COLLABORATIVE ADAPTATION OF COGNITIVE RADIO PARAMETERS USING ONTOLOGY AND POLICY APPROACH

Shujun Li; Mieczyslaw M. Kokar; David Brady; Jakub Moskal
Department of Electrical and Computer Engineering, Northeastern University, Boston
li.shuj@neu.edu, kokar@coe.neu.edu, d.brady@neu.edu, jmoskal@ece.neu.edu

ABSTRACT

Cognitive radio is expected to be aware of the changing channel environment and adjust the services and resources accordingly by changing the radio's transmission parameters. In order to achieve automatic adaptation, the radio must have the ability to reason about the facts from the environment or from other radios/nodes, i.e. to infer implicit knowledge from the explicitly represented knowledge. It requires a proper language to represent the knowledge and policies and an inference engine that can process the knowledge and policies. However, such awareness and reasoning capabilities, especially combined with optimization and adaptation, have not been previously reported in the literature. Ontology-Based Radio (OBR) has shown its potential to fulfill the awareness and reasoning requirement of cognitive radio. We aim to refine the OBR concept and apply it to a link optimization use case. The general goal is to maximize the information bit rate per transmitter watt of power. This is attained by fine-tuning the parameters in the transmitter and receiver. In this paper, we show that using the ontology and a policy-based approach, two radio nodes are capable to read/modify their knobs and meters, and thus improve the link performance. In this approach, all the internal and external information of the radio is represented in Web Ontology Language (OWL) and the rules to fine-tune each parameter are written in a declarative form and interpreted by a reasoner.

1. INTRODUCTION

Cognitive radio is expected to have the capabilities to (1) sense the environment and collect information of the environment; (2) be aware of the external situation, the internal state and its own capabilities; (3) automatically adapt its parameters and optimize multiple objectives; (4) reason about communications situations, objectives and radio configurations. Some of these capabilities, such as spectrum sensing and opportunistic utilization, are currently actively pursued by various wireless research projects. However, the capabilities of awareness and reasoning, especially combined with optimization, have not been previously reported in the literature.

Awareness, including situation awareness and self-awareness, is the ability to interpret and derive understanding from the input information [1,2]. More specifically, it refers to the perception of the elements in the surrounding environment, the comprehension of their meaning and the projection of their status in the near future [3]. It also refers to the understanding of its previous actions and results, its current performance and capabilities.

Real awareness can be achieved only if the agent can reason about the facts it gets from the environment or from other agents. Reasoning refers to the ability to infer implicit knowledge from the explicitly represented knowledge. Reasoning requires (1) a proper language to represent the knowledge and policies, and (2) a reasoning engine that can process the knowledge and policies.

Awareness and adaptation are the most critical requirements in cognitive radio. Due to the growing complexity and heterogeneity of the communication networks, cognitive radio needs to be aware of the changing context and adjust the services and resources accordingly. The changing context includes the spectrum environment, the user and demands, the network topology, etc.

However, there are a few limits in the current radios that prevent them to fulfill awareness and adaptation: (1) Local information is stored in a data model that does not have high expressivity and machine understandable semantics. For example, in SNMP (Simple Network Management Protocol) or CMIP (Common Management Information Protocol), the information that can be got and set is limited to scalar variables. It is not possible to exchange complicated information such as the structure of a component. XML technology can be integrated with the existing protocols to provide a means to express more complicated information [14,15]. However, the XML-represented information cannot be processed by the inference engine. Therefore, the radio cannot reason and adapt its behavior without human intervention. (2) Signaling messages are limited in the frame structure defined by the protocol, and therefore hinder the ability to achieve the vertical and horizontal mobility among the communication networks.

Therefore, a new approach is required to replace the current static design solution with a more flexible and adaptable solution, e.g., a policy based solution. Also, the radios need to use a language that all of them "understand", so that they are capable to exchange and process information about their situations, goals, components and capabilities [6].

Various academic communities and wireless research projects are actively working towards such a goal. In [12], the authors propose the concept of Ontology-Based Radio (OBR). In the OBR approach, all the internal/external information and the signaling messages are represented in the Web Ontology Language (OWL). OWL is a formal language with high expressivity and computer processable semantics and therefore is capable of expressing complicated information and can be processed by the inference engine. Bearing the same concept, the Modeling Language for Mobility (MLM) working group in the Wireless Innovation Forum is leading an effort to develop a formal language, with computer understandable semantics, that could be used to describe all aspects of network operations and management [6][7][10][11]. Papers [5] and [9] discuss the language issues that arose in the process of developing the ontology and policies for cognitive radio. In [8], we use a public safety use case to demonstrate how to combine ontology, policy and inference engine to control the radio behavior. In addition, the IEEE P1900.5 working group is making an effort to define a policy language to specify interoperable control of the cognitive radio functionality.

Aligned with the above efforts, this paper aims to refine the OBR concept and test it on a link optimization use case. The contributions of this paper include that: (1) we developed a Cognitive Radio Ontology in OWL to represent the basic terms of wireless communications; (2) based on this ontology, we developed a set of policies and rules to optimize the link performance; (3) we showed that the ontology and policy approach can infer implicit knowledge and this implicit knowledge can bring benefits to the communication efficiency.

In the rest of this paper, Section 2 describes a refinement of the OBR architecture. Section 3 introduces the concepts of inference engine, policy and the ontology we have developed. Section 4 shows how to use OBR to solve a link optimization problem. Section 5 briefly discusses the reasoning capability of the OBR and the benefits of such an approach. Finally, Section 6 concludes the paper and outlines the future work.

2. ARCHITECTURE OF OBR

In this paper, we aim to use the OBR approach to adjust some of the radio parameters. There are two types of parameters: knobs and meters. Knobs refer to the adjustable parameters that control the radio's operation and thereby affect the radio performance. Meters refer to the utility or cost functions that are intended to be maximized or minimized in order to achieve optimum radio operation.

In order to enable two radios to negotiate about their knobs and meters, we propose a refined architecture of the OBR as shown in Figure 1.

Radio Platform provides the digital signal processing and software control, as well as the interfaces to communicate with the RF, sensors, information source/sink and the policy reasoner.

System Strategy Reasoner (SSR) is an internal component of the cognitive radio. It forms strategies to control the operation of the radio. The strategies reflect the opportunities, capabilities of the radio and waveform, and the needs of the network and users [17].

All the incoming messages from the RF are first processed by the Radio Platform. Data messages are passed to the radio application (we call it Data Sink), whereas the control messages end up in the SSR. Similarly, all the outgoing control messages are generated by the SSR and then passed to a buffer. The data messages and control messages will be merged in the buffer and then passed to the Radio Platform. After being processed in the Radio Platform, the messages will be sent out through the RF.

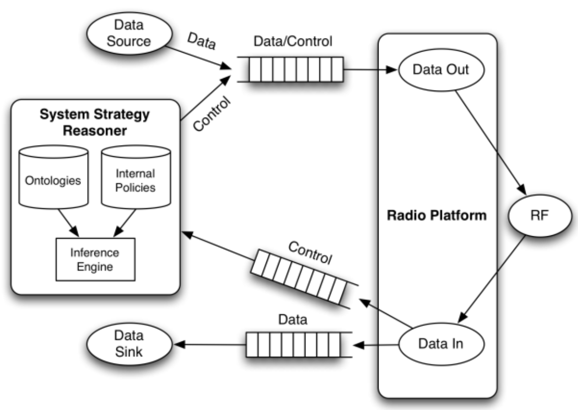

Figure 1 A Refined Architecture of Ontology-Based Radio

There are different control models to describe the control mechanism of the OBR. Figure 2 shows an example of the closed loop feedback control model. SSR can be viewed as a controller. The controller calculates the knobs as a function of the goal and the observed meters. Then, the plant (Radio Platform), being the actual operational part of the radio, collects the knobs, the meters and other sensed information from the environment and computes the Quality of Service metric (QoS). The QoS reflects the overall performance of system. The QoS is sent to the controller (SSR) as a feedback. The controller then evaluates whether the goal is achieved. If not, then the controller changes its input to the plant (knobs) as to achieve the control goal [18].

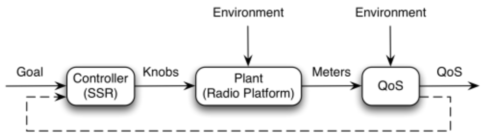

Figure 2 Example Control Model of Ontology-Based

As shown in Figure 3, the control messages can be added to an extensible payload field as needed, rather than embedded in the predefined protocol-dependent header or trailer. This mechanism not only adds signaling flexibility to the existing protocols, but also makes it possible to add the inference capability to a radio without much change of the lower layer architecture.

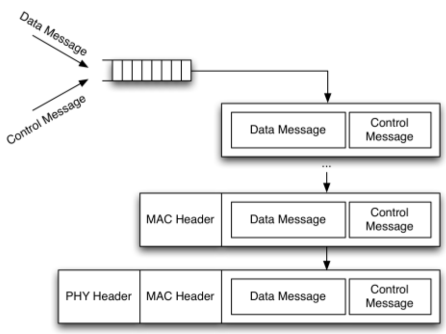

Figure 3 Extensible Payload Field

3. ONTOLOGY AND POLICY

The SSR provides an inference capability to OBR. As shown in Figure 1, SSR consists of an inference engine, an ontology and a set of policies. When the radio starts functioning, the ontology, the policies and the dynamic facts generated by the radio are loaded into the inference engine. Then the inference engine infers new facts and generates control messages such as a command to change the parameter values.

An ontology is a formal, explicit specification of a set of concepts in a specific domain and the relationships between these concepts. The term “formal” means that the ontology is machine processable for the purpose of knowledge reuse and sharing [16]. We developed a Cognitive Radio Ontology to cover the basic terms of wireless communications from the PHY and MAC layer. This ontology is written in OWL and has 230 classes and 188 properties. The documentation of Cognitive Radio Ontology is published in [11] and [9]. Using the classes and properties defined in this ontology, we can develop policies to control the behavior of the radio.

A policy is a set of rules written in a policy language. In our implementation, we use BaseVISor as the inference engine. BaseVISor is a forward-chaining rule engine optimized for handling facts in the form of RDF triples. The engine also supports XML Schema Data Types. An RDF triple consists of a subject, a predicate and an object. The subject and the object denote resources (things in the domain of discourse), and the predicate denotes a relationship between the subject and the object.

The following is an example of a rule in RDF triples form:

```
<rule name="checkPerformance">
 <body>
   <triple>
    <subject variable="X" />
    <predicate resource="rdf:type" />
    <object resource="rad:SignalDetector" />
   </triple>
   <triple>
     <subject variable="X" />
     <predicate resource="rad:signalToNoiseRatio"
/>
     <object variable="SNR" />
    </triple>
   <lessThan>
     <param variable="SNR" />
     <param rdf:datatype="xsd:float">15</param>
   </lessThan>
   <greaterThan>
     <param variable="SNR" />
     <param rdf:datatype="xsd:float">10</param>
   </greaterThan>
 </body>
 <head>
   <assert>
     <triple>
       <subject variable="X" />
       <predicate resource="rad:performance" />
       <object
rdf:datatype="xsd:string">acceptable</object>
     </triple>
   </assert>
 </head>
</rule>
```

This rule states that if the SNR is smaller than 15 and larger than 10, then the performance is acceptable. In the BaseVISor syntax, the subject, predicate and object element can be a resource, a XML data type or a variable. If an element is a resource, e.g. SignalDetector, then this element is defined in the ontology.

As is the case with some inference engines, it is possible to extend the BaseVISor functionality by adding new functions. These are called *procedural attachments* or *functions*.

The following is an example of computing the objective function in Section 3.

```
<bind>
   <param variable=”objFunc_PowdB”/>
   <computeObjFunc>
       <param variable=”PowdB_new”/>
       <param variable=”trainPeriod”/>
       <param variable=”m”/>
       <param variable=”v”/>
   </computeObjFunc>
</bind>
```

In the above example, <*computeObjFunc*> is a user-defined procedural attachment. This function has four arguments and returns the value of the objective function. The return value is bind to variable *objFunc_PowdB.*

4. AN EXAMPLE: LINK OPTIMIZATION

In this paper, we use a link optimization use case to show how OBR can be used to achieve automatic adaptation in cognitive radio.

In the link optimization use case, there is a transmitter-receiver pair. The general goal of link optimization is to maximize the information bit rate per transmitted watt of power, subject to a set of constraints. This is attained by fine-tuning the parameters in the transmitter and the receiver. Here, the OBR provides a means to exchange the control messages between the transmitter and the receiver.

The tunable parameters (knobs) and the observable parameters (meters) of the transmitter and receiver are shown in Table 1.

Table 1 Parameters of Transmitter and Receiver

Tx	PowdB	Transmission Power (Unit:dBm)	Knob
	trainPeriod	Length of training sequence (Unit: channel symbol)	Knob
	m	Index of (2^m-1, 2^m-1-m) Hamming Code	Knob
	v	Integer of QAM modulation, 4^v is the size of QAM constellation	Knob
	Payload	Size of payload field. Set payload=128 bytes	Fixed
	fracSpacing	Number of samples per symbol. Set fracSpacing=2	Fixed
	sampleRate	Number of samples per second. Set sampleRate=1000	Fixed
Rx	M	Number of feedback taps	Knob
	N1	Number of precursor feedforward taps	Knob
	N2	Number of postcursor feedforward taps	Knob
	mSNR	Mean Signal-to-Noise Ratio	Meter

4.1. Objective Function

The goal is to maximize the information bit rate per transmitted watt of power:

$$InformationBitRatePerTxPower = \frac{BitsOfPayload/Time}{TxEnergy/Time} = \frac{BitsOfPayload}{TxEnergy} = \frac{payload\cdot 8}{TxEnergy}$$

The transmission energy (Unit: Joules) equals to:

$$TxEnergy = TxPower\cdot Time = \frac{10^{\frac{PowdB}{10}}}{1000}\cdot\frac{TotalNumberOfSymbol\cdot fracSpacing}{SampleRate}$$

The total number of symbols in the packet is:

$$TotalNumberOfSymbol = [(32+8\cdot payload)\cdot(1+\frac{m}{(2^m-m-1)})\cdot\frac{1}{2v}+trainPeriod]$$

Assume that $payloadsize = 128$, $SampleRate = 1000$, and $fracSpacing = 2$ are fixed. Then the goal is to minimize the following objective function:

$$\boxed{objFunc = 10^{\frac{PowdB}{10}}[\frac{528\cdot(1+\frac{m}{2^m-m-1})}{v}+trainPeriod]}$$

4.2. Constraints

(1) The reported equalizer $mSNR$ must be between $10dB$ and $15dB$. Intuitively, a value greater than $10dB$ yields good detection performance, but a value greater than $15dB$ indicates that the data rate could be increased, or the transmit power should be decreased. Hence, the constraint for $mSNR$ is: $10dB \le mSNR \le 15dB$.

(2) $PowdB$ is the transmission power in dBm. Here, we set the upper bound of $PowdB$ as: $PowdB \le 0dB$. Since both $PowdB$ and $mSNR$ are in dB, a drop of $mSNR$ results in an equal drop in $mSNR$. Thus, $\Delta PowdB = \Delta mSNR$.

(3) m controls the coding overhead of $(2^m-1, 2^m-1-m)$ Hamming code. m does not affect the $mSNR$. m has natural lower bound of 3 and has no natural upper bound. Here, we set the constraint of m as: $3 \le m \le 7$.

(4) v controls the size of the QAM modulation. The natural lower bound of v is: $v \ge 1$. Increasing v by one unit drops the $mSNR$ by approximately $6dB$. Thus, $\Delta v = \Delta mSNR/6$.

(5) $trainPeriod$ affects the $mSNR$ in a less clear way. If $trainPeriod$ is less than $5\cdot(M+N1+N2)$, then the equalizer does not fully converge, thus $mSNR$ will be decreased. If $trainPeriod$ is greater than $5\cdot(M+N1+N2)$, then it will have little effect on $mSNR$, but will work against the minimization of the metric. Hence, the constraint of $trainPeriod$ is : $5\cdot(M+N1+N2) \le trainPeriod \le 10\cdot(M+N1+N2)$.

(6) M, $N1$, $N2$ have a threshold influence on $mSNR$: the $mSNR$ will increase with M, $N1$, $N2$, until a sufficiently large equalizer for the multipath is achieved. After that point, increasing the equalizer dimensions will have no effect, except to increase the shortest possible training sequence.

4.3. Policy

4.3.1. Policies to establish the communications link

The link optimization is accomplished by five transmissions, shown in Figure 4.

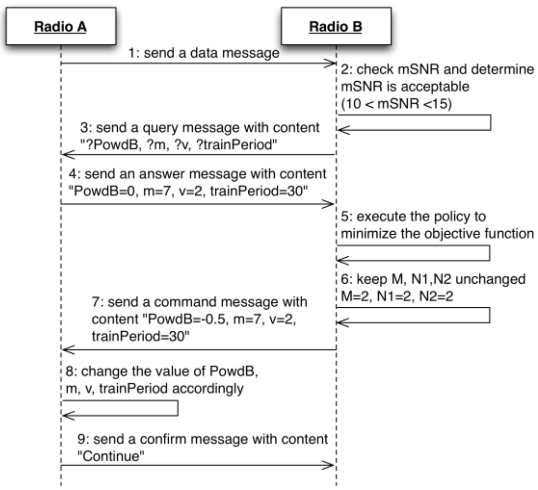

Figure 4 Sequence Diagram of Link Optimization

Suppose we initialize radio A as the transmitter and radio B as the receiver. Then after radio A sends a data message to radio B, radio B will check the performance by measuring the *mSNR*. If the *mSNR* is within an acceptable range, then radio B will start to minimize the objective function. Radio B will first send a "query" message to radio A, asking for the current values of its parameters. When radio A receives this query, it will infer the answer and send the answer back to radio B. After radio B receives the answer, it will trigger the policy to compute the new values of its local parameters and the parameters of radio A. Then radio B will set its local parameters to the new values and generate a command to radio A. The "command" message contains the new values of radio A's parameters. After radio A receives the command, it will execute the command and set its parameters accordingly. A "confirm" message is then sent to radio B.

The following is an example rule of reacting to the received message:

```
Rule "checkPerformance":
If the radio receives a data message, then (1)
check mSNR; (2) generate a query message with
content "?PowdB, ?trainPeriod, ?m, ?v"; (3)
send the query to the originating radio.
```

4.3.2. Policies to do the link optimization

The goal of link optimization is to minimize *objFunc*. However, the decrease of *objFunc* will worsen the performance and decrease the *mSNR*. In other word, there is a tradeoff between the decrease of *objFunc* and the improvement of *mSNR*.

We implemented three sets of policies with different preferences. Policy1 decreases the *objFunc* as much as possible while not guaranteeing *mSNR* within the acceptable range. Policy2 decreases *objFunc* to an intermediate level while maintain *mSNR* in the acceptable range as possible. Policy3 decreases *objFunc* while guaranteeing the *mSNR* within the acceptable range.

The following shows the description of Policy3; it contains 4 rules:

```
Rule 1:
If mSNR > 15, then tune M, N1, N2 as follows:
     ΔM = -2, ΔN1 = -2, ΔN2 = -2.

Rule 2:
If mSNR > 12.5, then tune one of these
parameters: PowdB, trainPeriod, m or v.

(1) Compute the following:
     PowdB_new= min((12.5 - mSNR + PowdB), 0)
     trainPeriod_new = min (7.5 *(M+N1+N2),
     trainPeriod)
     m_new = 7
     v_new= max (v, floor ((mSNR - 12.5)/6)+v)

(2) Compute the following objective function:
     objFunc (PowdB_new, trainPeriod, m, v)
     objFunc (PowdB, trainPeriod_new, m, v)
     objFunc (PowdB, trainPeriod, m_new, v)
     objFunc (PowdB, trainPeriod, m, v_new)

(3) Choose the smallest objective function
from (2) and tune the corresponding parameter
to the new value.

Rule 3:
If mSNR <= 12.5, then tune one of these
parameters: PowdB, trainPeriod, m or v.

(1) Compute the following:
     PowdB_new= min((15 - mSNR+PowdB), 0)
     trainPeriod_new = min (10 *(M+N1+N2),
     trainPeriod)
     m_new = 0
     v_new= max ( v, floor ((mSNR-15)/6)+v)

(2) Compute the following objective function:
     objFunc (PowdB_new, trainPeriod, m, v)
     objFunc (PowdB, trainPeriod_new, m, v)
     objFunc (PowdB, trainPeriod, m_new, v)
     objFunc (PowdB, trainPeriod, m, v_new)

(3) Choose the smallest objective function
from (2) and tune the corresponding parameter
to the new value.

Rule 4:
If mSNR < 10, then tune M, N1, N2 as follows:
     ΔM = +2, ΔN1 = +2, ΔN2 = +2.
```

All the above policies are written in the BaseVISor syntax.

In our implementation, we use MATLAB to emulate a Rayleigh multipath channel between two radios. Each radio is connected to BaseVISor. When a new message comes in, BaseVIsor will be invoked to do the reasoning. The outputs of BaseVISor include new values of the parameters for the next transmission.

In order to evaluate whether the policy is able to adapt to the change of the channel environment, we linearly increase the number of multipath from 2 to 16. Assume the radios are operating in half-duplex mode. The default parameter values are: $PowdB = 0$, $m = 3$, $v = 1$, $trainPeriod = 100$, $M1 = 2$, $N1 = 2$, $N2 = 2$. First, we set the number of multipath to 2. Then radio A sends the 1st data message to radio B. When radio B receives the 1st data message, it measures $mSNR$ and $objFunc$. Based on the current values of $mSNR$ and $objFunc$, the two nodes will follow the steps described in Figure 4 to negotiate the parameters for the 2nd data message and then set their parameters to the new values. Then we change the channel environment by setting the number of multipath to 4. After that, radio A sends the 2nd data message to radio B and repeats the above steps. In total, radio A sends 8 data messages to radio B. The simulation results and the comparison of these three policies are shown in Figure 5. It can be seen that without link optimization, $objFunc$ remains at the same value and $mSNR$ fluctuates as the number of multipath changes. With link optimization, $objFunc$ is significantly decreased. Policy1 decreases $objFunc$ by 66% at the price of decreasing $mSNR$ by 1.83dB. Policy2 decreases $objFunc$ by 55% at the price of decreasing $mSNR$ by 0.83dB. Policy 3 decreases $objFunc$ by 36% while increasing $mSNR$ by 0.09dB.

5. INFERENCE IN OBR

One of the advantages of the OBR approach is that it can infer implicit knowledge. In this section, we use the link optimization use case to show what benefits this implicit knowledge can bring to the communications.

In the case of link optimization, the radios need to exchange information about their knobs and meters. So theoretically, radios might need to send values of 3,000 of such parameters. This would impose a lot of communications burden leading to a high need for spectrum. However, not all the 3,000 parameters are needed in the link optimization. Generally, in each transmission, these 3,000 parameters can be divided into 3 groups: (1) parameters that need to change; (2) parameters that are fixed; (3) parameters that we don't care. For example, in our use case, we only care about 7 parameters: $PowdB$, $trainPeriod$, m, v, M, $N1$, $N2$. Suppose radio B needs to send a command to radio A, requesting it to change the values of $PowdB$ to -5.5 and keep $trainPeriod$, m, v unchanged. To address this scenario, we can extend the ontology developed so far to include the class *Configuration*. The Configuration class will include all the 3,000 parameters as its properties. Then any combination of these 3,000 parameters can be viewed as the super-classes of Configuration. For instance, as shown in Figure 6, class *Config1* includes property PowdB; class *Config2* includes properties trainPeriod, m and v. Config1 and Config2 are the super-classes of Configuration.

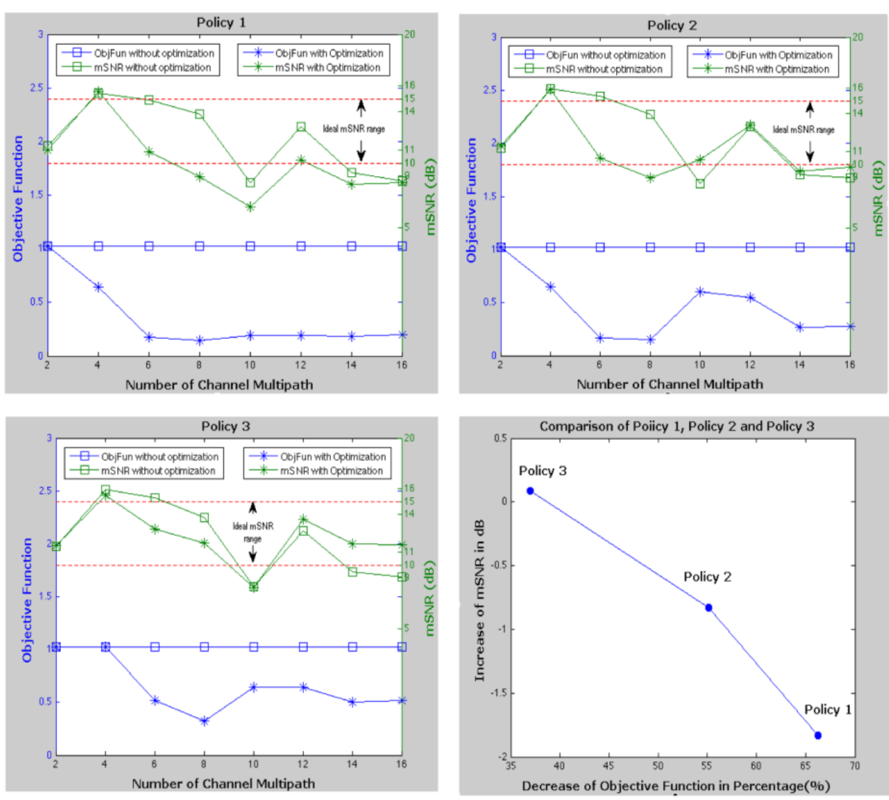

Figure 5 Comparisons of Policy 1, Policy 2 and Policy 3

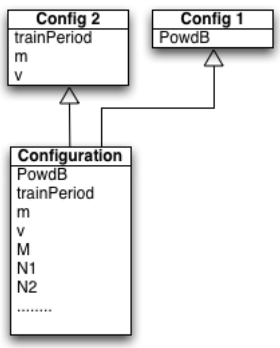

Figure 6 Configuration Class

Assume that radio A and B already share a common ontology as shown in 6. Then in order to request radio A to change the values of *PowdB* to -5.5 and keep *trainPeriod* , *m* , *v* unchanged, radio B only needs to send a command message to radio A containing the following information: (1) the name of class *Config2*; (2) an instance of class *Config1* that includes the new value of *PowdB*.

The OWL representation of the instance of class *Config1* is shown below:

```
<Config1 rdf:ID="Config1_instance">
  <PowdB>
    <Power rdf:ID="Power_instance">
      <hasValue>
        <FloatValue rdf:ID="FloatValue_instance">
          <hasFloat rdf:datatype="xsd:float"
          >-5.5</hasFloat>
        </FloatValue>
      </hasValue>
      <hasUnitOfMeasure>
        <UnitOfMeasure rdf:ID="dBm"/>
      </hasUnitOfMeasure>
    </Power>
  </PowdB>
</Config1>
```

When radio A gets this command, it is able to infer that radio B commands it to change *PowdB* to -5.5 and keep *trainPeriod*, *m*, *v* unchanged. This simple example shows that if the radios have this kind of information encoded in their ontologies or rules, they do not need to send all the information, but instead may infer the rest of the values locally.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we refine the OBR concept and use a link optimization example to show how to use the ontology and policy approach to optimize the performance of a communication link. Using our proposed architecture and the ontology and rules we developed, the two radios are capable of negotiating their knobs and meters, and thus improve the link performance. Also, we presented a simple example to show that the ontology and policy approach has the potential to achieve some degrees of awareness and reasoning, and thus brings improvement to the communications, such as lightening the communication burdens.

In the future, we will implement the link optimization on GNU radio and further investigate the control overhead and the time efficiency of the OBR.

7. ACKNOWLEDGEMENT

This work has been partially supported by a DARPA contract through System Planning Corporation (Agreement No. SRA-0775). The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

8. REFERENCES

[1] WIF Forum Cognitive Radio Working Group, “Cognitive Radio Definition and Nomenclature”, Working Document SDRF-06-P-0009-V0.5.0, May 30, 2008.

[2] M. M. Kokar, D. Brady, K. Baclawski, B. A. Fette (Ed.), “Cognitive Radio Technology”, Chapter 13 “The Role of Ontologies in Cognitive Radios”, Elsevier, August 2009.

[3] M. Endsley and D. Garland, Situation Awareness, Analysis and Measurement, Lawrence Erlbaum Associates, Publishers, Mahway, New Jersey, 2000.

[4] M. M. Kokar, K. Baclawski, Y. A. Eracar, “Control Theory-Based Foundations of Self-Controlling Software”, IEEE Intelligent System, May/June, 1999.

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

[6] M. M. Kokar, D. Hillman, S. Li, “Towards a Unified Policy Language for Future Communication Networks: A Process”, DySPAN Conference, Chicago, IL, 2008

[7] WIF Forum MLM Working Group, “Use Cases for MLM Language in Modern Wireless Networks”, Working Document SDRF-08-P-0009-V.1.0.0, 2008.

[8] S. Li, M. M. Kokar, J. Moskal, “Policy-driven Ontology-Based Radio: A Public Safety Use Case”, SDR Forum Technical Conference, Dec. 2008.

[9] S. Li, M. M. Kokar, D. Brady, “Developing an Ontology for the Cognitive Radio: Issues and Decisions”, SDR Forum Technical Conference, Dec. 2009.

[10] M. Cummings, S. Li, M. M. Kokar, “Activities of SDR Forum MLM Working Group on a Language for Advanced Communication Systems Applications”, SDR Forum Technical Conference, Dec. 2008.

[11] WIF Forum MLM Working Group, “Description of Cognitive Radio Ontology v.1.0”, 2010.

[12] J. Wang, M.M. Kokar, K. Baclawski, and D. Brady, “Achieving Self-Awareness of SDR Nodes through Ontology-Based Reasoning and Reflection”, Software Defined Radio Technical Conference, 2004.

[13] T. W. Rondeau, “Application of Artificial Intelligence to Wireless Communications”, Ph.D. dissertation, VT, 2007.

[14] D. Shin, C. Shim, “XNMP – an XML based Network Management Protocol over VoIP”, Proceedings of the 6th International Conference on Software Engineering, Artificial Intelligence, Networking and Paralled/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks, VT, 2005.

[15] M. J. Choi, W. Hong, H. T. Ju, "XML-based Network Management for IP Networks", ETRI Journal, 2003.
[16] R. Studer, V. R. Benjamins, D. Fensel, "Knowledge Engineering: Principles and Methods", Data and Knowledge Engineering, pp. 161-197, 1998.
[17] G. Denker, D. Elenius, D. Wilkins, B. A. Fette (Ed.), "Cognitive Radio Technology", Chapter 17 "Cognitive Radio Policy Language and Policy Engine", Elsevier, August 2009.
[18] M. M. Kokar, K. Baclawski, Y. Eracar, "Control theory-based foundations of self-controlling software", IEEE Intelligent Systems, pp. 37-45, May/June 1999.