DESCRIBING RADIO HARDWARE AND SOFTWARE USING OWL-DL FOR SOFTWARE DOWNLOAD AND CERTIFICATION

L. Stanchev
Dept. of Computer Science, Indiana University Purdue University Fort Wayne, IN, USA

T. Cooklev & D. Clendenen
Dept. of Engineering, Indiana University Purdue University Fort Wayne, IN, USA

ABSTRACT

Recently, several researchers have discovered the need for radios to use description techniques. Previous research describes information such as the current frequency band, waveform, and so on. However, this information is presented at a level that is not sufficient to determine software/hardware compatibility for over-the-air software download. For example, a device should not attempt to download a wideband waveform if its radio front-end is only narrowband, or if its baseband hardware cannot provide the required MIPS for the new waveform. Over-the-air software download is one of the most interesting features of software-defined radios. The compatibility between software and hardware prior to software download previously had to be verified manually. The approach that is described here removes the need for man-in-the-loop. It uses OWL-DL to describe the components of a software defined radio and the software to be downloaded. As a result, the problem of compatibility is reduced to that of checking a subsumption constraint in an OWL-DL ontology. We show the variability of our approach through examples.

1. INTRODUCTION

Figure 1 shows generic computer hardware/software architecture. It is characterized by three “layers”. The first layer is composed of hardware. The second layer is system software, which provides an interface between the hardware and the third layer of application software. The purpose of the system software is to isolate the hardware from the software. Software-defined radio (SDR) systems have a similar architecture - see Figure 2. The SDR architecture supports multiple baseband standards (a.k.a. waveforms) and each baseband standard can have a different controller standard.

There are two main differences between the architectures in Figures 1 and 2. The first difference is that the hardware in SDR cannot be made fully independent of the software by the system software. In other words, the hardware may not support all waveforms, which are software-defined. The second difference is that the SDR architecture has a special architectural component. It is the *switcher* (or the reasoner – see [1]) and it monitors functional requests. The switcher provides the control functions that must exist outside of the waveforms that are supported by the SDR.

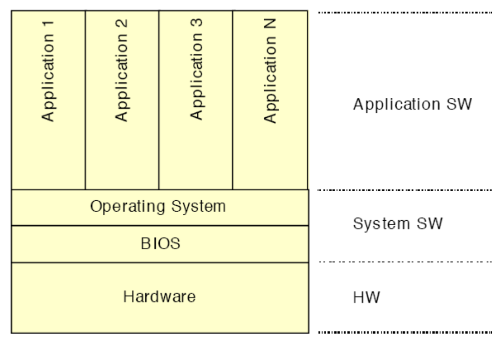

Figure 1. Generic Computer Architecture

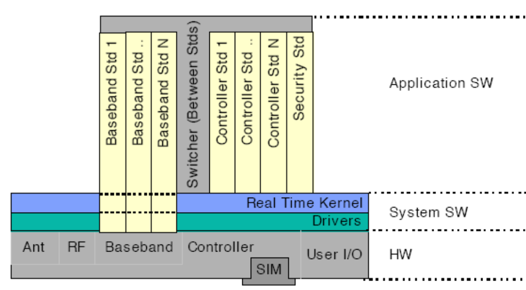

Figure 2. Generic SDR Architecture

The switcher decides which waveform will be active at any one time and switches between waveforms. In a sense, it is the glue that holds the waveform software together.

Consider the scenario where a user wants to download additional baseband software on a SDR device ([2]). This software may or may not be compatible with the device. For example, if the device supports only the Bluetooth air interface standard (AIS), then it will not be able to operate software that requires Wi-Fi capabilities. The general problem is that of determining whether the communication capabilities of a SDR device are compatible with the requirements of the software that can be potentially downloaded on the device. Solving this problem will allow the user to automatically determine whether the software that they want to download can be used on their SDR device. It may be the case that the user has access to a plethora of software products and knowing which ones are compatible with their device is critical.

The problem of deciding whether the capabilities of the software that is candidate for downloading match with the capabilities of a SDR devise is not trivial. The reason is that the capabilities of different SDR devices and communication software are very diverse. Different vendors produce products with distinct characteristics and capabilities and therefore it is difficult to describe all products in a standardized way. For example, it is not enough to denote that a SDR device has two antennas and two RF chains; we also need to describe the type and properties of the antennas and the properties of the RF chains. For example, relational database is not rich enough to store such information.

Existing solutions list the minimal requirements for the software that is available for download. The user needs to manually check whether their SDR device meets these requirements before downloading it. However, if the number of potential software products to download is big, then doing this check manually will be unfeasible. Moreover, a list of the minimal requirements presents a very coarse-grained picture of the software.

In the paper we propose that both the SDR devices and the communication software for them be described as OWL-DL knowledgebases ([3]) over the same ontology. Then the problem of deciding whether the communication capabilities of a software product are compatible with those of a SDR device will be reduced to checking for satisfiability in an OWL-DL knowledgebase. Although the later problem has higher than polynomial complexity, there exist commercial implantations (e.g., Racer [4] and FaCT++ [5]) that perform the satisfiability check in satisfactory time.

The organization of this paper is as described next. In Section 2 we present related research and provide justification for pursuing an ontology-based solution to the problem of matching SDR devices with compatible software. In Section 3 we describe the software download and certification problem in greater details. Sections 4 and 5 present review of the radio architecture and the Manchester OWL Syntax, respectively. Section 6 shows an example of how an OWL-DL reasoner can be used to determine if there are conflicts between the hardware of a SDR device and the software that is candidate for downloading. Chapter 7 summarizes our approach and outlines its limitations.

2. MOTIVATION AND RELATED RESEARCH

OWL, which stands for Web Ontology Language, is powerful enough to describe a domain, where a domain consists of individuals that belong to classes. OWL can be used to describe the data and object properties of the individuals and the relationship between the individuals and classes in the domain. There are three dialects of OWL: OWL-Lite, OWL-DL, and OWL-Full. OWL-DL is the most popular dialect because it provides significant expressive power and supports practical reasoning algorithms.

In the radio world, the individuals in the knowledgebase are devices and components, such as RF sections, analog-to-digital and digital-to-analog converters (ADC and DAC, respectively), digital hardware, filters, and mixers. The constraints of the components and subcomponents will include the capabilities of antennas, filters, ADC/DAC, and digital hardware, to name a few. Such descriptions are extensible and allow all components and subcomponents to be described in sufficient depth. Furthermore, the technique can be applied beyond devices to network and service capabilities.

An ontology is useful when there are complex interactions between the individuals in a domain. In such cases database technology is inadequate. In a world with low levels of volatility with a small numbers of radio types, modes of operation, end-user services, and simple, fixed economic relationships among service providers, the construction of formal descriptions is not required. In this case the necessary information can be stored in centralized databases. This was the world of yesterday and, to a large extent, today. However, the wireless universe is fast becoming as complex as the Internet, with countless waveforms, implementations, and capabilities. In the future, as radio devices switch between air interface standards, services, networks, and operators, they will need to change their configuration on the fly. A data model that can represent complex relationships between individuals, such as an OWL-DL ontology, is best suited for describing such a complex system.

The OWL-DL language can be used to describe the current configuration of a radio, its potential configuration and functionality, the characteristics of waveforms and air interface standards (AIS), the type of information being handled, the environment (spectral, physical/geographic, and in some cases situational, such as whether special emergency conditions exist) and the type of end users involved.

The need for description techniques for identifying the various objects in the wireless universe, their configurations and capabilities, and the services that the users are requesting has been noted in several publications [6-10].

These techniques are collectively called Networking Description Language (NDL), Metalanguage, or Modeling Language for Mobility (MLM). We use the term metalanguage for all of these description techniques. The paper [9] introduces an ontology that describes waveforms and digital modulation parameters, such as bits and symbols. The ontology described here is at a different level of granularity. Furthermore the work in [9] does not capture the hardware architecture of a radio device.

The research that is presented is related to the Software Communications Architecture (SCA) of JPEO (Joint Program Executive Office). The SCA can be considered as one particular implementation of the architecture in Figure 2. It includes an operating environment (OE). The OE consists of an operating system (OS), CORBA middleware (including the OMG-defined Event and Naming Services), and the elements defined by the Framework Control and Service Interfaces. In SDR, the role of the SCA is to provide a common infrastructure for managing the software and hardware elements and ensuring that their requirements and capabilities are commensurate. Additionally, the SCA ensures that once software components are deployed on a system, they are able to execute and communicate with the other hardware and software elements present in the system. The SCA accomplishes these tasks by defining a set of interfaces that isolate the applications from the hardware.

All SCA compliant systems require certain software components to be present in order to provide for component deployment, management, and interconnection. These components include the *DomainManager* (including support for the *ApplicationFactory* and *Application* interfaces), *DeviceManager*, *FileManager*, and *FileSystem* interfaces. The SCA defines a set of files that are referred to as the Domain Profile. The SCA Domain Profile elements identify the capabilities, properties, inter-dependencies, and location of the hardware devices and software components that make up an SCA-compliant system.

The Domain Profile is a hierarchical collection of eXtensible Markup Language (XML) files that define the properties of all software components in the system. XML is a description language that provides explicit structure for the description of information. In the SCA, XML files describe the layout of the system and the waveform applications, their location, names, and so on.

This work extends previous research by developing ontology-based descriptions to support software download and certification. In previous research, solutions to the problem of checking for conflicts between hardware capabilities and software specifications have not been proposed.

3. SOFTWARE DOWNLOAD AND CERTIFICATION FOR SDR DEVICES

The need for software download in a SDR device arises naturally, just as the need for software download in a computer. Consider a scenario where a user requests a service from a device. Service requests can come from the infrastructure or from the radio itself in response to changes in environmental conditions. If the requested service is within the handset's currently configured capabilities, the service is initiated. If not, the switcher searches its local repository for software code modules that will allow satisfaction of the request. If such software is found within the device, then the switcher installs it. If not, the switcher asks the wireless network infrastructure if it can provide the software for the requested service. Assuming that the answer is yes, the network can provide the code. Then one must determine that there is no conflict and that the software can actually be run on the hardware.

Previously, the assumption was that this step could be done manually. Clearly, this is inadequate. In this paper we describe one implementation where this can be done automatically. If the software is compatible with the hardware, then the switcher can check the software to determine that the software is from a trusted source before the device can finally proceed to actually perform the software download. Before the software is actually run on the device, additional operations must take place. These additional operations ensure that all resources required by the waveform software are available. In tactical radios these tasks are done by the SCA.

SDR technology presents many challenges, such as technical, regulatory, and business. The regulatory and certification problems are some of the most significant since it is not possible to test every software module on every hardware platform. Considering this, it is very difficult to ensure that every software module will behave appropriately on every hardware module. If this problem is solved, it is likely that a very significant industry of third-party software vendors will appear. In other words, users will be able to download software onto their wireless devices from any software vendor, similar to the computer software industry today. Therefore the problem is of very high significance and no satisfactory solution exists at present.

The paper proposes that a third-party software vendor can create metalanguage description for software. Then, it can present the software together with the description to a known certification lab. The certification lab checks the software to determine if the description is accurate and adequate considering the possible descriptions of radio hardware. The lab may suggest modifications to the metalanguage description to ensure that the software will be run only on hardware devices with which all relevant government regulations are satisfied. If the lab finds that the vendor's description is accurate and adequate, it certifies, or approves, the metalanguage description. It can be noted that the certification of the software will almost certainly include other steps, such as verifying conformance with a standard. These steps are well known and are not examined in detail here. We propose using OWL-DL for describing radio software and radio hardware. The intent of this work is to ensure that the behavior of the software/hardware combinations is predictable as long as the descriptions match.

4. DESCRIPTION OF RADIO HARDWARE

Since we will be describing radio hardware, it is important to review radio architecture. Most wireless communication systems employ architecture with analog components, digital components, and ADC/DAC between. The analog components include an antenna system and associated front-end amplifiers, switches, filters, down-converters and up-converters. The baseband digital signal processing algorithm as implemented on digital hardware is shown in Figure 3.

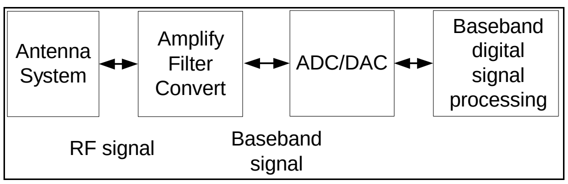

Figure 3. Wireless Transceiver Architecture

The basic functions of the transceiver are down/up conversion, channel selection, interference rejection and amplification.

Down conversion is required for receivers. A receiver subsystem takes the weak signal from the antenna, converts the signal from the transmission radio frequency (high - RF) to baseband frequency (low – typically low end of the desired signal will approach zero Hertz), filters out the noise (from external sources out of band / in band, and internally generated sources) and unwanted channels, amplifies the signal to a level that can be used efficiently by the rest of the system and delivers the signal to the baseband subsystems.

Up conversion is required for transmitters. A transmitter subsystem takes the signal (much stronger than the received signal at the antenna, but much lower power than the signal to be transmitted) from the baseband subsystem, converts the signal up from baseband frequency to the desired transmission radio frequency, amplifies the signal to the desired transmission level, filters out any noise introduced in the process (sometimes referred to as spurious emissions) and delivers the signal to the antenna.

In this work we focus on the main parameters of radio hardware. These main parameters include the following:

- Number of antennas, center frequency, and bandwidth of the antenna. Both the center frequency and the bandwidth can be specified in terms of the ranges in which they can be tuned.
- The number of antennas may be different from the number of down/up conversion chains. The antenna(s) may be connected to a switch or an analog front-end block; in general there might be 0 or more switches.
- One or more receiver analog down-conversion blocks are specified by bandwidth and receiver sensitivity.
- One or more transmit analog front-end blocks with parameters that include bandwidth, center frequency, and third-order intercept point.
- One or more ADC with parameters including SNR and sampling frequency. The ADC could be placed at IF or at baseband level. If the ADC is placed at the IF level, there usually will be a digital down-conversion block.
- The transmitter will include one or more DACs, specified in terms of the spurious-free dynamic range (SFRD).
- One or more digital hardware modules each described using MIPS and memory. Both MIPS and memory are usually specified in terms of upper limit.

5. THE MANCHESTER OWL SYNTAX

This section presents a few examples of the Manchester OWL syntax, where the reader should refer to [11] for complete overview. Our later examples will use this syntax.

The objects in the domain are referred to as *individuals*. An example individual `antenna1` can be defined as shown below.

`Antenna` **and** *center_frequency_tuning_range* **value** [>=50, <=6000] **and** *current_center_frequency* **value** 2400 **and** ((*connected_to* **value** `rx_1`) **or** (*connected_to* **value** `rx_2`) **or** (*connected_to* **value** `tx_1`) **or** (*connected_to* **value** `tx_2`)) **and** *connected_to* **exactly** 1 `Transceiver`

It describes an antenna with center frequency tuning range between 50 MHz and 6000 MHz and current center frequency of 2400 MHz that is connected to exactly one of four transceivers (rx_1 and rx_2 are receivers, while tx_1 and tx_2 are transmitters). At any point, a switch will connect exactly one of the transceivers to the antenna. Note that rx_1, rx_2, tx_1 and tx_2 are also example of individuals (all individuals will start with a small letter in the paper).

Every individual can have both *data properties* and *object properties*. A data property has a value that is a primitive type (e.g. integer, float, date, etc.), while an object property has a value that is another individual. In our example, *current_center_frequency* is a data property, while *connected_to* is an object property.

Every individual is of a certain type, which corresponds to the class it belongs to. For example, all receivers belong to the class `Receiver`. A class can be defined using other classes. For example, the class `Transceiver` is defined as "`Receiver` **or** `Transmitter`". A restriction can be specified on a class. For example, we can define that the class `Receiver` is subsumed by the class with description "*connected_to* **exactly** 1 `Analog_Digital_Converter`". We will denote this subsumption as follows.

`Receiver` **SubClassOf** *connected_to* **exactly** 1 `Analog_Digital_Converter`

Another example of a subsumption is shown below.

`IEEE_802.11a` **SubClassOf** *contains* **min** 1 (`Antenna_System` and *connected_to* **min** 1 (*Receiver* and (*connected_to* **min** 1 (`Analog_Digital_Converter` and *number_of_bits* **some int** [>=6]))))

It denotes that the software standard 802.11a requires the devices that support this standard to contain at least one antenna systems that is connected to at least one receiver that is connected to at least one ADC with number of bits greater or equal to 6.

6. EXAMPLE DESCRIPTION OF SDR DEVICE AND CHECKING FOR CONFLICTS

6.1. Example Description of SDR device

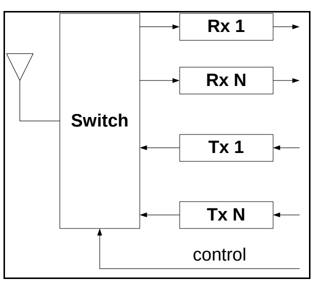

Figure 4. Example SDR Architecture

Consider a device that has the architecture shown in Figure 4. From this device, we will create the individuals shown in Figure 5. We will denote that the device `device1` contains the antenna `antenna1`, the receivers `rx_1` and `rx_2`, the transmitters `tx_1` and `tx_2`, the digital analog converters `DAC1` and `DAC2`, and analog digital converters `ADC1` and `ADC2` and the digital signal processing module `DSPM1` as shown below.

`device1` **SubClassOf** (*contains* `antenna1`) **and** (*contains* `ADC1`) **and** (*contains* `ADC2`) **and** (*contains* `DAC1`) **and** (*contains* `DAC2`) **and** (*contains* `rx_1`) **and** (*contains* `rx_2`)

The fact that an antenna can be connected to exactly one of the four transceivers will be denoted as follows.

`Antenna_System` **SubClassOf** (*connected_to* **value** `rx_1`) **or** (*connected_to* **value** `rx_2`) **or** (*connected_to* **value** `tx_1`) **or** (*connected_to* **value** `tx_2`)

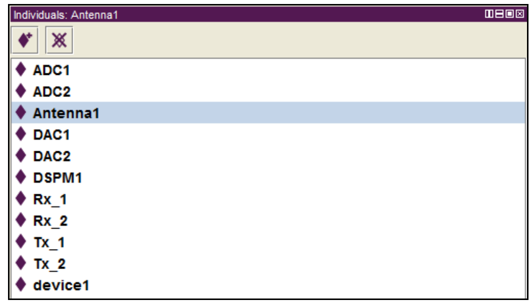

Figure 5. Individuals for the Example Architecture

6.2. Checking for Conflicts

As an example, consider the description of the air interface standard IEEE 802.11a ([12-13]). IEEE 802.11a operates in the so-called UNII frequency bands over 5 GHz and one channel occupies 20 MHz. It requires one antenna and one RF chain. 802.11a is a time-division duplexing (TDD) system – a device cannot transmit and receive at the same time. Devices transmit and receive at the same frequency at different time instants. The center frequency must be greater than 5.15 GHz and smaller than 5.825 GHz. The standard requires receiver sensitivity of at most -82 dBm, and SNR of at least 18 dB. It achieves data rates between 6 Mb/s and 54 Mb/s. The lowest data rate should be achieved for SNR values of 18 dB or less and the highest data rate should be achieved for SNR values of 35 dB or less. Since the SNR of a converter is about 6 dB per bit, we assume that at least 6 bits of resolution are required from the ADC to achieve the required SNR. We assume that the digital hardware generally must provide at least 9000 MIPS to implement the standard. The implementation of 802.11a will require the device to able to perform 64-point FFTs and decoding of a convolutional code with constraint length 7 within a certain period of time. Here, it is assumed that any digital hardware offering 9000 MIPS can do this, otherwise a more detailed description may be required. In general, this depends on the software implementation.

Part of the description of the IEEE 802.11a standard is shown below.

`IEEE802.11a` **SubClassOf** *contains min* 1 (`Antenna_System` **and** (*current_bandwidth* **some int**[>=20]) **and** *connected_to* **min** 1 (`Receiver` and (*connected_to* **min** 1 (`Analog_Digital_Converter` and (*number_of_bits* **some int**[>=6]) and *connected_to* **min** 1 (*Digital_Signal_Processing_Module* **and** *MIPS* **some int**[>=9000])))))

This describes that the device must contain minimum one antenna, the bandwidth of the antenna must be at least 20 MHz, the antenna must be connected to at least one receiver, which in turn must be connected to at least one ADC with number of bits at least 6 and connected DSPM with MIPS at least 9000. Note that measurement units are not part of the current OWL-DL syntax.

We can check if our device `device1` belongs to the class IEEE802.11. If this is the case, then our device supports the protocol. If this is not the case, then either our device does not support the protocol or we do not have enough evidence to conclude that the device supports the protocol.

Another example is the latest IEEE 802.11n standard. This standard defines operation for up to 4 antennas. However, operation with two antennas is mandatory; operation with four antennas is optional. Operation with 20/40 MHz channels is similar: support for 20 MHz channels is mandatory and support for 40 MHz channels is optional. Note that the transmission spectral masks for 20 MHz and 40 MHz operation are different.

Having mandatory and optional features is typical for waveforms. Transmit beamforming and space-time block coding (STBC) are other optional features. As far as the ability to download some software onto a certain radio hardware platform, the mandatory features are more important. For example, a device that supports two spatial streams in 20 MHz channels should be considered compatible with 802.11n and should be allowed to download those portions of the 802.11n software that it can run. This means that the code may have to be structured so that devices can download only those parts with which they are compatible. No other description technique allows this at present.

Another important parameter that determines which software can be run on a given devices is the processing power of the digital hardware. A minimum MIPS is required to implement the mandatory portions of the software and even more MIPS are requires for the optional features.

7. CONCLUSIONS

We propose using OWL-DL to describe radio software and hardware. We presented an ontology that can be used prior to over-the-air software downloads onto radio hardware platforms to check for compatibility. While we consider software download of waveform software, the technique can be applied for all types of software that radio devices can download.

One characteristic of our approach is that both the capabilities of the SDR device and the communication software need to be described by a domain expert. Although this task is time-consuming, it needs to be performed only once. We believe that this work is significantly less than manually checking for the compatibility of every possible SDR device – communication software pair.

Another limitation of our approach is that both the capabilities of the SDR device and the communication software need to be described using concept descriptions over the same ontology. In other words, in order for the compatibility test to be reliable, all SDR devices and communication software must reference the same standardized ontology.

The running example ontology of this paper can be downloaded from: http://stanchev.ipfw.edu/~lubo/SDROntologyv4.3.owl

8. REFERENCES

[1] B. Fette (Ed.), “Cognitive radio technology,” Newnes Publication, August 2006.

[2] J. Polson, E. Christensen, B. Tarver, S. Gifford, “Common software download requirements for software-defined radios,” SDR Forum Technical Conference, 2002.

[3] Web Ontology Language (OWL), http://www.w3.org/2004/OWL

[4] Volker Haarslev and Ralf Moller, Racer: A Core Inference Engine for the Semantic Web, *Second International Workshop on Evaluation of Ontology-based Tools*, 2003

[5] D. Tsarkov and I. Horrocks, FaCT++ Description Logic Reasoner: System Description, *Lecture Notes in Computer Science*, volume 4130, pages 292-297, 2006

[6] T. Cooklev, M. Cummings, “Networking Description Language for Ubiquitous Cognitive Networking”, SDR Forum Technical Conference, Washington, DC, 2008.

[7] Mark Cummings; Todor Cooklev; Bryan Lyles; P. A, Subrahmanyam; "Commercial Wireless Metalanguage Scenario", SDR Technical Conference, Denver, Co. Nov. 2007.

[8] Mieczyslaw M. Kokar, Donald Hillman, Shujun Li, Bruce Fette, Preston Marshall, Mark Cummings, Todd Martin, John Strassner, “Towards a Unified Policy Language for Future Communication Networks: A Process,”

[9] S. Li, M. Kokar, “Developing an ontology for the cognitive radio, SDR Forum Technical Conf., Washington, DC, 2009.

[10] M. Kokar, L. Lechowicz, “Language Issues for Cognitive Radio,” Proceedings of the IEEE, vol. 97, April 2009.

[11] Matthew Horridge, Nicholas Drummond, John Goodwin, Alan Rector, Robert Stevens and Hai Wang, The Manchester OWL syntax, *OWL: Experiences and Directions*, 2006

[12] Michael J. Meeuwsen, Omar Sattari, Bevan M. Baas, A Full-rate software implementation of an IEEE 802.11A compliant digital baseband transmitter, *Proceedings of the IEEE Workshop on Signal Processing Systems, 2004.*

[13] Byung Wook Lee, Sung Ho Cho, “An implementation of the SDR baseband platform for OFDM communication systems,” SDR Forum Tech. Conf., 2005.