DEVELOPING AN ONTOLOGY FOR THE COGNITIVE RADIO: ISSUES AND DECISION

Shujun Li, Mieczyslaw M. Kokar, David Brady
Department Electrical and Computer Engineering, Northeastern University, Boston, MA, USA
shli@ece.neu.edu, mkokar@neu.edu, Brady@neu.edu

ABSTRACT

An ontology defines the basic terms in a domain and the relationships among them. It is used to share information among people, machines, or both in order to facilitate further analysis of the domain knowledge. In the cognitive radio domain, two radios can achieve interoperability by exchanging the knowledge about their communication parameters and protocols. The knowledge, which includes information like the capabilities, configuration and system state of the radio, can be used to reconfigure the radios in a flexible way. Although there is some research on developing ontologies in the cognitive radio domain, the language used to model the ontology, e.g. UML, is not sufficiently machine-understandable since UML does not have formal, computer-processable semantics. In this paper, we describe our efforts to develop an ontology for the cognitive radio domain, specifically, for the lower layers (physical layer, data link layer and the network layer), using a formal declarative language. This ontology will provide extensible standard vocabularies, which will server as the basis for rule-based inference and constraint solving capabilities. Since there are many alternative ways to model the knowledge in a domain, and since the criteria for choosing the best solution are application-dependent, we considered and compared different approaches to conceptualize the knowledge of the cognitive radio domain based on various criteria, like the extent of the coverage of the knowledge of the domain, the ability of inferring facts that are not explicit in the knowledge representation and the extendibility of the ontology. This paper summarizes some of the results of our analysis.

1. INTRODUCTION

In philosophy, ontology is the study of the nature of being or existence. The concept of ontology can be further extended to artificial intelligence, computer science and information science. Generally, it refers to a formal, explicit specification of a set of concepts in a specific domain and the relationship between these concepts [1]. The term "formal" means that the ontology is machine processable for the purpose of knowledge reuse and sharing.

Ontology plays an important role in cognitive radio in that it provides the foundation of self-awareness, interoperability and reasoning ability [2]. Firstly, knowledge in the radio is sometimes embedded in the software rather than explicitly expressed, e.g. the length of the training sequence and the parameters of the modulation scheme. Besides, the knowledge base of a node involves state information that is changed in run-time, e.g. SNR and the estimated value of the channel parameters. The use of ontology can enable the radios to understand their structure and modify their functioning at run-time to optimize their performance. Secondly, the use of ontology makes it possible to exchange information between radio agents across different organizations, providing a shared understanding of common domain. In some cases, the optimization is not only based on the local parameters but also on the parameters of the channel and other radios in the network. Hence, interoperability can enable multi-criteria optimization on the network-level. Thirdly, ontology differs from database for its inference and reasoning ability. Furthermore, additional facts can be deduced when an ontology is combined with rules or policies. A rule is an if-then statement, i.e. if a hypothesis is true, then the conclusion holds. Policy is a set of rules that can be either external policy such as the frequency bands at specific location authorized by FCC or internal policy for performance optimization. In this way, the behavior of a radio is controlled by some policies rather than device-specific software, resulting in a more flexible mechanism.

The approach to design an ontology varies depending on the goal and the knowledge domain. A common approach is bottom-up, in which the ontology starts with the most specific concepts and then these concepts are grouped into several categories on a more abstract level [3]. Since our ontology covers the basic concepts in the cognitive radio domain, it is easier to use the bottom-up approach, starting from the most specific concepts such as transceiver, channel and protocol. However, the use of purely bottom-up approach would require classification, i.e., the concepts at the lowest level would have to be classified into a hierarchy (a lattice). The result then might be incompatible with any existing top-level ontologies known in the literature. To avoid such a situation, we decided to use a hybrid ("sandwich") approach in which we introduce concepts primarily in a bottom-up fashion, yet we use an upper ontology as for guidance.

Basically, an ontology can be evaluated in terms of (1) coverage of knowledge, (2) inference ability and (3) extendibility. The ontology presented in this paper aims to provide the vocabularies to represent a specific waveform or Air Interface Specification (AIS) in the cognitive radio domain, with focus on the physical layer, data link layer and network layer. Extendibility refers to the ability to add classes, properties and constraints to an ontology without violating its consistency. An example will be given in Section 3.6 to discuss this issue. In this paper, we are not going to discuss the inference ability of the ontology. This issue will be explored in the future.

This paper is organized as follows: Section 2 describes the division of the top-level classes and properties. Section 3 will move on to the issues and decisions that arise during the process of designing the internal concept structure. We will cover the issues such as how to classify the relationship between signal, symbol and sample, how to represent the input and output of a radio component, etc. Conclusions and future work are given in Section 4.

2. TOP-LEVEL DIVISION

An upper ontology defines the most general concepts that are the same across different domains. Choosing an appropriate upper ontology as a reference model is beneficial to the top-level classes classification.

Among the well-know upper level ontologies, we chose DOLCE, the Descriptive Ontology for Linguistic and Cognitive Engineering [4], as our reference model. The fundamentals of DOLCE are the distinction between Endurant, Perdurant and Quality. Endurant, also known as Object in our ontology, refers to the entity that is wholly presented at any given snapshot of time. Examples include material objects such as a piece of paper or an apple, and abstract objects such as an organization or a law. Conversely, Perdurant, also known as Process in our ontology, is the entity that is only partly presented at any snapshot of time. A process can have temporal parts or spatial parts. For example, the first movement of a symphony is a temporal part of a symphony, whereas the symphony performed by the left side of the orchestra is a spatial part of a symphony. In both cases, a part of a process is also a process itself. Note that an object cannot be a part of a process, but rather participate in a process, for example, a person is not a part of running, but rather participates in running. In addition, the input and output of a process are objects. For instance, the input of modulation is a signal, where modulation is a process and signal is an object. Qualities, also known as Attributes in our ontology, are the basic entities that can be perceived or measured. Attributes cannot exist on their own; instead they must be associated to either an object or a process. In addition, each attribute has its value and unit of measure. For example, color is an attribute of an object. Its value is red. Data rate is an attribute of a data stream; its value can be 120 and the unit of measure is bit/sec. According to the classification described above, the top-level classes in our ontology are shown in Figure 1, including (1) Object, (2) Process, (3) Attribute, (4) Value, and (5) UnitOfMeasure.

There is also a need regarding the distinction between property and attribute (c.f. the discussion in [5]). An attribute is a feature of an object that is independent of the context that the object is in. For instance, the size of a cup is this cup's attribute. Conversely, the property of an object depends on the context, for example, whether the cup is full or empty depends on the context, thus it should be modeled as a property. However, since the ontology presented in this paper is formalized in OWL (Web Ontology Language) using the Protégé tool, there is no explicit distinction

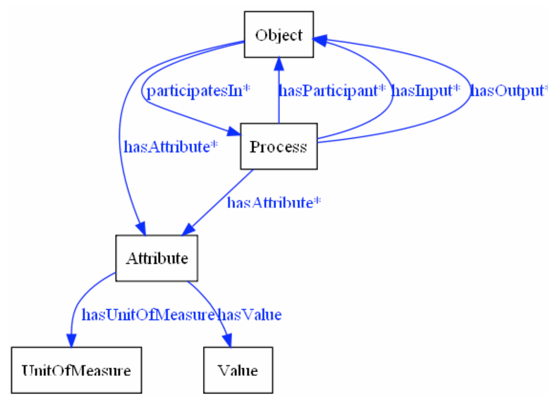

Figure 1 Top-Level Classes

between attribute and property in OWL. Take *packet field* as an example. A packet consists of a sequence of packet fields. The size of a packet field is an attribute of packet field, but whether a packet field is optional or compulsory is a property. OWL only provides two types of properties: (1) object-type property, which links an individual to another individual, and (2) data-type property, which links an individual to an XML Scheme Datatype value (e.g. Integer, Boolean, etc.). If we only use the features provided by OWL, both *packetFieldSize* and *isOptional* should be modeled as datatype properties, i.e. *packetFieldSize* is linked to an integer value whereas *isOptional* is linked to a Boolean value. In addition, OWL does not provide a built-in feature to represent the unit of a value, e.g. to say the *packetFieldSize* of a packet field is 32bits. The Attribute in our ontology only refers to the feature of an object that has a value and a unit and does not depend on context. Other

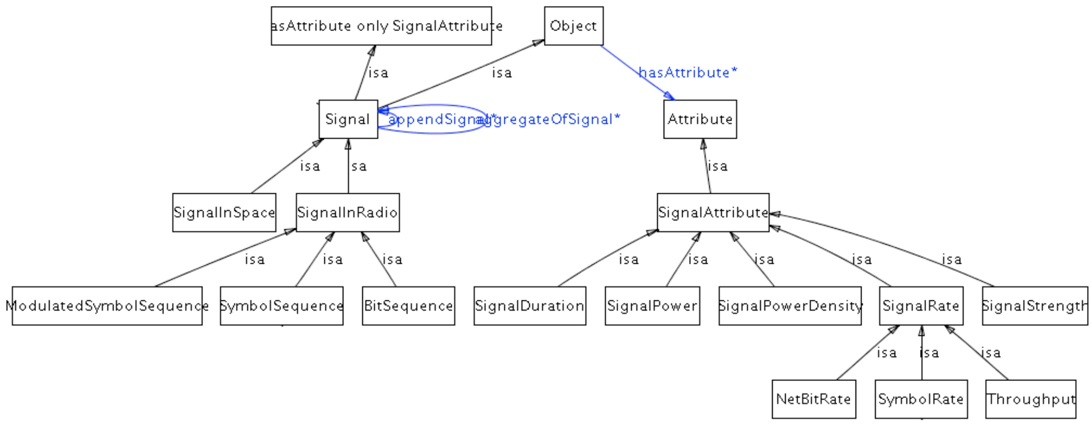

Figure 2 Structures of Signal and Signal Attributes

relations that an object participates in are represented directly as OWL properties.

3. INTERNAL CONCEPT STRUCTURE

3.1 Signal, Symbol and Sample

First of all, let's look into the most important element in a radio -- Signal. Signal is any time-varying or spatial varying quantity. There are different views on the classification of Signal. Signal can be divided into continuous and discrete signal; then further divided into quantized signal and unquantized signal. However, since our ontology is developed for the cognitive radio domain, almost all the signal processings (before the DAC or the amplifier) are implemented in software. Thus, the signal in the radio can be viewed as discrete signal. In our ontology, we divide the Signal class into (1) SignalInRadio and (2) SignalInSpace.

There are three basic properties associated with Signal, shown in Figure 2. First, a part of a signal is also a signal. Second, a signal can be appended to another signal, producing a new signal. Third, a signal consists of a sequence of samples. Besides the above properties, signal duration and signal rate are also the basic attributes for all kinds of signals. In addition, SignalInSpace has attributes such as signal power, power strength and power density. Based on the signal processing in the radio, SignalInRadio can be further divided into (1) BitSequence, (2) SymbolSequence, and (3) ModulatedSymbolSequence. For instance, the modulator in the radio usually groups the incoming data bits into codewords, one for each symbol to be transmitted. Here, the binary data bits are modeled as BitSequence and the codewords are modeled as SymbolSequence. The SymbolSequence will be mapped to the amplitudes of the I and Q signals, and then multiplied by the baseband frequency to produce the ModulatedSymbolSequence. After that, the ModulatedSymbolSeuqence, which is also the output of the Modulator, will be frequency shifted into a passband signal.

Next, we will look into another two classes that are closely related to Signal -- Symbol and Sample. The term Symbol is somewhat ambiguous. On one hand, symbol refers to the physically transmitted signal that is placed on the channel. It is a state of the communication channel that persists for a fixed period of time [6]. For example, in passband transmission a Symbol usually refers to a sine wave tone, whereas in baseband transmission a symbol usually refers to a pulse rather than a sine wave tone. Another example is the concept of chip in spectrum spreading. Spectrum spreading is usually used in CDMA. In this scheme, an information bit is represented by a *chip* sequence. After spectrum spreading is applied, the symbol rate of the physically transmitted signal is actually the chip rate. Thus, if Symbol is defined as described above, then Chip can be modeled as a subclass of Symbol. On the other hand, Symbol may be used at a higher level and refer to one information bit or a block of information bits that will be

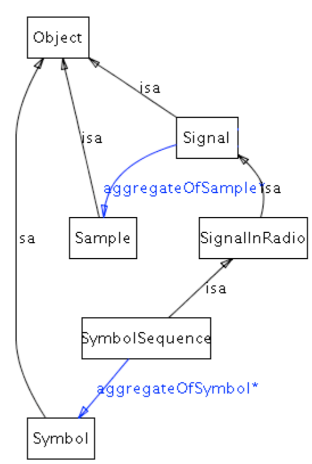

Figure 3 Relationships Among Signal, Symbol and Sample

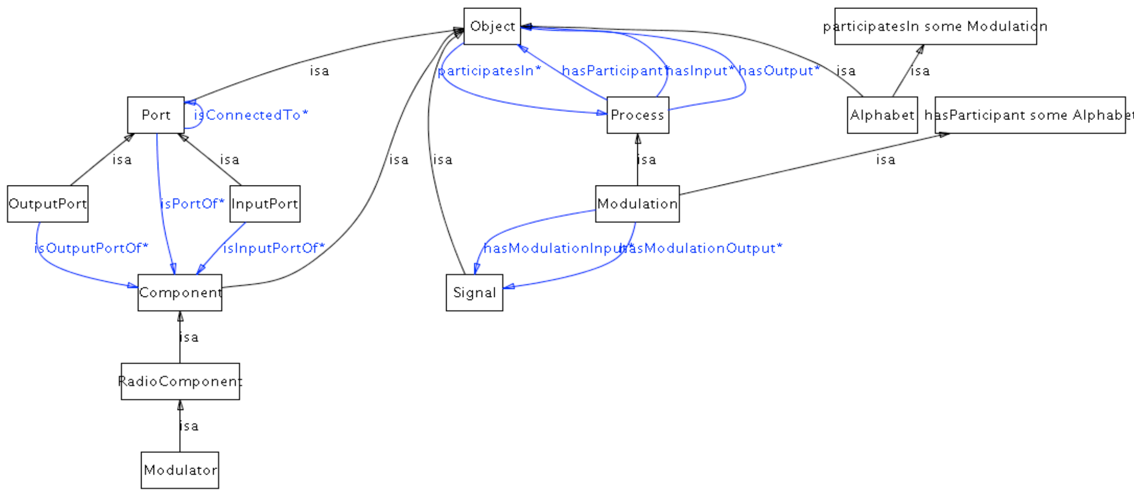

Figure 4 Relationship Among Alphabet, Modulation and Modulator

modulated using a conventional modulation scheme such as QAM [6]. The SymbolSequence described in the preceding paragraph refers to this definition. We now turn to the concept of Sample. Sample refers to a value taken at a point in time or space. Hence, a Signal can be viewed as an aggregation of samples. The relationships among Signal, Symbol and Sample are shown in Figure 3.

3.2 Alphabet, Modulation and Modulator

The relationships among Alphabet, Modulation and Modulator are good examples to show the relationship between Object and Process. Digital modulation is a process that takes digital signal as input and converts it to analog signal. Then the analog signal will be transmitted to the wireless channel. Thus, Modulation is a subclass of Process. The changes in the carrier signal are chosen from a finite number of M alternative symbols, which is called Modulation Alphabet or Alphabet. Alphabet is actually a lookup table that has two columns: index and symbol value.

Modulation alphabet is often represented on a constellation diagram. A constellation diagram represents the possible symbols that may be selected by a given modulation scheme as points in the complex plane. The coordinates of a point on the constellation diagram are the symbol values. If the alphabet consists of $M = 2^N$ alternative symbols, then each symbol represents a message consisting of N bits. The index of each symbol value implies the bit pattern for that particular symbol. For a particular modulation, the content in the alphabet table is unchanged, thus Alphabet is a subclass of Object. On one hand, Alphabet participates in the Modulation process; on the other hand, Modulation can have Alphabet as its participant.

Modulator is an electronic device that performs modulation. In our ontology, Modulator is modeled as a subclass of Component [7]. A Component is a special type of Object, each instance of Component has input and output ports. In addition, a Component can have subcomponent, for instance, a modulator has pulse-shaping filter as its subcomponent. The relationships among Alphabet, Modulator and Modulation are shown in Figure 4.

3.3 Agent

Agent is another special type of Object. The definition of Agent varies in different domains. In artificial intelligence, Agent refers to an autonomous entity which observes and acts upon an environment and directs its activity towards achieving goals [8]. The essence of an agent includes: (1) reaction to the environment, i.e. an agent is able to sense the environment and react properly to the changes of the environment; (2) autonomy, i.e. an agent can perform a task without human intervention; (3) persistency, i.e. if a software program is an agent, then it should be executed continuously over time rather than invoked on demand; (4) goal-directed, i.e. an agent should be capable of choosing among multiple options and select the one that can achieve the goal [9].

The above properties distinguish an agent from an ordinary software program or module. In the domain of cognitive radio, a radio component typically has input and output. It can perform a task on its own by running a predefined algorithm. It could be said that the radio component senses the environment via the input and responds to the environment via output. In this sense, the radio component is capable of reacting to the environment and has some degree of autonomy. However, in order to become an agent, a radio component must have goal-directed behavior, i.e. it does not simply sense and react upon the environment autonomously [9], it must be able to achieve a set of goals, e.g. avoid detection and interference, maximize throughput, etc. In this ontology, we do not restrict any of the radio components as a subclass of Agent. Instead, we define that an Object is an Agent if and only if it has a goal. Given such a necessary and sufficient condition, a radio component can be classified by a reasoner whether or not it is an agent.

3.4 Input and Output of a Component

A component is a self-contained part of a larger entity. It often refers to a manufactured object or a software module. On one hand, a component can be part of a larger component; on the other hand, it can have smaller components as its subcomponents. In addition, a component usually has input and output ports. One component is connected to another component by ports. Port is modeled as a subclass of Object. A port can be connected to another port if the two ports are carrying the same type of signal. For example, if a modulator takes a digital signal as the input and outputs analog signal, then the output port of this modulator can be connected to another port that also carries analog signal. There are two approaches to conceptualize this issue. One way would be linking each port to a particular signal type by an object-type property called *portType*. Two ports can be connected if their *portType* are the same. Another approach is more straightforward. In the preceding section, we discussed the relationship between Modulator and Modulation. Generally, the functionality performed by a radio component corresponds to a process. For instance, Transmitter corresponds to Transmission; SourceEncoder corresponds to SourceCoding, etc. Instead of restricting the portType of a radio component, we can restrict the input and output of its corresponding process to particular types of signal, e.g., the input and output of the modulation process can be linked to digital signal and analog signal respectively. In our ontology, the second approach is adopted, as shown in Figure 5.

3.5 AIS and Protocol

Air Interface Specification (AIS) is closely related to the term Waveform. Based on the definition provided by the P1900.1 document [10], waveform refers to "the set of transformations and protocols applied to information that is transmitted over a channel and the corresponding set of transformations and protocols that convert received signals back to their information content". AIS is the specification of a set of processes that are applied to the transmitted and received information. For instance, if two radios want to communicate with each other, the signals provided by the two radios must both satisfy the AIS, whereas the details of implementation may be different. In this sense, AIS is equivalent to the term Waveform defined above. Since AIS is a specification of a set of processes, AIS can be viewed as a process as well. Hence, in this ontology, AIS is modeled as a subclass of Process. We now turn to the relationship between AIS and protocol. Typically, AIS is layered, with

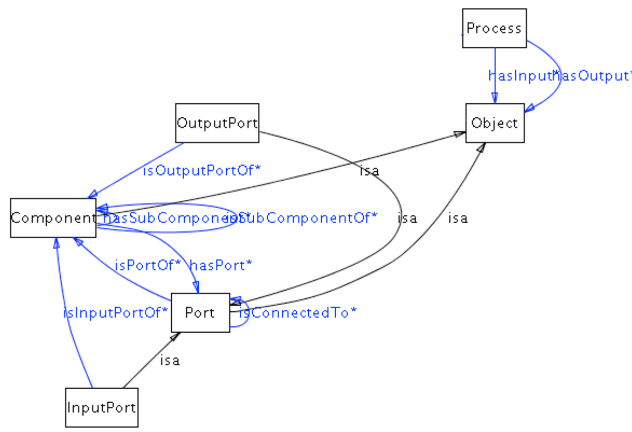

Figure 5 Input and Output of A Component

interfaces defined for each layer. Each layer consists of one or more protocols that perform the layer's functionality. For example, in cdma2000 1xEV-DO [11], the AIS is divided into several layers, such as physical layer, MAC layer, security layer, connection layer and so on. The MAC layer consists of multiple protocols such as Control Channel Protocol and Forward Traffic Channel Protocol. Hence, AIS can be viewed as an aggregation of protocols. From another point of view, AIS is also an aggregation of various processes, i.e. AIS provides the specification for modulation, channel coding, source coding, etc. In our ontology, we only focus on the physical layer, data link layer and network layer of the AIS. The relationships among AIS, Protocol and Process are illustrated in Figure 6.

3.6 Attribute and Extendibility

Each object or process is associated to one or more attributes. Some attributes are shared among different objects or processes. For instance, if a physical transmission medium is divided into several logical sub-channels based

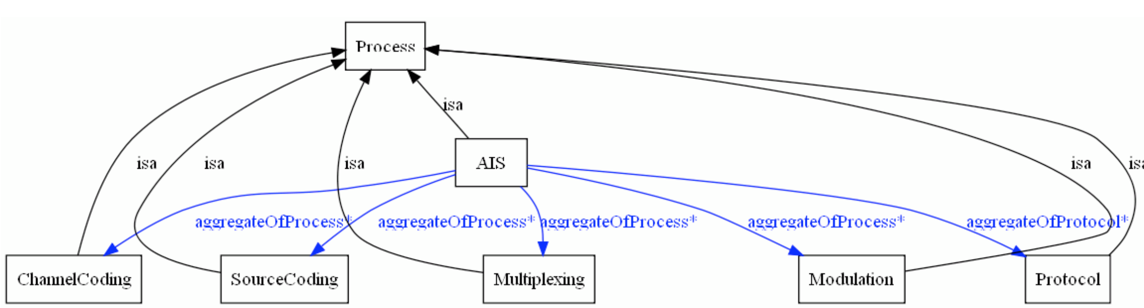

Figure 6 Relationships Among AIS, Protocol and Process

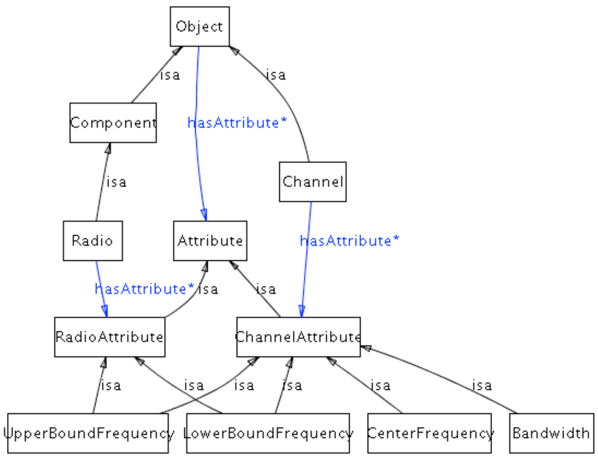

Figure 7 Radio Attribute and Channel Attribute

on frequency, then each sub-channel has a range of wave frequencies. In this sense, *LowerBoundFrequency* and *UpperBoundFrequency* are attributes associated with a wireless channel. However, these attributes can also be associated with a radio device, referring to the boundaries of the operating frequency of a radio. There are at least two ways to conceptualize this issue.

In the first approach, we can make *LowerBoundFrequency* and *UpperBoundFrequency* as the subclasses of Attribute. Channel can be associated with *LowerBoundFrequency* by an object-type property *hasChannelLowerBoundFrequency*; and Radio can be associated with *LowerBoundFrequency* by another object-type property *hasRadioLowerBoundFrequency*. The advantage of this approach is that the relationship between an object (or a process) and its corresponding attribute is very clear, but the drawback is that we need to add a property to each relationship and each property must be named differently.

In the second approach, shown in Figure 7, we can group *LowerBoundFrequency* and *UpperBoundFrequency* into ChannelAttribute, which is a subclass of Attribute. As we saw in Section 2, each object is by default associated with an attribute. We can therefore link Channel to ChannelAttribute by adding a restriction. In addition, classes in OWL can have more than one superclass, hence we can add *LowerBoundFrequency* and *UpperBoundFrequency* as the subclasses of RadioAttribute. This approach provides more flexibility to add new attributes. For instance, if we need to add *CentralFrequency* and *Bandwidth* as the new attributes of Channel, then we only need to add them as the subclasses of ChannelAttribute, without adding more properties to link Channel to the new attributes. The disadvantage of this approach is that Channel and its attribute are linked by a restriction on the property that is inherited from their superclasses.

In this ontology, the second approach is adopted. The relationships between Radio and RadioAttribute, Channel and ChannelAttribute are shown in Figure 7.

4. CONCLUSION AND FUTURE WORK

In conclusion, this paper aims to summarize the issues arisen during the process of designing an ontology in the cognitive radio domain, with focus on comparing different approaches to conceptualize the top-level class division and the internal concept structure. In the future, we will continue our work in the following aspects: (1) representing a waveform, e.g. FM3TR, using this ontology, (2) exploring the inference and reasoning ability with this ontology.

5. REFERENCE

[1] R. Studer, V. R. Benjamins, D. Fensel. "Knowledge Engineering: Principles and Methods", Data and Knowledge Engineering. pp. 161-197, 1998.

[2] B. A. Fette (Ed.), "Cognitive Radio Technology", Newnes Publication, August 2006.

[3] N. Noy, C. Hafner, "The state of the art in ontology design: A survey and comparative review", AI Magazine, 18(3): 53–74, 1997.

[4] C. Masolo, S. Borgo, A. Gangemi, "DOLCE : a Descriptive Ontology for Linguistic and Cognitive Engineering". Technical report, Institute of Cognitive Science and Technology, Italian National Research Council, 2003.

[5] R. Mizoguchi, "On Property: Property vs. Attribute", Technical Report, ISIR, Osaka University, [Online] Available: http://www.ei.sanken.osaka-u.ac.jp/main/documents/OnProperty.pdf.

[6] D. J. Arndt, "Demystifying Radio Frequency Interference: Causes and Techniques for Reduction", Trafford Publishing, pp.88, May 2009.

[7] M. M. Kokar, L. Lechowicz, "Language Issues for Cognitive Radio", Proceedings of the IEEE, 97(4): 689-707, 2009.

[8] S. J. Russell, P. Norvig, "Artificial Intelligence: A Modern Approach (2nd ed.)", Prentice Hall, chapter. 2, 2003

[9] S. Franklin, A. Graesser, "Is it an agent, or just a program?", Intelligent Agents III, Springer, pp. 21–36, 1997.

[10] "IEEE Standard Definitions and Concepts for Dynamic Spectrum Access: Terminology Relating to Emerging Wireless Networks, System Functionality, and Spectrum Management", IEEE Std 1900.1, September 2008.

[11] "cdma2000 High Rate Packet Data Air Interface Specification", 3GPP2 C.S0024-A Version 2.0.

ACKNOWLEDGEMENT

We gratefully acknowledge the following people who have made vital contributions to our research -- Bruce Fette, Joseph Mitola, Mark Cummings, Vince Kovarik, P. A. Subrahmanyam, Todor Cooklev, Bryan Lyles and others who participated in the discussions of the SDR MLM Working Group.