "learn," and "act." Our cognition loop steps are to (1) collect the radio environment parameters; (2) synthesize the information into scenario representation; (3) compare scenarios with the radio and user database and decide either to use an existing successful case setting or find another better case setting; (4) find the optimal setting using some optimization algorithms if a new optimal setting is needed; (5) update the database using the new optimal parameter setting; (6) reconfigure the attached radio platform for this scenario using the optimal parameter setting. Our cognition loop is more closely directed towards PHY and MAC level cognition than Mitola's definition which is more like human cognition.

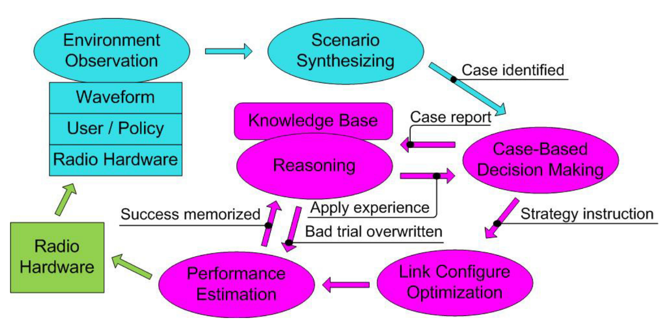



**Figure 5 - Cognition loop,** adapted from [14]

*The Cognitive Engine Execution*

In the Cognitive Engine, the cognitive controller follows the cognition loop and coordinates and executes attached function components by sending commands and data through an interface to the components. The learning and adapting ability is achieved through the cooperation of the seven components.

The sensor observes external and internal environments. For example, one signal sensor can collect external surrounding environment parameters including propagation path, radio position and location, and the spectrum availability, etc. Another sensor can collect internal data, or "read the meters", which displays the radio's self performance and operating parameters such as received signal power, noise power, bit error rate (BER), frame error rate (FER), and battery life, etc. The decision maker and knowledge base will first perform scenario synthesizing and case-based decision making, and then they will estimate performance from the optimizers' result and update the knowledge base. Our optimizer, the multi-objective based wireless system genetic algorithm (WSGA), will perform the link configure optimization. The policy verifier, together with the decision maker, verifies the optimizer's result to follow regulatory rules and updates the knowledge base. The radio platform, which can be any SDR system with both hardware and software architecture, will receive the policy verifier's results and reconfigure itself, or "turn the knobs" such as transmit power, modulation, coding, symbol rate and spectrum shaping, to achieve a good QoS user setting. In addition, the human friendly user interface displays some necessary parameters and creates command buttons bound to the cognitive controller's control commands.

Each component is launched as a separate process that interfaces and exchanges data between processes through some generic interface. This architecture has two advantages. First, it enables distributed processing, where different components can reside on one single processor, as in our current system, or it can reside on different processors or hosts [13]. The other advantage is that this architecture adopts a standard interface to enable future changes or development of components and algorithms. Upgrades of any component will not affect other components. This advantage simplifies the developing and testing of the cognitive radio system.

*Application Programmable Interface (API)*

The cognitive engine's architecture openness is achieved through the attachable function modules, the cognitive controller, and the application programmable interface (API) which interfaces the components. Essentially, different components need a common language to exchange domain knowledge and parameters. In addition, a mechanism is also required to coordinate parameter and command sending and receiving among different components without any conflict and intolerant time delay. To achieve the above function, we created the API which rides on the eXtensible Markup Language (XML). XML is used as the common language to convey command, domain knowledge, and parameters among different components. The operating system possessed Internet Protocol-based network socket functions are used for sending and receiving the aforementioned XML files. For example, the sensed data such as propagation path, radio position and location, received signal strength, noise power, and BER are first formulated as a hierarchical data structure stored in XML files in the sensor domain. Then the XML files are transferred to the cognitive controller through the API. The cognitive controller will parse the XML files and get the original sensed data.

*Wireless System Genetic Algorithm (WSGA)*

WSGA is a multi-objective based genetic algorithm designed to find the optimal parameter setting for configuring the radio platform by responding to the user's requirement and the QoS. It models the physical radio system as a biological organism and optimizes its performance through genetic and evolutionary processes.

Radio resources encompass spectrum, power, time, etc, which determine QoS for communication. The QoS can be commonly specified through eight important objectives which include: BER, signal to interference plus noise ratio (SINR), bandwidth, spectrum efficiency, throughput, power, computational complexity, and interference [15]. Finding the above optimal parameter setting in terms of radio resources is modeled as a multi-objective problem. Objective space is defined as the set of objectives that represent the radio performance with each one modeled as a function of resources. Multiple objects are usually needed to fully describe the radio performance and they may not be independent from each other. The multi-objective problem is modeled as:

$$\min/\max(\bar{y}) = f(\bar{x}) = [f_1(\bar{x}), \dots, f_n(\bar{x})] \quad (5)$$

subject to:

$$\bar{x} = (x_1, x_2, \dots, x_m) \in X$$
$$\bar{y} = (y_1, y_2, \dots, y_n) \in Y$$

where $Y$ is the objective space, and $X$ represents the resources.

The WSGA uses chromosomes to encode input parameters like payload size, power, coding techniques, encryption, equalization, number of sub-carriers, network protocol, retransmission requests, and spreading technique/code. It uses a Pareto ranking selection method [16] to determine the chromosomes' survival to the next generation in terms of their fitness to maximize the objectives. During the fitness evaluation, the WSGA awards points for every objective that an individual wins. The WSGA uses one crossover and mutation point operation chosen from uniform random numbers with a static probability of crossover and mutation occurring. The use of constraints to a multi-objective problem gives the WSGA the opportunity to incorporate regulatory and physical restrictions during chromosome evolution. If a trait determined by the chromosome exceeds the limits of the radio's capabilities, like finding a center frequency outside the tunable range of the radio, or breaks the law by transmitting too much power in a band, then the WSGA forces random mutations on the gene until it is legal, thus preserving large portions of the chromosome structure as well as introducing legal genes into the population. The WSGA terminates when it reaches a desired level of objectives or the specified maximum number of generations.

In Figure 6, we show how the WSGA optimizes the waveform. The XML file which defined the waveform describes the available knobs and the range SDR is capable of. Objective functions are received from the controller to describe functions in a library, and the WSGA calls the functions to calculate meters from the current set of knobs.

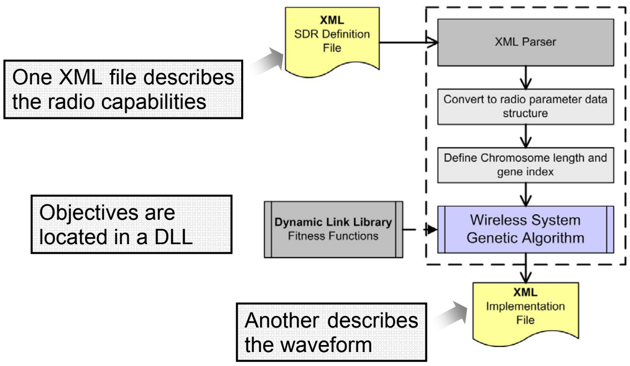



**Figure 6 - WSAG waveform optimization**, adapted from [13]

*Case Based Decision Maker*

Built on case-based theory, the decision maker combined with the knowledge base uses feedback to aid future optimization which will heuristically learn from experience instead of using pure object optimization. The optimized problem is saved in the data-base with the solution and performance, and when the new problem is received, the Cognitive Engine is looking for the similarly optimized problem [13]. The similarity between the new problem and old cases is input as one of the augments of objective function. This is how the Cognitive Engine learns from experience and knowledge. The multi-objective problem is now described as

$$\min/\max(\bar{y}) = g(\bar{x}) = [f_1(\bar{x}), \dots, f_n(\bar{x}), s_1, s_2, \dots, s_k] \quad (6)$$

subject to:

$$\bar{x} = (x_1, x_2, \dots, x_m) \in X$$
$$\bar{y} = (y_1, y_2, \dots, y_n) \in Y$$

Where $X$ and $Y$ are same as in Equation (5) and $s_i (i = 1,2,..,k)$ is the similarity between the new problem and the cases stored in the database. In Figure 7, it shows how a case-base is applied to the optimization process to learn from feedback.

This mechanism offers a significant advantage to real time processing in a cognitive engine where a quick solution needs to be provided as situations and environments change. It can find a sub-optimal parameter setting good enough to support a QoS level within a short time instead of finding the best parameter setting in a long time. On the other hand, it can also narrow the searching space for the optimizer which starts searching at the local variable space including a similar stored case and does not have to search the whole variable space. However, the database size and its data organization will affect the performance in that only a certain amount of cases can be stored in the database. The Cognitive Engine's decision rule governs the organizing and manipulation of cases relative to time and prioritization. It uses the maximum utility forgetfulness [13] which replaces the case with the lowest utility with the new case. For our simulation experiment, a maximum of 100 cases can be stored in the database.

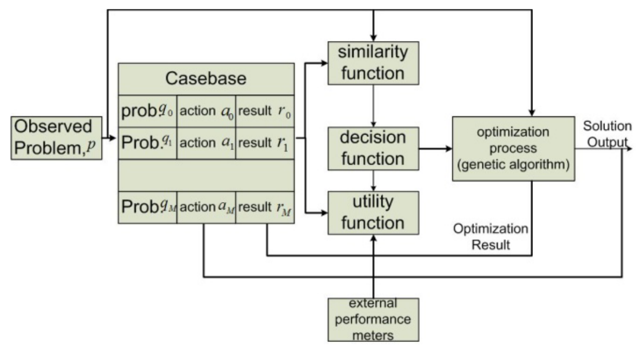



**Figure 7 - Case-base application to optimization process**, adapted from [13]

*Simulation and Experiment of Cognitive Engine*

To verify the Cognitive Engine's learning and adapting ability, Thomas W. Rondeau carried out a sequence of simulation and over the air experiment [13] which generated new parameter settings subject to different QoS, here different objectives. Both simulation and experiment were performed on the Cognitive Engine with selected function components of GNU Radio SDR platform with USRP RF front-ends, power spectral density (PSD) sensor and WSGA optimizer. During the simulation [13], the Cognitive Engine collected a set of meters such as transmit power and symbol rate. It also specified a set of objectives such as BER and SINR. The optimizer, i.e., WSGA, combined with the decision maker, was able to find an optimal parameter setting within 400 generations and tested the performance subject to the objectives.

For the experiment [13], Thomas W. Rondeau set up a digital communication link between two Cognitive Engine nodes with one master and one slave. Given the objective of high data throughput to the slave with low error, the master was able to find the vacant spectrum among several interference nodes and configured the radio to produce a 200 kbps QPSK signal with a 12 dBm transmit power within a short time. In addition, the waveform's frequency and power were verified by the policy engine through a regulatory spectrum mask.

## 4. Public Safety Cognitive Radio

As a system level application, we attach a specific set of components to the Cognitive Engine and design the prototype PSCR especially for public safety communication [17], as shown in Figure 8. It can scan the public safety spectrum (multiple bands and multiple waveforms, all incompatible) and configure itself to interoperate with any public safety waveform. The reconfiguration happens within 0.1 seconds after its signal sensor determines that a signal is present.

The PSCR aims at providing universal interoperable communication service of voice and data to solve the interoperability problem. This problem is that various incompatible public safety waveforms cannot communicate with each other. This is a widely existing and severe problem [18]. The PSCR is designed to sense the frequency band of interest, detect and identify existing public safety waveforms and networks, and then be reconfigured to talk to any detected channel. Furthermore, it serves as a gateway to bridge incompatible waveforms, different frequency bands, and networks. It can also serve as a multi-mode multi-band wireless terminal for the user.

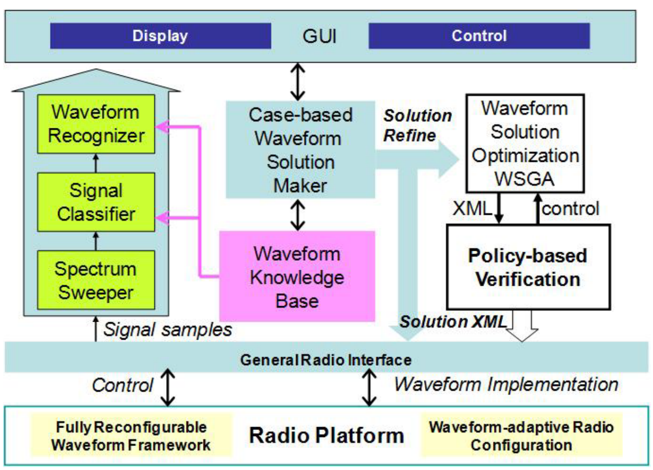



**Figure 8 – PSCR System Block Diagram,** adapted from [19]

*PSCR Architecture*

Corresponding to the Cognitive Engine architecture, we have selected GNU Radio as the radio platform and developed several important functions in order for radio reconfigurability to be achieved. We have developed a GUI which will function as both the cognitive controller for controlling and the user interface for displaying. The sensor includes a spectrum sweeper, which is an energy detector for signal detection and a signal classifier for waveform recognition. The waveform knowledge base stores public standard waveform parameters which are used to help the signal classifier and also to determine radio platform settings through the case-based waveform solution maker. The API uses XML for data formulation and sends data and command through Internet Protocol-based network socket functions. The optimizer and policy verifier are currently being implemented for possible other functions in the PSCR. We will introduce most components in detail together with the functions they provide in the following section.

*PSCR Working Modes*

PSCR provides three working modes to achieve its design goal:

**Scan mode** is used for a PSCR node to detect any active signal in a specified spectrum range and recognize its waveform parameters and modulations. Its function is achieved through the signal sensor.

**Talk mode** is used for the PSCR to establish a link with any detected standard public safety network for voice and data communication. Once a network is specified by the user, either selected from our database or detected from our sensors, the radio platform will reconfigure itself to establish a communication link between the specified network and our PSCR. Currently PSCR supports the analog push-to-talk function and digital link based on several waveform modulations.

**Gateway mode** is used to bridge two incompatible public safety networks.

The above three working modes are displayed and controlled through our graphical user interface (GUI) which is designed in the Java. The snapshots are shown in Figure 9 for scan mode, talk mode, and gateway mode respectively from bottom to top.

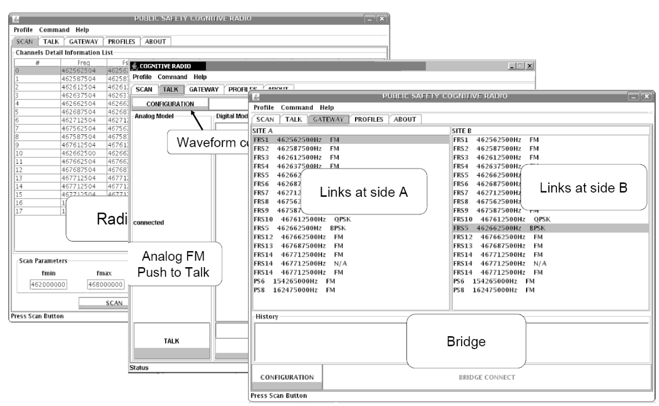



**Figure 9 – PSCR GUI Display,** adapted from [17]

*PSCR Sensor*

Currently, the PSCR uses a specially designed sensor which includes the spectrum sweeper, the signal classifier, and the waveform recognizer. The spectrum sweeper is essentially an energy detector based on FFT. It uses GNU Radio and USRP to collect signal samples, just the same as our SSCR's channel monitor. The signal classifier is based on a

K-nearest-neighbor (KNN) algorithm and utilizes the fact that the averages of the standard deviation of the complex envelope of public safety waveforms are different at the same SNR [20]. The SNR value for a detected signal is computed through our energy estimation in the spectrum sweeper. The signal classifier is trained beforehand through typical public safety waveforms at different SNR values. The threshold borders between different waveforms at each SNR value is identified and will be used to classify any detected waveform's modulation.

The waveform knowledge base, implemented in a MySQL database system, stores the training result and the standard public safety waveform specifications. It is used to determine the standard parameters for public safety waveforms after the sensor identifies the spectrum and modulation of any detected signal.

*Radio Platform Reconfiguration at PHY/MAC Layer*

The PSCR's reconfigurability is achieved through the radio platform based on GNU Radio and USRP. It is fully reconfigurable at the PHY and MAC layers for the above working modes. We designed a multithreaded control system to coordinate the reconfiguration at MAC and PHY layers, as shown in Figure 10.

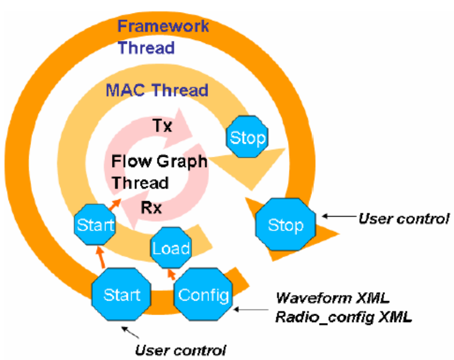



**Figure 10** – **CWT Waveform Framework Multi-threading Control**, adapted from [19]

The framework thread will receive commands from the GUI controlling function, and it will get parameters from the sensor or the knowledge base. All the commands and parameters are formulated in XML for exchanging. According to the command, the framework thread initiates and coordinates the three working modes. Correspondingly, it parses the XML file and extracts the right parameters for a specific application's PHY/MAC layer configuration, as shown in Figure 11.

The flow graph thread creates waveforms and their modulations, demodulations, filtering, coding, decoding, etc. Its name comes from GNU Radio's concept of "flow graph." The flow graph thread coordinates with the MAC thread to transmit and receive voice and data. For the data link example, the flow graph thread receives the necessary parameters and initiates a new signal "flow graph" consisting of filter, gain (control), demod(ulation), FEC (forward error correction), etc. It is coupled with a new MAC sensing function, as shown in Figure 12. Most individual signal processing blocks are available in GNU Radio and the radio RF front end is based on USRP. Based on this data link, we are able to build a wireless IP network through GNU Radio's virtual network interface: the TUN/TAP open source library [21]. The application layer of an existing Internet Protocol-based network seamlessly works with the above wireless network through interfacing with the TUN/TAP function block. We tested text messaging, web browsing, voice over IP, and audio streaming.

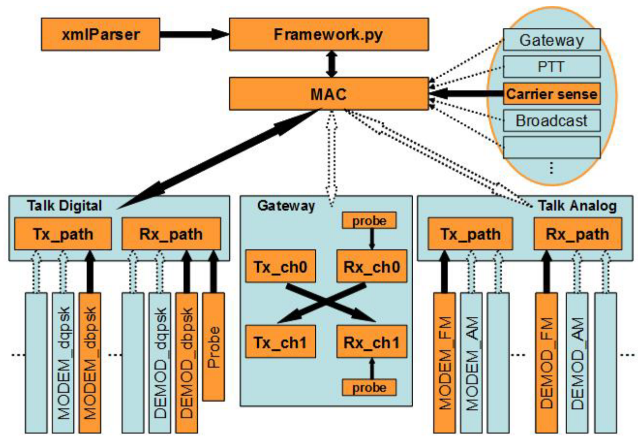



**Figure 11 – PSCR PHY/MAC Waveform Framework,** adapted from [19]

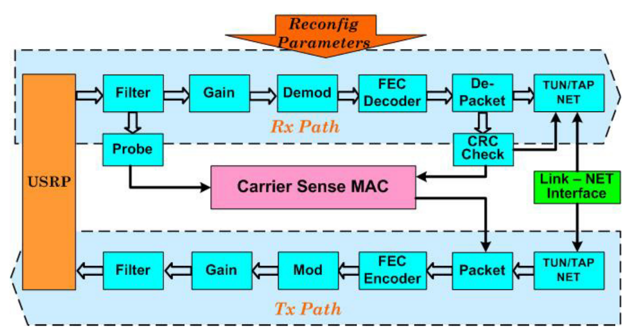



**Figure 12 – PSCR PHY/MAC Waveform Framework,** adapted from [19]

The current Radio Platform can support legacy FM, narrowband BPSK, QPSK, 8PSD, and GMSK digital modulations at multiple public safety channels. It can support analog FM talk in talking mode and the gateway mode. For example, we bridged the Cobra MicroTalk PR 100-2VP (22 Channels) 2-Way Radio and a Motorola Radius P110 police hand-held radio.

## 5. Conclusions

In this paper, we introduce our cognitive radio development from spectrum sharing to cognitive radio learning and adapting. We design an open cognitive radio architecture and implement several functions with simulation results shown. Furthermore, we develop the PSCR and demonstrate our cognitive architecture's feasibility.

However, we still need to add more functions to accomplish the above mentioned functions. We also need to add broadband waveforms such as OFDM.

Furthermore, the full cognitive radio system is a significant software undertaking, which includes not only a software defined radio, but also a plethora of modules responsible for cognition, reasoning, decision-making, etc. The success of such a complex system depends on the reliability and quality of the embedded software, the full re-configurability of the radio hardware, and the high computation speed of the computing component. In particular, several critical cognitive radio modules require advanced computation performance beyond the abilities of today's best General Purpose Processor (GPP). We are working to port the whole system into the PlayStation 3 powered by Cell BE.

## 6. Acknowledgements

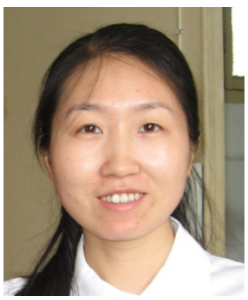
This work was supported by the National Science Foundation under grant CNS-0519959 and by the National Institute of Justice, Office of Justice Programs, US Department of Justice under Award No. 2005-IJ-CK-K017. The opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation or the Department of Justice.

## References

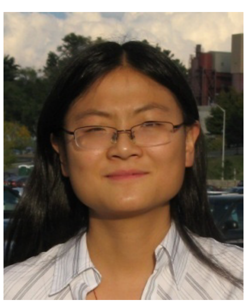
[1]"Shared Spectrum Company," www.sharedspectrum.com.
[2] J. Mitola, "Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio," Royal Institute of Technology (KTH), 2000.
[3] J. Mitola, *Software Radio Architecture: Object Oriented Approaches to Wireless Systems Engineering*: John Wiley and Sons, 2000.
[4] K. N. Steadman, A. D. Rose, and T. T. Nguyen, "Dynamic Spectrum Sharing Detectors," in *New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, Dublin, Ireland, 2007. .
[5] J. H. Reed, *Software Radio: A Modern Approach to Radio Engineering,* Englewood Cliffs, NJ: Prentice-Hall, 2004.
[6] Ettus Research, LLC., "The USRP," *http://www.ettus.com/*.
[7] E. Blossom, "Exploring GNU Radio," *http://www.gnu.org/software/gnuradio/doc/exploring-gnuradio.html*, November 2004.
[8] F. Ge and C. W. Bostian, "A Wideband Spectrum Sensing Approach with Agility and Low SNR Sensitivity," in *Software Defined Radio Technical Conference,* Denver, Colorado, 2007.
[9] D. Cabric, S. M. Mishra, and R. W. Brodersen, " Implementation issues in spectrum sensing for cognitive radios," in *Asilomar Conference on Signals, Systems, and Computers*, 2004, pp. 772 - 776.
[10] R. S. Roberts, W. A. Brown, and J. H. H. Loomis, "Computationally efficient algorithms for cyclic spectral analysis," in *IEEE Signal Processing Magazine*. vol. 8, 1991, pp. 38-49.
[11] N. J. Carter, "Implementation of Cyclic Spectral Analysis Methods," vol. Master: Naval Postgraduate School, Monterey, CA, 1992.
[12] K. Kim, I. A. Akbar, K. K. Bae, J. Um, C. M. Spooner, and J. H. Reed, "Cyclostationary Approaches to Signal Detection and Classification in Cognitive Radio," in *New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, Dublin, Ireland, 2007, pp. 212-215.
[13] T. W. Rondeau, "Application of Artificial Intelligence to Wireless Communications." vol. PhD: Blacksburg: Virginia Polytechnic Institute and State University, 2007, p. 198.
[14] T. W. Rondeau, C. W. Bostian, D. Maldonado, A. Ferguson, S. Ball, and B. Le, "Cognitive radios in public safety and spectrum management," in *Telecommunications Policy and Research Conference,*. vol. 33, Sept. 2005.
[15] J. G. Proakis, *Digital Communications*, 4 ed.: New York: McGraw Hill, 2000.
[16] R. Purshouse and P. Fleming, "Evolutionary many-objective optimisation: an exploratory analysis," in *IEEE Congress on Evolutionary Computing,*, 2003, pp. 2066 - 2073.
[17] B. Le, P. Carcia, Q. Chen, B. Li, F. Ge, M. Y. EI-Nainay, T. W. Rondeau, and C. W. Bostian, "A Public Safety Cognitive Radio Node System," in *Software Defined Radio Technical Conference* Denver, Colorado, 2007.
[18] J. Brito, "Sending Out an S.O.S.: Public Safety Communications Interoperability as a Collective Action Problem," *Federal Communications Law Journal,* vol. 59, pp. 457-492, 2007.
[19] B. Le, "Building a Cognitive Radio - From Architecture Definition to Prototype Implementation," in *Electrical and Computer Engineering*. vol. Doctor of Philosophy: Virginia Polytechnic Institute and State University, 2007.
[20] B. Le, "Modulation Identification Using Neural Network for Cognitive Radios," in *Software Defined Radio Forum Technical Conference*, Anaheim, CA, 2005.
[21] "Virtual Point-to-Point(TUN) and Ethernet(TAP) Devices," in *http://vtun.sourceforge.net/tun/*.
[22] "Improving Spectrum Usage through Cognitive Radio Technology," *IEEE USA Position*, Nov 13, 2003, Available online: http://www.ieeeusa.org/policy/positions/cognitiveradio.asp.
[23] Federal Communications Commission, "NOTICE OF PROPOSED RULE MAKING," http://www.fcc.gov/Bureaus/Engineering_Technology/Notices/2000/fcc00430.doc, December 8, 2000.

## BIOGRAPHY

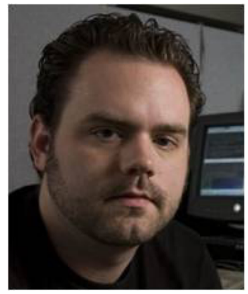
***Fe Ge*** *is currently a PhD student in the Department of Electrical and Computer Engineering at Virginia Tech, Blacksburg, VA. His PhD research is primarily focusing on cognitive radio development in the Cell processor. He received his master's degree in Computer Science and Engineering from the University of South Carolina in 2006. From 2004 to 2006, he worked as a research assistant in computer vision and image processing during his master's degree pursuit. He received his Bachelor's degree in Engineering Mechanics from Tsinghua University, Beijing, in 2002.*



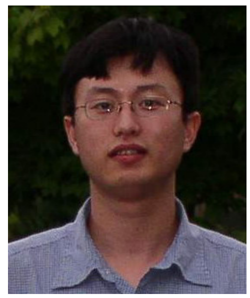
***Qinqin Chen*** *is currently a PhD student in the Department of Electrical and Computer Engineering at Virginia Tech, Blacksburg, VA. Her research interests primarily include digital gateway design for cognitive radio and reconfigurable SDR platform. She received her Master's degree in Communication and Information Systems from University of Science & Technology of China in 2005 and Bachelor's degree in Electrical Information Engineering from Wuhan University, China in 2002. Before transferring to Virginia Tech in 2006, she spent almost one year on the research of VLSI Design for Soft-decision Decoding of Reed-Solomon Codes, as a first-year Ph.D. Student majoring in Computer Engineering at Oregon State University.*



***Ying Wang*** *is currently a PhD student in the Department of Electrical and Computer Engineering at Virginia Tech, Blacksburg, VA. Her research interests primarily include wideband cognitive radio and cognitive OFDM development. She received her Master's degree from the University of Cincinnati in 2006 working in Wireless Communication Lab focusing on wideband and fast fading channel communication research. She received her bachelor's degree in Information Engineering from Beijing University of Posts and Telecommunications, China in 2003.*



***Thomas W. Rondeau*** *is a research engineer with the Centre for Telecommunications Value-chair Research (CTVR) at Trinity College, Dublin, Republic of Ireland. He received his Ph.D. from the Bradley Department of Electrical and Computer Engineering at Virginia Tech in September of 2007 under Dr. Charles Bostian. His current research interests are in improving communications through software and cognitive radio. Tom received his B.S. degree in Electrical Engineering from Virginia Tech in May 2003, graduating Summa Cum Laude, and he received his M.S. degree in Electrical Engineering from Virginia Tech in 2005. Tom is also a developer in the open source GNU Radio project.*



***Bin Le*** *received both his M.S. and Ph.D. degrees in Electrical Engineering at Virginia Polytechnic and State University, where he served as a research assistant in the Center for Wireless Telecommunications (CWT). After graduation, he joined Cognitive Radio Technologies (CRT), LLC as a system engineer. He also serves as a research associate for Wireless @ Virginia Tech at the same time. His research interests include cognitive radio and network, software defined radio and machine learning algorithms. He is a member of IEEE Communications Society, and a member of cognitive radio working group of Software Defined Radio Forum (SDRF).*

***Charles W. Bostian*** *received the Ph.D., M.S., and B.S. (with highest honors) in Electrical Engineering, all from North Carolina State University, in 1967, 1964, and 1963, respectively. Dr. Bostian joined the Virginia Tech faculty in 1969 following a short period of employment with Corning Glassworks Electronics Research Laboratory and service as an officer in the U.S. Army. From 1972 through 1988, he headed Virginia Tech's Satellite Communication Group. Since 1993 he has directed the University's Center for Wireless Telecommunications (CWT). Bostian is the co-author of two widely used textbooks, Solid State Radio Engineering and Satellite Communications, now in its second edition. His current teaching interests are in RF design and in undergraduate circuit analysis. Bostian's primary research interests are in cognitive electronics and radio system design. Currently he directs National Science Foundation (NSF), National Institute of Justice (NIJ) and Defense Advanced Research Projects Agency (DARPA) projects on cognitive radio. He holds a Virginia Tech Alumni Distinguished Professorship and is a Fellow of the IEEE.*