GENERAL RADIO INTERFACE BETWEEN COGNITIVE ALGORITHMS AND RECONFIGURABLE RADIO PLATFORMS

Bin Le (bin.le@crtwireless.com, Cognitive Radio Technologies, Lynchburg, VA),
Thomas W. Rondeau, and Charles W. Bostian (Center for Wireless Telecommunications, Virginia Tech, Blacksburg, VA)

ABSTRACT

In cognitive radio (CR) systems, an interface is needed to convey information between the cognitive, application algorithms (i.e., the cognitive engine (CE)) and the supporting radio platform. In this paper, we explicitly define such an interface as a set of software functions, documents, and additional application tools required for the CE to recognize, configure and control the radio platform.

In the paper we describe how the radio system is represented to the CE, how the CE represents generic waveform parameters, and how platform independent concept is implemented in the interface design for the CE to control different radios.

1. INTRODUCTION

We define a cognitive radio (CR) as an intelligent communication device that is aware of its environment and application needs and reconfigure itself to optimize quality of service [1]. A general CR solution is defined in the form of a software package that can work with reconfigurable radio platforms to provide cognitive functionality. This software package, called cognitive engine (CE), consists of a set of general machine learning and application specific algorithms and it can be applied to radio platforms with hardware implementation. As shown in the CR system architecture block diagram in Figure 1, the CE manages radio resources and adapts radio operation for performance optimization. The interaction between the CE and the radio platform is through a standard interface in between.

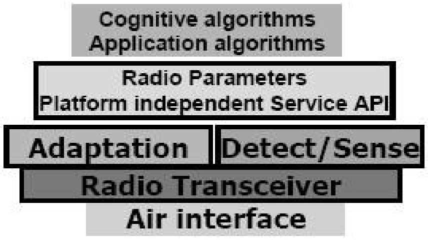

Figure 1. Cognitive radio system model

Although the CE can work with any radio that is able to provide needed reconfigurability, the software defined radio (SDR) platform is preferred for its maximal degree of flexibility and reconfigurability [2]. The radio passes information to the CE with a standard parametric format, called a profile. The CE makes a decision and passes the decision to the radio also in the profile format for the radio to achieve performance requirements.

Such a platform independent interface model, called the "egg model", is shown in Figure 2. It reflects the hardware independent design principle of our CR solution, which makes it possible to bridge machine learning with radio operation. Platform independent interface has two design flexibilities: (1) it supports general machine learning and application specific algorithms; and (2) it supports heterogeneous SDR and reconfigurable radio platforms.

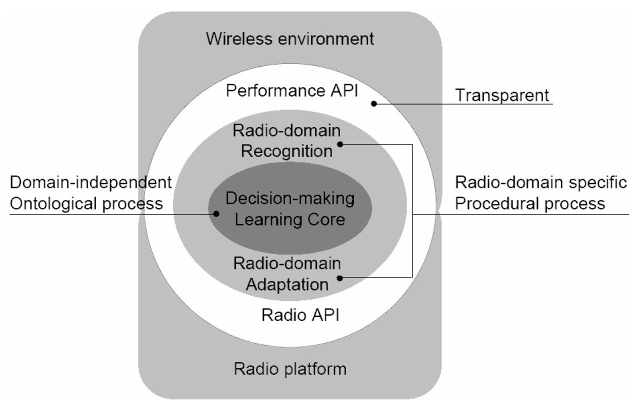

Figure 2. The egg model of general radio Interface

In the following of this paper, section 2 introduces domain knowledge profile concept and radio related profiles definition; section 3 details the radio platform abstraction and knowledge design; section 4 details the waveform solution interface and the resulting radio configuration and control structure; section 5 illustrates the whole interface integration between CE and radio platform; finally a summary is given in section 6.

2. INTERFACE PROFILE

2.1 Radio Interface Related Profiles

The importance of domain knowledge design for machine learning has been well explained in [2] and [3]. For our CR radio system, the domain knowledge design is to enable machine reasoning and learning for radio applications. It can be divided into two steps: the first is to pick efficient methodologies of observing and modeling useful domain information and abstract such information with a general representation for machine reasoning and learning, such as CBR-GA learning core in our CR system [1]; the second is to implement a suitable language to describe such domain knowledge for information transfer between intelligent units.

The representation of domain knowledge is called a profile (See the detailed ontological definition in chapter 12 of [2].). The format of a domain knowledge profile depends on the modeling method and characteristics for the specific domain [4]. Understanding that the CR system sits at the center of the radio, user and policy domains [5], we define the following domain knowledge profiles related to the radio interface at the current developmental stage:

- *Radio environment profile*: contains the information recognized from the radio environment, including spectrum energy occupancy, waveform format at channels of interest, and available link opportunities.
- *Performance profile*: contains a set of metric parameters as the evaluation of radio performance, such as BER, PER, SINR, data throughput, power consumption, etc. These metrics are platform independent and thus can be applied to different radio hardware system.
- *Radio platform definition profile*: contains the list of available radio functional resources, their related processing capability in terms of dynamic range and working modes, and their functional characteristics like dependencies, handling rules, etc.
- *Waveform profile*: contains the communications-function level representation of a waveform specification, such as carrier frequency, modulation scheme, pulse shape, symbol rate, coding scheme, frame format, MAC protocol, etc. The waveform profile gives a complete definition of a waveform that can be implemented on different radio platforms.
- *Radio platform configuration profile*: contains the configuration parameter set of the radio platform to carry required waveform and link operations. Such a configuration is platform specific and the settings inside depends on the required radio functionalities.

Representing domain knowledge in the form of domain profiles enables the independence of information processing methodology from the specific information content. For the CR system, it ensures that a general machine reasoning and learning structure can be designed and applied across multiple domains. An example of CR reasoning and decision process with multiple domain profiles is shown in Figure 3. The CE takes in radio definition profile to form dimensions and boundaries of its supported operational space; and takes in radio environment, policy and user profiles to form performance objectives and reasoning associations, so that a viable solution can be projected and optimized [5]. The generated solution is then formulated into waveform and radio configuration profiles with a standard format that can be applied to various radio platforms.

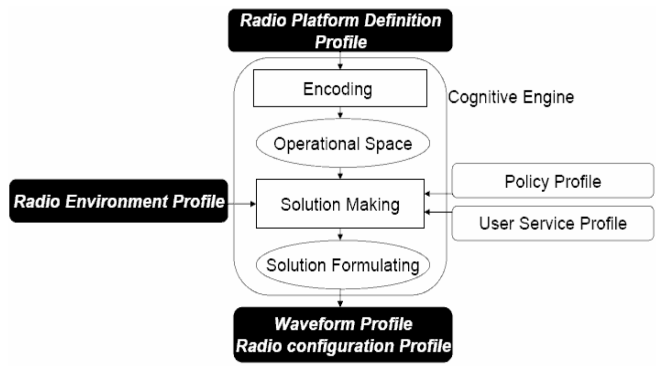

Figure 3. Radio cognition with domain profiles

2.2 Profile Description Format: XML

As a profile is defined to stand for the domain knowledge, it needs to be described in a standard data format to transfer such information between different processing units. Such profile describing format is termed language in our CR system design. Machine learning is based on an information processing structure of a computational system [6], and such information processing is symbolic based. The domain language can be viewed as a separate interface component for the information transfer and processing between individual intelligent functional units.

Since language is the vehicle of the domain profile, the key challenge of profile design, which is to synthesize the massive, discrete information collected directly from domain observation, is translated to the language design to create a script with a data structure that not only remains generally applicable for various information processing interfaces but also bears a resemblance in parametric format to support machine logic and calculation. It also should be legibly descriptive for human understanding.

To balance between speed and flexibility, we choose extensible markup language (XML) to describe the domain profile. XML has been developed since 1996, simple in structure and concise in content. It has been used by a wide variety of applications, and is human legible [7]. XML provides a generic solution to information connectivity between programs otherwise difficult to interface. It follows the unified modeling standard (i.e., the UML [8]) thus provides object-oriented, hierarchical-structured, self-explanatory scripting method with which we can flexibly grow and define data structures as well as data content.

XML scripting is standard and can be realized regardless of programming languages like C/C++, Python, Java, etc. A machine reads XML data by a simple data parsing process

In our CR system, the radio platform passes domain knowledge to the CE in XML, containing the profiles such as radio resource, radio environment, and radio performance. The CE tells the radio in XML the solution profiles of the waveform to implement and the radio operational settings. The CR knowledge base is applied and evolved with the profile information in XML as well. Via XML all the knowledge profiles can be shared and transferred across the network as well as inside one CR node.

3. RADIO PLATFORM KNOWLEDGE

If a CE wants to control the radio, it must first know the radio's capabilities: it would be worthless to create a waveform solution not supported by the radio. We must therefore have a representation of the radio, i.e. the radio platform profile, that abstracts the radio's implementation details [9], thus can represent various radio platforms regardless of their hardware realization. For example, the radio platform profile does not explain how a specific modulation or channel coding is implemented inside the radio, but only what modulations or channel coding algorithms are available with needed configuration.

In an ideal SDR platform, all modulations and all channel coding algorithms can work with each other, but perhaps (usually) a specific SDR only allows certain block sizes with a modulation of a certain order. When learning the radio's capabilities, if such dependency limitation exists, it must be properly addressed in the profile.

The radio platform profile is provided by the radio maker and preloaded to the CE. Although this is not an elegant solution, it avoids an additional query protocol layer between the radio platform and cognitive algorithms, which is very helpful in improving system efficiency and real-time response, especially for radio applications. Profile loading helps the CE form a more confident operational space than ad-hoc querying and machine understanding. Normally, at one CR node during one operational session, it is unlikely for the CE to switch from one platform to another. Therefore, this platform knowledge preloading approach is suitable. Radio capabilities do not change fast, and so the listing of these capabilities will be done once by the radio maker, with periodic updates that will correspond to device or software upgrades.

The unit and value of the parameters in the radio profile XML description can be coded totally device specific as far as these XML scripts of different radios share the same scripting format and the same data structure. Therefore, there is no need (and practically very difficult) to ask various radio platforms to obey the exact same performance evaluation criteria. The radio definition profile design follows the object oriented approach in modeling the radio. Each processing resource element in the radio platform is viewed as an object, and its related characteristics, such as working modes, processing range, data I/O format, and computational cost are coded as the attributes of this object. The hierarchical functional dependency between radio resources elements are also inherently encoded by XML scripting.

XML formats data in a tree structure. To represent a radio in XML, the radio node first branches off into receiver and transmitter branches; then it branches into components like RF front-end and modem with different working modes and operational ranges that the radio supports. The leaf nodes contain data, or values, associated with each component, such as symbol rate. Each node may contain a set of attributes, such as type and unit, to enhance adaptive parsing. Following the tree structure of XML, a parser can be completely agnostic of the data format, and the CE can easily "walk" through the tree to find components, values, and attributes of radio resources. A partial radio platform profile XML script is shown in Figure 4. Complete set of radio platform profiles are implemented in XML in our CR node system [10] [5].

```
<Tx>
    <PHY>
        <rf>
            <tx_freq units="Hz" mult="1"> ...
            <tx_power units="mW" mult="0.001"> ...
        </rf>
        <rf> ... another supported RF ... </rf>
        <mod type="PSK">
            <tx_mod_bits> ...
            <tx_mod_differential> ...
            <tx_rolloff units="na"> ...
            <tx_gray_code> ...
            <tx_symbol_rate units="Hz" mult="1">
        </mod>
        <mod type="XXX"> ... another mod ...
    </PHY>
    <PHY> ... another supported PHY ... </PHY>
    <LINK>
        <frame>
            <tx_pkt_size units="bytes"> ...
            <tx_access_code> ...
        </frame>
    </LINK>
    <LINK> ... another link ... </LINK>
</Tx>
<Rx>
    <PHY> ... Rx PHY ... </PHY>
    <LINK> ... Rx Link ... </LINK>
</Rx>
```

Figure 4. Partial radio platform profile XML

4. RADIO CONFIGURATION AND CONTROL

As shown in Figure 3, the processing flow starts and ends with the radio platform. The CE must first know the radio platform's capabilities, to setup the basis of the operational space; then make waveform (currently including PHY and MAC) solutions by reasoning knowledge profiles from all domains. The solution is then passed back to the radio platform via the solution part of the CE-radio interface. The solution interface structure is shown in Figure 5. The solution profiles are interpreted into configuration and control instructions, and then mapped to the radio-provided application programmable interface (API) and control logic handler respectively. These API and control logic handler associated services are already registered as radio resources in the radio platform definition profile that is preloaded to the CE. Therefore, one configuration and control instruction set contained in a solution can be viewed as one specific setting of the radio resource layout.

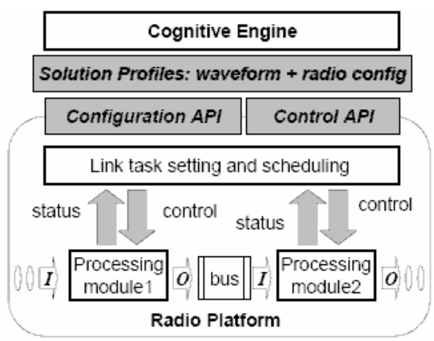

Figure 5. Solution interface between CE and radio platform

More specifically, in the current development stage of our CR system, the cognitive capability mainly focuses on PHY and MAC layers, and thus the solution made by the CE focuses on the waveform that is to be deployed in the radio platform. Therefore, a CR generated solution mainly consists of two parts, waveform profile and radio configuration profile. They jointly determine the configuration and control of a specific link session to be launched, maintained and adapted by the radio. A partial waveform profile XML and its associated radio configuration profile XML are displayed in Figure 6 and 7. The radio configuration profile is to configure the radio to support the waveform specified by the waveform profile.

The API is the set of functions to invoke the operations required from the interface. It is provided by the radio platform for CE to configure the radio's waveform service setting and control the radio's processing behavior. There are two types of API defined in CE-radio interface: configuration API and control API. The configuration API can be viewed as a set of knobs with specific values; and the control API is a set of algorithm handlers to control the behavior of the radio operation, such as duplexing control, data frame timing, MAC timing control, state switching with specific transceiving modes, etc [5].

```
<?xml version="1.0" encoding="utf-8"?>
<waveform type="digital">
  <MAC>
    <mac_protocol type="carrier_sense">
      <carrier_threshold units="dBm">40</carrier_threshold>
      <contention_window units="second">0.5</contention_window>
      <min_delay units="second">0.001</min_delay>
    </mac_protocol>
  </MAC>
  <Tx>
    <PHY>
      <rf>
        <tx_freq range="100e6,500e6" units="Hz">462662500</tx_freq
        <tx_power range="0,100" units="mW">0.1</tx_power>
      </rf>
      <mod type="psk">
        <tx_mod_bits>1</tx_mod_bits>
        <tx_mod_differential>1</tx_mod_differential>
        <tx_roll_off range="0,1" units="na">0.35</tx_roll_off>
        <tx_gray_code>1</tx_gray_code>
        <tx_symbol_rate range="10e3,500e3" units="symbols/s">3500C
      </mod>
    </PHY>
    <LINK>
      <frame>
        <tx_pkt_size range="1,1500" units="bytes">512</tx_pkt_size
        <tx_access_code>None</tx_access_code>
      </frame>
    </LINK>
  </Tx>
  <Rx>
    <PHY> ... </PHY>
    <LINK> ... </LINK>
  </Rx>
</waveform>
```

Figure 6. Partial waveform XML: BPSK at VHF band

```
<?xml version="1.0" encoding="utf-8"?>
<radio_platform type="GNU Radio">
  <Debug> ... </Debug>
  <Radio>
    <tx_usrp_subdev_spec_a>rfx2400</tx_usrp_subdev_spec_a>
    <tx_usrp_subdev_spec_b>rfx400</tx_usrp_subdev_spec_b>
    <rx_usrp_subdev_spec_a>rfx2400</rx_usrp_subdev_spec_a>
    <rx_usrp_subdev_spec_b>rfx400</rx_usrp_subdev_spec_b>
    <fusb_block_size>0</fusb_block_size>
    <fusb_nblocks>0</fusb_nblocks>
    <tx_usrp_pga_gain_scaling>1.0</tx_usrp_pga_gain_scaling>
    <rx_usrp_pga_gain_scaling>0.5</rx_usrp_pga_gain_scaling>
    <tx_usrp_interp>None</tx_usrp_interp>
    <rx_usrp_decim>None</rx_usrp_decim>
    <tx_samples_per_symbol>4</tx_samples_per_symbol>
    <rx_samples_per_symbol>4</rx_samples_per_symbol>
    <control> ... </control>
  </Radio>
  <NET>
    <interface>gr0</interface>
    <IP>192.168.200.2</IP>
  </NET>
  <MAC>
    <tun_device>
      <tun_device_filename>/dev/net/tun</tun_device_filename>
      <IFF_TUN>0x0001</IFF_TUN>
      <IFF_TAP>0x0002</IFF_TAP>
      <IFF_NO_PI>0x1000</IFF_NO_PI>
      <IFF_ONE_QUEUE>0x2000</IFF_ONE_QUEUE>
    </tun_device>
  </MAC>
  <SRC>
    <digital>TUNTAP</digital>
    <format> ... </format>
  </SRC>
  <SINK> ... </SINK>
</radio_platform>
```

Figure 7. Partial radio configuration XML to support the waveform shown in Figure 6

5. PLATFORM INDEPENDENT INTERFACE

To enable the platform independence of the CE-radio interface in calling the radio API, the radio's behavior can be modeled as a set of hierarchical finite-state-machines. The control API consists of both the choice of the session algorithm and its related settings. These session algorithms are determined by both PHY and MAC settings given by the waveform profile XML.

Generally speaking, a radio is running multi-threading processing for the needed waveform / link session, different waveform parameters at different communication layers adjust the corresponding level of working threads in the radio, such as digital signal processing threads at PHY layer versus packet header lookup threads at network layer; and different algorithm settings trigger different level of events along with these working threads, such as RF emission switching versus packet switching. The CE only needs to tell the radio what PHY and MAC specifications to deploy through the profiles that inform both the radio configuration and control APIs; and the radio will initialize specific processing threads to create a link session such as transmitting a group of packets or broadcast several seconds of music with the specified waveform.

It is important to point out that in our platform independent radio interface, the management of configuration and control APIs are through independent threads. This greatly enhances the flexibility for CE to instruct the radio to carry one set of settings without affecting the other. In other words, waveform settings can be reconfigured in real time while the service session keeps running, therefore, and on-line reconfigurability is supported between PHY and MAC by this interface. Such multi-threading design is implemented and integrated with our fully-functional software radio system, called waveform framework, connected with a radio frequency (RF) interface board, called universal software radio peripheral (USRP) [11], to form a software defined radio platform. It supports real-time reconfigurability across PHY, MAC and network layers [5].

By putting everything together, a complete platform independent radio interface is integrated, shown in Figure 8. The CE controls the radio to conduct radio environment sensing and analyze collected signal and channel data for environment recognition. The resulting radio environment profile is described in the radio environment XML and passed to CE. The radio platform knowledge profile is loaded to CE by the radio definition XML, while the radio's status and performance are reported to CE via the performance XML. The CE makes the solution with the collected knowledge and user's performance requirements, and then formulates the solution in both the waveform profile and the radio configuration profile passed to the radio platform via two corresponding XML scripts. Both of them are interpreted down to specifications of radio APIs. Such interpretation is implemented by an XML parsing module in the interface [12].

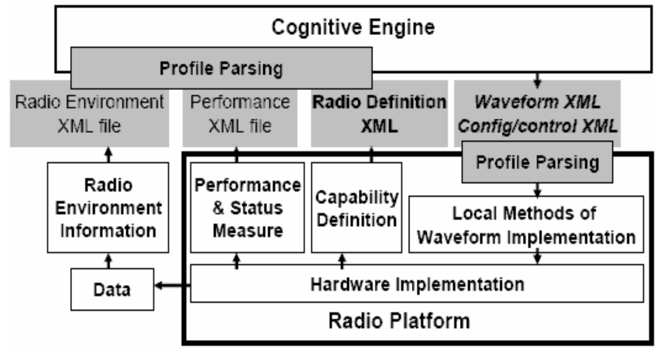

Figure 8. Platform independent CE-radio interface

6. SUMMARY

In this paper we present a general radio interface that is responsible for delivering platform knowledge to the CE, and carrying configuration and control information to instruct the radio specific operations. The radio platform knowledge is representation as a functional resource profile which shields the implementation details. Thus it can be used to represent different radio hardware. The CE generated solution consists of standard waveform and radio configuration profiles that are interpreted into radio configuration and control APIs. Following UML modeling, processing resources and solution components are described in XML meta-data format. This CE-radio interface serves as the bridge between cognitive wireless applications and radio engineering. It also largely defines the CR overall system hierarchy.

This work is supported by NIJ Award No. 2005-IJ-CX-K017 and NSF Grant No. 9983463, DGE-9987586, and CNS-0519959.

7. REFERENCES

[1] B. Le and C. W. Bostian, "Radio Domain Cognition - a System Approach in Theory and Implementation " *Virginia Tech IP (VTIP) No. 05-077,* Oct. 31 2005.

[2] B. A. Fette, *Cognitive Radio Technology*: Newnes, 2006.

[3] J. Mitola, "Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio," Royal Institute of Technology (KTH), 2000.

[4] E. Hohveldin, "Communication profiles," in *Software Defined Radio: Architectures, Systems and Functions*, M. Dilinger, Ed.: Hohn Wiley & Son Ltd, 2003.

[5] B. Le, "Building a Cognitive Radio - From Architecture Defintion to Prototype Implementation," *PhD Disseration, Electrical and Computer Engineering Dept. Virginia Tech* June 2007

[6] J. Wiles and J. Hallinan, *Computational Intelligence*: Whiley-Interscience, 2003.

[7] W3C: World Wide Web Consortium XML section: http://www.w3.org/XML/

[8] OMG: UML homepage: http://www.omg.org/gettingstarted/what_is_uml.htm.

[9] SDRF, "Hardware abstraction layer workgroup report on results of request for information," *SDRF RFI doc: SDRF-04-A-0009-V0.00,* July 17 2004.

[10] B. Le, F. A. G. Rodriguez, Q. Chen, B. P. Li, F. Ge, M. ElNainay, T. W. Rondeau, and C. W. Bostian, "Public safety cognitive radio node system," in *Software Defined Radio Forum Tech. Conf.* , Denver, CO, 2007.

[11] USRP: Ettus Research LLC. homepage: http://www.ettus.com/.

[12] D. Scaperoth, B. Le, T. W. Rondeau, D. Maldonado, and C. W. Bostian, "Cognitive radio platform development for interoperability," in *Military Communications Conference (MILCOM)* Washington D. C., 2006.

[13] B. Le, T. W. Rondeau, and C. W. Bostian, "Cognitive radio realities," *Cognitive Radio special issue, Journal on Wireless Communications and Mobile Computing* to appear fall 2007.