A Semantic Framework for Enabling Radio Spectrum Policy Management and Evaluation⋆

Henrique Santos[1][0000−0002−2110−6416], Alice Mulvehill[1,3][0000−0003−2981−0675], John S. Erickson[1,2][0000−0003−3078−4566], James P. McCusker[1][0000−0003−1085−6059], Minor Gordon[1][0000−0003−1928−4130], Owen Xie[1][0000−0002−8689−3385], Samuel Stouffer[1][0000−0002−8929−5989], Gerard Capraro[4][0000−0002−1836−9231], Alex Pidwerbetsky[5][0000−0001−7436−8105], John Burgess[5][0000−0001−9539−6922], Allan Berlinsky[5][0000−0001−5314−7178], Kurt Turck[6][0000−0001−9224−0164], Jonathan Ashdown[6][0000−0001−7202−1095], and Deborah L. McGuinness[1][0000−0001−7037−4567]

[1] Tetherless World Constellation, Rensselaer Polytechnic Institute, Troy NY, USA
[2] The Rensselaer Institute for Data Exploration and Applications, Troy NY, USA
[3] Memory Based Research LLC, Pittsburgh PA, USA
[4] Capraro Technologies Inc., Utica NY, USA
[5] LGS Labs, CACI International Inc., Florham Park NJ, USA
[6] Air Force Research Laboratory, Rome NY, USA
{oliveh,mulvea,erickj4,mccusj2,gordom6,xieo,stoufs}@rpi.edu,
dlm@cs.rpi.edu, gcapraro@caprarotechnologies.com,
{a.pidwerbetsky,john.burgess,allan.berlinsky}@caci.com,
{kurt.turck,jonathan.ashdown}@us.af.mil

Abstract. Because radio spectrum is a finite resource, its usage and sharing is regulated by government agencies. These agencies define policies to manage spectrum allocation and assignment across multiple organizations, systems, and devices. With more portions of the radio spectrum being licensed for commercial use, the importance of providing an increased level of automation when evaluating such policies becomes crucial for the efficiency and efficacy of spectrum management. We introduce our Dynamic Spectrum Access Policy Framework for supporting the United States government's mission to enable both federal and non-federal entities to compatibly utilize available spectrum. The DSA Policy Framework acts as a machine-readable policy repository providing policy management features and spectrum access request evaluation. The framework utilizes a novel policy representation using OWL and PROV-O along with a domain-specific reasoning implementation that mixes GeoSPARQL, OWL reasoning, and knowledge graph traversal to evaluate incoming spectrum access requests and explain how applicable policies were used. The framework is currently being used to support live, over-the-air field exercises involving a diverse set of federal and commercial radios, as a component of a prototype spectrum management system.

Keywords: Dynamic spectrum access · Policies · Reasoning

---

1   Introduction

Usable radio spectrum is becoming crowded[7] as an increasing number of services, both commercial and governmental, rely on wireless communications to operate. Techniques known as Dynamic Spectrum Access (DSA) [26] have been extensively researched as a way of promoting more efficient methods for sharing the radio spectrum among distinct organizations and their respective devices, while adhering to regulations.

In the United States, spectrum is managed by agencies that include the National Telecommunications and Information Administration[8] (NTIA) and the Federal Communications Commission[9] (FCC). The NTIA publishes revised versions of its Manual of Regulations and Procedures for Federal Radio Frequency Management[10] (commonly referred to as the NTIA Redbook) which is a compilation of regulatory policies that define the conditions that non-US, as well as federal and non-federal US, organizations, systems, and devices must satisfy in order to compatibly share radio spectrum while minimizing interference.

With the advent of 5G,[11] more parts of the radio spectrum are being licensed for commercial usage and, with the increased availability of cognitive radios [25] (devices that are able to automatically adjust their operating frequency), the importance of providing an increased level of automation when evaluating spectrum policies becomes crucial for the sustainability of spectrum management. This issue is currently under investigation by the National Spectrum Consortium[12] (NSC), a research and development organization that incubates new technologies to tackle challenges about radio spectrum management and utilization.

In this paper, we describe the Dynamic Spectrum Access Policy Framework (DSA Policy Framework). The DSA Policy Framework supports the management of machine-readable radio spectrum usage policies and provides a request evaluation interface that is able to reason about the policies and generate permit or deny results to spectrum access requests. This is accomplished via the utilization of a novel policy representation based on semantic web standards OWL and PROV-O that encodes its rules in an ontology. This ontology, combined with background knowledge originating from a number of relevant select sources, is stored in a Knowledge Graph that is used by a domain-specific reasoning implementation that mixes GeoSPARQL [22], OWL reasoning, and knowledge graph traversal to evaluate policies that are applicable to spectrum access requests. Effects (Permit/Deny/Permit with Obligations) are assigned and explanations are provided to justify why a particular request was permitted or denied access to the requested frequency or frequency range.

---

[7] http://bit.ly/FCC_AWS
[8] http://ntia.doc.gov
[9] http://fcc.gov
[10] http://bit.ly/NTIA_Redbook
[11] http://bit.ly/FCC_5G
[12] http://nationalspectrumconsortium.org

2   Sharing the radio spectrum

Radio spectrum policies specify how available spectrum should be used and shared. The applicability of existing policies must be checked for a variety of activities that use spectrum in many settings including various different types of requesters (e.g. systems and devices). For example, training exercises will request spectrum usage for a certain time frame and geographic region for potentially hundreds of radios with a wide variety of capabilities. During a training exercise, local policies are typically created to manage the spectrum that radios used in the exercise will require, and to minimize interference between federal and commercial (non-federal) radios that may be operating in the same area and within the same frequency range. Throughout this paper the following definitions are used:

- *High-level policies*: Policies as documented by authoritative agencies, including NTIA and FCC. These policies are not prone to change in the short term, although they may evolve when new versions of documents are released.
- *Local policies*: Specializations of high-level policies. These are created to locally manage the spectrum requests of various devices that want to operate in specific locations and/or for specific time periods.
- *Sub-policies*: A sub-section of a policy, sometimes referred to as "provisions" in NTIA Redbook policies.
- *Spectrum manager*: The role of a human who is responsible for managing policies. The spectrum manager verifies if existing policies are sufficient to support some activity and creates local policies to accommodate specific spectrum requirements.
- *Spectrum system*: This role represents some external system that is being used to generate spectrum requests on behalf of entities that require the use of a specific frequency or set of frequencies.

3   Dynamic Spectrum Access Policy Framework

The DSA Policy Framework supports the following objectives:

- Serve as a centralized machine-readable radio spectrum policy repository.
- Provide policy management features (including creation and customization) for a wide range of radio spectrum domain users.
- Use machine-readable policies as a basis for automatically evaluating radio spectrum access requests.

As shown on the top of Figure 1, the DSA Policy Framework provides two major functions: *Policy Management* and *Request Evaluation*. *Policy Management* enables the spectrum manager to create local policies by referencing relevant higher level policies and adding customization. *Request Evaluation* utilizes all policies to automatically process spectrum requests. The results include references to any policy that was involved in the evaluation. As policies evolve,

the underlying knowledge representation evolves, enabling the *request evaluation* module to use the most current policy information to reason and assign effects (permit, deny, permit with obligation) to spectrum requests. Spectrum managers can verify evaluation results in the presence of newly created policies through the *Request Builder* tool.

To support the DSA Policy Framework infrastructure we leveraged Whyis [20], a nanopublication-based knowledge graph publishing, management, and analysis framework. Whyis enables the creation of domain and data-driven knowledge graphs. The DSA Policy Framework takes advantage of the use of nanopublications [15] in Whyis, which allows it to incrementally evolve knowledge graphs while providing provenance-based justifications and publication credit for each piece of knowledge in the graph. This is particularly useful for the spectrum domain because policies do change and new policies can be created, potentially triggered by multiple sources. The DSA Policy Framework also makes use of the SETLr [19] Whyis agent, which enables the conversion of a number of the identified knowledge sources or derivatives to the Resource Description Framework (RDF), thereby bootstrapping the DSA Knowledge Graph.
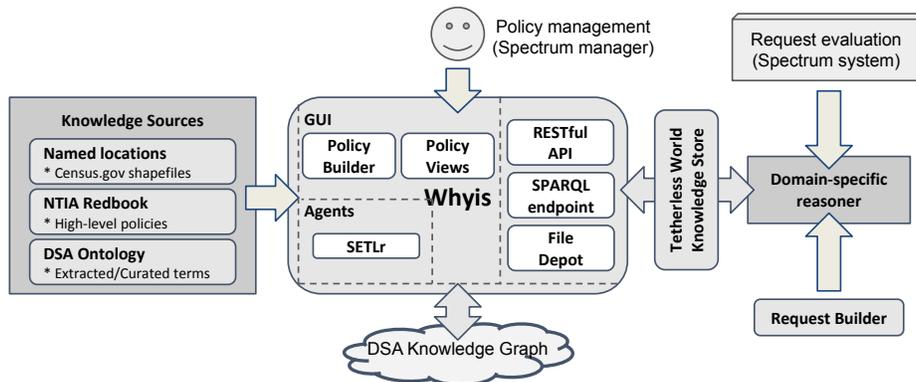

Fig. 1. DSA Policy Framework architecture

3.1   Knowledge Sources

In order to support the creation and interpretation of machine-readable radio spectrum policies, information from several sources was mined and incorporated into a Knowledge Graph. Spectrum policies and term definitions were obtained from the NTIA Redbook, an IEEE Standard Definitions and Concepts document [3], and from FCC 14-31 [4] (a FCC policy publication). Specific schemas associated with the spectrum domain were obtained from the Standard Spectrum

Resource Format (SSRF) [5]. Finally, information about geographical locations was obtained from the Census.gov shapefiles.[13]

During the *policy capture* process, Rensselaer Polytechnic Institute (RPI) collaborated with DSA domain experts from Capraro Technologies Inc. and LGS Labs of CACI International Inc. to select and analyze English text-based policies from the NTIA Redbook and from various FCC documents. In order to be used by the DSA Policy Framework, the English text was converted into a different representation, and many of the terms used in the English text were incorporated into a domain ontology. During this process, we observed that the text for many policies is logically equivalent to a conditional expression e.g., *IF* some device wants to use a frequency in a particular frequency range *AND* at a particular location, *THEN* it is either *PERMITTED* or *DENIED*.

More complex policies contain a set of conditional expressions, with each conditional expression focused on a particular request attribute, e.g., a request device type, frequency, frequency range. The spreadsheet displayed in Figure 2 contains an example of a complex policy from the NTIA Redbook called US91. Due to space constraints, we have omitted some of the sub-policies for US91 and the columns that document policy metadata and provenance, including the original text, source document, URL, and page number.

RuleID	Parsed logical rule	Requester	Affiliation	Frequency	Location	Effect	Obligation
US91	IF RefFreq is ε (≥ 1755 MHz AND ≤ 1780 MHz) then the following provisions apply						
US91-1	IF TR is AWS AND RefFreq is ε (≥ 1755 MHz AND ≤ 1780 MHz) AND (TR has successfully coordinated on a nationwide basis prior to operation, unless otherwise specified by Commision rule, order , or notice) THEN TR is Primary	AWS	non Federal	1755 MHz - 1780 MHz		Permit with Obligation	TR has successfully coordinated on a nationwide basis prior to operation, unless otherwise specified by Commision rule, order , or notice and TR is Primary
US91-2	IF TR is Federal AND RefFreq is ε (≥ 1755 MHz AND ≤ 1780 MHz) AND TR is operating in one the following locations AND (Until reaccommodated in accordance with 47 CFR 301) THEN it is Primary	TR	Federal	1755 MHz - 1780 MHz	Yuma Proving Ground, Fort Irwin, Fort Polk, Fort Bragg, White Sands Missile Range, Fort Hood	Permit with Obligation	Operate on a co-equal, primary basis with AWS stationsTR can't transmit Until re-accommodated in accordance with 47 CFR 301
US91-3	IF TR is Federal AND JTRS AND RefFreq is ε (≥ 1755 MHz AND ≤ 1780 MHz) AND TR is operating in one the following locations THEN It is Primary	JTRS	Federal	1755 MHz - 1780 MHz	Yuma Proving Ground, Fort Irwin, Fort Polk, Fort Bragg, White Sands Missile Range, Fort Hood	Permit with Obligation	JTRS is Primary

Fig. 2. Spreadsheet excerpt showing the NTIA Redbook US91 policy capture

US91 regulates the usage of the 1755-1780MHz frequency range and is an example of a spectrum range that must be efficiently shared by both federal and non-federal devices. Because spectrum usage can vary by device, affiliation and location, we decompose the policy into several sub-policies (US91-1, US91-2, and US91-3). A *parsed logical rule* is manually created for each policy by a domain expert. The elements of the logical rule are further expressed as attribute-value

---

[13] http://bit.ly/Census_shapefiles

pairs, e.g., `Requester = AWS`. The attribute (column) names map into the following elements of the *policy logical expression* that is used by the framework:

– *Requester:* the device requesting access
– *Affiliation:* the affiliation of the requester (Federal, Non-Federal)
– *Frequency:* the frequency range or single frequency being requested
– *Location:* location(s) where the policy is applicable
– *Effect:* the effect a policy yields, if the rule is satisfied (Permit, Deny, Permit with Obligations)
– *Obligations:* the list of obligations the requester needs to comply with in order to be permitted

The framework utilizes several ontologies to support policy administration and spectrum request processing, including a domain ontology called the DSA Ontology. DSA ontology terms were collected during policy capture and/or derived from the NTIA Redbook, IEEE Standards, SSRF, or other domain source. All terms were curated by an ontology developer and linked to external ontologies including PROV-O and the Semanticscience Integrated Ontology (SIO) [7].

3.2   Representing radio spectrum requests

Figure 3 shows the DSA request model and a sample request. The model is based on the World Wide Web Consortium's recommended standard for provenance on the web (PROV). The modeling of requests as activities and agents was influenced by the policy attributes described as columns in the policy capture spreadsheet (shown in Figure 2) and extended to include the action associated with the requester in a spectrum request (currently we represent only the `Transmission` action). In the model, the requester (`prov:Agent`) is linked with an action `prov:Activity` using the `prov:wasAssociatedWith` predicate. The location attribute is represented as `prov:Location` and linked to the requester using the `prov:atLocation` predicate. The time attribute, which describes when the request action is to start and end, is represented using the literal data type `xsd:dateTime` and linked to the action using the `prov:startedAtTime` and `prov:endedAtTime` predicates.

For attributes not natively supported by PROV, we use SIO, which enables us to model objects and their attributes (roles, measurement values) with the use of the `sio:Attribute` class and `sio:hasAttribute`/`sio:hasValue`/`sio:hasUnit` predicates. In the DSA request model, the attributes Frequency, Frequency Range, and Affiliation are represented as `sio:Attribute` and linked to the requester using `sio:hasAttribute`. For attributes that can assume a value (currently only Frequencies), literal values (`sio:hasValue`) and units of measurement (`sio:hasUnit`) are used to express the quantification of an attribute as a specified unit from the Units of Measurement Ontology (UO) [13]. In Figure 3, a sample radio spectrum request instance is shown, where the requester is a Generic Joint Tactical Radio System (JTRS) radio. The device is physically located at a location that is defined by the Well-Known Text (WKT) [2] string
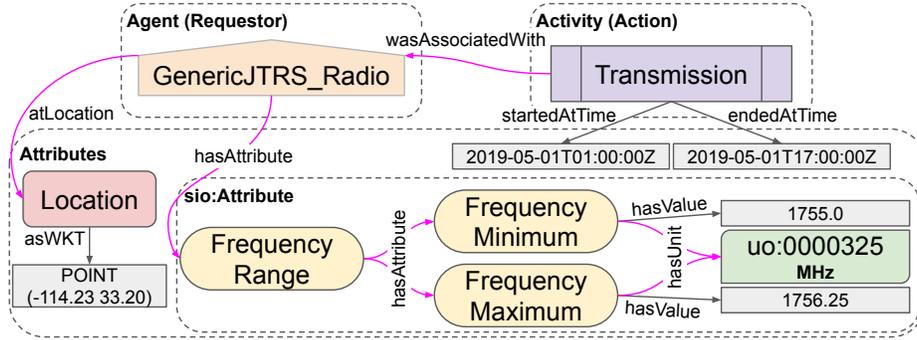
Fig. 3. The DSA request model and sample request

POINT(-114.23 33.20) and is requesting access to the frequency range 1755
- 1756.25 MHz using the FrequencyRange attribute, which is described
by the composition of the FrequencyMinimum and FrequencyMaximum at-
tributes, with their respective value and unit.

3.3 Representing radio spectrum policies

The DSA policy model was created to enable (1) the unambiguous representation
of rules as parsed during *policy capture*, (2) the reuse of an existing policy's rules
for creating local policies, and (3) the implementation of request evaluation
capabilities. The model is based on OWL, encoding policy rules as restrictions
in OWL classes that represent policies. The OWL restrictions are constructed
on the RDF properties of the DSA request model, presented in the previous
subsection, while constraining their ranges to the expected literal value or class,
as dictated by the policy rule being expressed. To demonstrate this, Listing 1.1
shows the OWL expression in Manchester syntax of an excerpt of the NTIA US91
policy that describes how a requester that is an example of a radio, categorized
as JTRS, can use the spectrum regulated by US91 (1755-1780 MHz) in specific
locations (US91-3 sub-policy in Figure 2).

The model advocates for the creation of an OWL class for each rule that
composes the complete policy rule expression. In the example, the OWL class
US91 has the restriction on the frequency range attribute, with the minimum
value 1755 and maximum value 1780 (lines 4-11). This class is a subclass of
Transmission (line 13), which is the action this policy regulates. In lines 15-
20, a class US91-3 restricts the requester to a JointTacticalRadioSystem
and appends that rule to the rules of its super-class US91. The last class of this
policy expression is US91-3.1, which encodes the location rule as a restriction
on the atLocation property (lines 24-25), constraining it to a location class.
When the composition of rules expressed by this last class is evaluated to be
true, the policy should assign the permit effect. This is expressed by asserting
US91-3.1 as a subclass of Permit (line 27).

```
 1 Class: US91
 2   EquivalentTo:
 3     Transmission and
 4     (wasAssociatedWith some (hasAttribute some
 5       (FrequencyRange
 6       and (hasAttribute some
 7             (FrequencyMaximum and
 8               (hasValue some xsd:float[<= 1780.0f])))
 9       and (hasAttribute some
10             (FrequencyMinimum and
11               (hasValue some xsd:float[>= 1755.0f]))))))
12   SubClassOf:
13     Transmission
14
15 Class: US91-3
16   EquivalentTo:
17     US91 and
18     (wasAssociatedWith some JointTacticalRadioSystem)
19   SubClassOf:
20     US91
21
22 Class: US91-3.1
23   EquivalentTo:
24     US91-3 and (wasAssociatedWith some
25                 (atLocation some US91-3.1_Location))
26   SubClassOf:
27     Permit, US91-3
```

Listing 1.1. OWL expression of part of the US91 policy in Manchester syntax

Many of the NTIA policies, including US91, contain location rules and, often, these rules are written in terms of location lists. To represent this, we defined OWL classes for the lists to be used in conjunction with the OWL expression of policies. Listing 1.2 shows the definition of the US91-3.1_Location class, which is used in the US91-3.1 restriction, as previously shown. We have used the GeoSPARQL predicate sfWithin in conjunction with OWL unions to express that the rule is satisfied if the location is specified in the list (lines 3-8). In this example, we leverage location information we imported from Census.gov shapes for US Federal locations.

```
 1 Class: US91-3.1_Location
 2   EquivalentTo:
 3     (sfWithin value White_Sands_Missile_Range) or
 4     (sfWithin value Ft_Irwin) or
 5     (sfWithin value Yuma_Proving_Ground) or
 6     (sfWithin value Ft_Polk) or
 7     (sfWithin value Ft_Bragg) or
 8     (sfWithin value Ft_Hood)
 9   SubClassOf:
10     Location
```

Listing 1.2. OWL expression of a location list in Manchester syntax

The creation of the OWL class hierarchy (and, therefore, the incremental addition of rules) maximizes the reuse of rules when spectrum managers create local policies. To illustrate this claim, Listing 1.3 shows a sample local policy which modifies the existing US91-3 sub-policy to `Deny` requests in a specific time window. Lines 4-5 contain the time restrictions on the PROV-O predicates. Local policies can have an explicit precedence level, as shown in line 7.

```
1 Class: US91-3.1-Local
2   EquivalentTo:
3     US91-3.1 and
4     endedAtTime some xsd:dateTime[>=2019-10-01T11:00:00Z] and
5     startedAtTime some xsd:dateTime[<=2019-10-01T17:00:00Z]
6   SubClassOf:
7     Deny, Priority_1, US91-3.1
```

Listing 1.3. OWL expression of a local policy in Manchester syntax

3.4   Policy management

The DSA Policy Framework provides a web interface to allow spectrum managers to have a comprehensive understanding of the DSA Knowledge Graph, including policies, locations, and entities in the DSA Ontology. It leverages Whyis default "views" with some extensions for supporting a domain-specific display of pieces of the Knowledge Graph. The structure and content of the interface are driven by the DSA Knowledge Graph, which ensures that it displays relevant and contextualized information and features.

The *Policy Faceted Browser* that allows a user to quickly find policies based on the selection of attribute values. Spectrum managers can, for instance, find policies applicable to a list of select locations or policies applicable to a specific device, or even to a combination of both. This is accomplished by domain-specific SPARQL queries that retrieve and group attributes from the policy's OWL structure. The user can view details of a policy or reuse a policy's rules.

The *Policy Detail* view provides a display of policy metadata, including name, original text and identifier, and a human-readable version of the policy encoded rules. If the policy specifies locations, those locations will be displayed on a map.

The *Policy Builder* view can be used to build a policy from scratch or to create local policies by reusing existing policies' rules. The Policy Builder leverages the DSA Knowledge Graph to provide user support during policy creation. For instance, if a user wants to create a rule for an specific device, the Builder will display known devices as represented in the KG. More than that, the Builder "understands" the semantics of the rule which means that if the user wants to add a frequency range rule, for instance, the Builder will prompt the user to enter both lower and upper bound values. Users can also set policy effects, precedence, and obligations. In the back end, the policy is converted to the DSA policy model in OWL and stored as a new piece of knowledge in the DSA Knowledge Graph.

Knowledge curation is a time-consuming task and it might impact the pace at which updated knowledge becomes available for use when creating new policies.

To overcome this, the DSA Policy Framework supports policy additions that refer to terms not currently in the ontology, by allowing input of new terms along with the annotation that those terms need in order to be curated by the appropriate ontology owner. This allows a spectrum manager to input and test a new policy without having to wait for an ontology update first.

3.5   Request evaluation: Domain-specific reasoning

To enable the evaluation of spectrum requests against policies, we have implemented a domain-specific reasoner that combines various Semantic Web computational approaches to assign `Permit/Deny` effects to requests, while fulfilling requirements, including geographical reasoning, policy precedence evaluation, and explanations for denied requests. The domain-specific reasoner follows a four-phase pipeline, with a set of requests as input and the assigned effect, a list of obligations, and a list of reasons for each request as output. The reasoner initiates by creating an in-memory RDF graph originated by the merge of the request RDF graph and the DSA Knowledge Graph.

Next, in the **geographical reasoning phase**, the reasoner elicits the geographical relationships among the requests' WKT locations, and the named locations present in the DSA KG, by using GeoSPARQL to infer triples like `:req_location geo:sfWithin :NAMED_LOCATION`. The inferred triples are then asserted back into the graph.

In the **OWL reasoning phase**, the reasoner makes use of the HermiT OWL reasoner [14] to perform Description Logic (DL) reasoning over the updated graph. The domain-specific implementation relies on the DL services of:

– *Classification* for computing all subclass relationships, allowing the inferred class hierarchy to be leveraged when querying policy Effect and Precedence.
– *Realization* of computing classes (policies) that individuals (requests) belong to. "Belonging" to a class means that an individual satisfies the constraints of that class; realization can be understood as determining *policy applicability*.
– *DL Query* for retrieving the individuals that were determined to belong to specific classes.

Using the list of applicable policies retrieved from HermiT, the domain-specific reasoner decides precedence, in the **precedence evaluation phase**, by a simple evaluation of which policy has the highest precedence level. Policies with no explicit precedence are assumed to have the lowest precedence. The highest precedence policy effect is then assigned to the request. The reasoner follows these heuristics to explain the assignment of a `Deny` effect to a request, in the **explanation phase**:

– Requests can be denied by a specific applicable policy with a `Deny` effect. In this case, the "reason" for the denial is the identity of this policy and the specification of what attributes of the request fulfill the rules contained within. The reasoner retrieves the policy's rules and presents them as reasons for denial.

– Requests can be denied due to a lack of a policy with a `Permit` effect (i.e. there is no applicable policy with either a `Deny` or `Permit` effect). In these cases the reasoner finds the rules that the request did not satisfy in order to be assigned a `Permit`. The reasoner calculates the paths in the OWL class hierarchy from those policies that the request was reasoned to belong to, to the policies that would result in a `Permit`. These paths contain classes that the request was determined not to belong to. Unfulfilled rules for each policy found to be in the path are retrieved and presented as reasons for the `Deny`.

As as example, the request in Figure 3 would ultimately be determined to belong to the `US91`, `US91-3`, and `US91-3.1` classes displayed in Listing 1.1 and, therefore, assigned the `Permit` effect. Nevertheless, when the local policy shown in Listing 1.3 exists in the graph, the request is then reasoned to belong to the `US91-3.1-Local` class as well. Since this local policy contains a time constraint rule with a higher precedence, the reasoner assigns the effect to be a `Deny` and returns the reason, "the request is in a prohibited time window."

Conversely, if the request is modified to a different location outside the locations expressed in Listing 1.2, it would be reasoned to belong only to `US91` and `US91-3` classes, based on the frequency range and requester attributes of the request. These applicable classes don't express an *explicit* `Permit` or `Deny` effect, so the reasoner must default to `Deny`. The reasoner then calculates the path to the class `US91-3.1` that would `Permit` it *if* the conditions had been met, and retrieves the rules of each class in the path (in this case, only the rules of the `US91-3.1` class) and returns the reason, "the request is not in a permitted location."

4   Evaluation of the Framework

In order to evaluate the DSA Policy Framework's semantic representation of the domain, we identified several fundamental spectrum policy constructs. They are displayed in bold in the first column of Table 1. For each of them, we worked with domain experts to identify the elements that were required in order to effectively represent policies and support request evaluation. The table contains columns for Policy Representation (high-level and local policies) and Request Evaluation. "Yes" in the columns indicates that the policy construct is either *Relevant* or it has been fully addressed and *Implemented*. "Partial" indicates that the current implementation meets a simplified version of the requirement, while "uc" means that the construct element is currently under consideration.

The basic structure of a policy varies among source documents. The NTIA Redbook uses provisions to distinguish policy behavior with regards to attributes (specific device or location, for instance). Provisions are described as sub-policies in the policy capture spreadsheet and in the Framework, and they are leveraged during request evaluation. Policies can be specified in the *Parser logical rule* format, which is enabled in the Policy Builder. The framework does not currently represent the individual elements of an obligation. Instead, it is represented as text or as a canned identifier to support request evaluation.

Domain Policy Construct	Policy Representation		Request Evaluation	
	Relevant	Implemented	Relevant	Implemented
Provision	yes	yes	yes	yes
Parsed logical rule	yes	yes	no	no
Obligation	yes	partial	yes	partial
Requesters				
Device	yes	yes	yes	yes
Organization	yes	yes	yes	yes
Dependency	yes	uc	uc	uc
Licensee	yes	partial	yes	partial
Affiliations				
Federal/Non-Federal	yes	partial	yes	partial
Named requester	yes	yes	yes	yes
Frequencies				
Frequency range	yes	yes	yes	yes
Single frequency	yes	yes	yes	yes
Named bands	yes	yes	uc	uc
Units	yes	yes	yes	yes
Time				
Timezones	yes	yes	yes	yes
Policy validity	yes	yes	yes	yes
Locations				
Named locations	yes	yes	yes	yes
Relative locations	yes	uc	yes	uc
Polygons/Circles	yes	yes	yes	yes
Geographical rules				
Specific location	yes	yes	yes	yes
List of locations	yes	yes	yes	yes
Precedence				
Levels	yes	yes	yes	yes
Explanations				
Policy triggered	yes	yes	yes	yes
Rules not satisfied	yes	yes	yes	yes

Table 1. DSA Policy Framework policy semantics coverage

The text in the source policies specifies regulations for a variety of requester types. While all requester types are relevant for policy representation, requests only express the device. However, the DSA KG does represent several types including Organization and Licensee, so if a request was received for one of these types, theoretically, it could be evaluated. If a policy specifies a dependency between requesters, this dependency is only currently treated as text. Policies can also regulate the spectrum usage by affiliation (Federal systems are permitted to use some frequency, for instance). The Framework allows affiliation rules to be specified in policies, however, affiliation reasoning is currently limited by the expressiveness of affiliations in the DSA Ontology, where only requesters with an explicit affiliation (named requester) are effectively reasoned.

Frequency rules are specified in terms of a range or single frequency in different units (MHz, GHz). Sometimes, a range is described as a band, which is a named frequency range. Requests do not express named bands. The framework supports all of these constructs, using the `Frequency` and `FrequencyRange` attributes, and the `sio:hasUnit` property. The time attribute exists in local

policies only, and they specify the validity of a policy. Time and timezones are supported using `xsd:dateTime` literals and PROV-O time predicates.

Most policies refer to locations by names or by coordinates (points, polygons, and circles), but sometimes a location is expressed in relation to another location. The framework uses Census.gov shapes to refine named locations and WKT literals to represent polygons and circles. Relative locations are still under development. Geographical rules are defined in terms of the requester being in a location or in a list of locations. The framework implements all of these constructs using `geo:sfWithin` predicate and OWL unions.

The framework implements all of the identified precedence needs. It can define and evaluate precedence levels. For explaining evaluation results, the framework implements all current explanation requirements. It outputs which policy was triggered by a request and presents reasons based on the presented heuristics. Finally, the constructs identified as "uc" are areas for future work.

5   System Adoption and Deployment

The DSA Policy Framework is being used in simulated scenarios, where it supports the research & development of other components of a dynamic spectrum management system. It currently contains approximately 165 high-level policies from the NTIA Redbook (including their sub-policies). The DSA Ontology contains 695 classes and is constantly evolving to address new domain constructs and support more precise request evaluation.

The framework is transitioning to support live, over-the-air field exercises involving a diverse set of federal and commercial radios. During these exercises, the Framework supports (1) the creation, deletion, and revision of local policies, (2) the real-time processing of numerous spectrum requests, and (3) the generation of explanations that describe how the spectrum requests were processed. The public released assets developed during the course of the project can be accessed at `https://github.com/tetherless-world/dsa-open/`.

6   Related Work

Kirrane [17] offers a comprehensive survey of access control models, well-known policy languages, proposed frameworks that utilize ontologies and/or rules to express policies, and a categorization of policy languages and frameworks against access control requirements. XACML 3.0, the eXtensible Access Control Markup Language [1], is a well-known policy language and *de facto* standard for representing attribute-based access control (ABAC) [16] policies and requests. Importantly, XACML provides a reference architecture for centralizing access control and a process model for evaluating requests against existing policies that inform the design of access control systems across domains and technologies.

Thi [24] proposes an OWL-based extension to XACML to support a generalized context-aware role-based access control (RBAC) model providing spatio-

temporal restrictions and conforming with the NIST RBAC standard [9]. Their work augments the XACML architecture with new functions and data types.

Muppavarapu [21] identifies the limitations of identity-based access control schemes used in the Open Grid Services Architecture (OGSA) and proposes the use of OWL to represent the ontology of an organization's resources and users. They further propose the use of semantics in conjunction with the XACML standard for better interoperability and reduced administration overhead.

Our approach combines OWL, PROV-O, and the HermiT OWL reasoner with an ontology, represented as a knowledge graph, to support the representation of policies governing access to available spectrum. Relevant related research is described in Dundua [8], where previous work is described that uses OWL for modeling and analyzing access control policies, especially ABAC, and considers how the ABAC model can be integrated into ontology languages. In addition, Sharma [23] describes how OWL can be used to formally define and process security policies that can be captured using ABAC models. This work demonstrates how models, domains, data and security policies can be expressed in OWL and how a reasoner can be used to decide what is permitted.

Kolovski [18] maps the web service policy language, WS-Policy [6], to the description logic fragment species of OWL and demonstrates how standard OWL reasoners can check policy conformity and perform policy analysis tasks.

Garcìa [12, 11] and Finin [10] offer important contributions on how end-to-end usage rights and access control systems may be implemented in OWL and RDF. Garcìa proposes a "Copyright Ontology" based on OWL and RDF for expressing rights, representations that can be associated with media fragments in a web-scale "rights value change." Finin describes two ways to support standard RBAC models in OWL and discusses how their OWL implementations can be extended to model attribute-based RBAC or, more generally, ABAC.

Our policy representation approach builds on previous work by matching the cross-domain policy expression semantics of XACML. It extends these semantics with the capacity to express rich spatio-temporal restrictions, enabling the implementation of a wide variety of attribute-based policies across domains. It leverages background knowledge from domain-specific knowledge graphs that are structured with a domain-derived ontology, enabling the inference of policy applicability based on attributes and constraints. Our approach uniquely conceptualizes policy requests as PROV activities and request evaluations as realizations. Finally, our approach provides a novel reasoner-based explanation in request evaluation results, enabling domain policy developers to understand the precise reasons for policy decisions.

7   Conclusion

We described a policy Framework that leverages a machine-readable, radio spectrum policy representation to support policy management and enable a domain-specific reasoner to efficiently process spectrum requests. The DSA policy model uses OWL restrictions on PROV-O properties to represent policy rules in a hi-

erarchical approach that maximizes the reuse of rules when local policies are created, therefore facilitating the creation of local policies. The hierarchical nature of this policy representation also supports the explanation of evaluation results, by traversing the graph to find rules that were not satisfied. Because it leverages a domain Knowledge Graph, built from multiple knowledge sources, the domain-specific reasoner allows rich semantics which otherwise would be difficult to achieve with approaches that rely on a "flat" representation of attributes.

During the course of the project, we encountered OWL reasoning performance issues when multiple requests are simultaneously received. To decrease reasoning time, we partitioned the request graph into smaller graphs and evaluated each in parallel, using multiple processor cores. This allowed us to match the required response time of under 10 seconds.

Future work includes additional support for enforcement of the DSA policy model. Although the DSA policy model's OWL hierarchy maximizes the reuse of rules, there is currently no enforcement. Overlapping rules can be created, which can lead to multiple policies with the same precedence level being triggered during request evaluation. The DSA Ontology is constantly changing as additional policies are added to the framework with terms that have yet to be defined. Existing policies can be affected by changes in the DSA Ontology as their rules reference entities in it and they may need to be reviewed with regards to the updated representation of the domain. We plan to provide spectrum managers a way of tracking policies that are subject to review due to ontology changes.

References

1. eXtensible Access Control Markup Language (XACML) Version 3.0. `http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html`
2. ISO/IEC 13249-3:2016 Information technology — Database languages — SQL multimedia and application packages — Part 3: Spatial
3. IEEE Standard Definitions and Concepts for Dynamic Spectrum Access: Terminology Relating to Emerging Wireless Networks, System Functionality, and Spectrum Management. IEEE Std 1900.1-2008 pp. 1–62 (2008)
4. Federal Communications Commission, FCC 14-31. Tech. rep. (2014)
5. Standard Spectrum Resource Format (SSRF), Data Exchange Standard, Version 3.1.0 (MC4EB Pub 8). Tech. rep. (2014)
6. Curbera, F., Hallam-Baker, P., Hondo, V.M., Nadalin, A., Nagaratnam, N., Sharp, C.: Web services policy framework (ws-policy) (2006), `https://www.w3.org/Submission/WS-Policy/`
7. Dumontier, M., Baker, C.J., Baran, J., Callahan, A., Chepelev, L., Cruz-Toledo, J., Del Rio, N.R., Duck, G., Furlong, L.I., Keath, N., Klassen, D., McCusker, J.P., Queralt-Rosinach, N., Samwald, M., Villanueva-Rosales, N., Wilkinson, M.D., Hoehndorf, R.: The Semanticscience Integrated Ontology (SIO) for biomedical research and knowledge discovery. Journal of Biomedical Semantics **5**, 14 (2014)

8. Dundua, B., Rukhaia, M.: Towards Integrating Attribute-Based Access Control into Ontologies. In: 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON). pp. 1052–1056 (2019)
9. Ferraiolo, D.F., Kuhn, D.R.: Role-based access controls (2009)
10. Finin, T., Joshi, A., Kagal, L., Niu, J., Sandhu, R., Winsborough, W., Thuraisingham, B.: ROWLBAC: representing role based access control in OWL. In: Proceedings of the 13th ACM symposium on Access control models and technologies. pp. 73–82. SACMAT '08, Association for Computing Machinery, Estes Park, CO, USA (2008)
11. García, R., Castellà, D., Gil, R.: Semantic copyright management of media fragments. In: DATA 2013 - Proceedings of the 2nd International Conference on Data Technologies and Applications. pp. 230–237 (2013)
12. García, R., Gil, R., Delgado, J.: A web ontologies framework for digital rights management. Artificial Intelligence and Law **15**(2), 137–154 (2007)
13. Gkoutos, G.V., Schofield, P.N., Hoehndorf, R.: The Units Ontology: a tool for integrating units of measurement in science. Database **2012** (2012)
14. Glimm, B., Horrocks, I., Motik, B., Stoilos, G., Wang, Z.: HermiT: An OWL 2 Reasoner. Journal of Automated Reasoning **53**(3), 245–269 (2014)
15. Groth, P., Gibson, A., Velterop, J.: The anatomy of a nanopublication. Information Services & Use **30**(1-2), 51–56 (2010), publisher: IOS Press
16. Hu, V.C., Kuhn, D.R., Ferraiolo, D.F., Voas, J.: Attribute-based access control. Computer **48**(2), 85–88 (2015)
17. Kirrane, S., Mileo, A., Decker, S.: Access control and the Resource Description Framework: A survey. Semantic Web **8**(2), 311–352 (2017)
18. Kolovski, V., Parsia, B., Katz, Y., Hendler, J.: Representing Web Service Policies in OWL-DL. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) The Semantic Web – ISWC 2005. pp. 461–475. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2005)
19. McCusker, J.P., Chastain, K., Rashid, S., Norris, S., McGuinness, D.L.: SETLr: the semantic extract, transform, and load-r. Tech. Rep. e26476v1, PeerJ Inc. (2018)
20. McCusker, J., Rashid, S.M., Agu, N., Bennett, K.P., McGuinness, D.L.: The Whyis Knowledge Graph Framework in Action. In: International Semantic Web Conference (P&D/Industry/BlueSky) (2018)
21. Muppavarapu, V., Chung, S.M.: Semantic-Based Access Control for Grid Data Resources in Open Grid Services Architecture - Data Access and Integration (OGSA-DAI). In: 2008 20th IEEE International Conference on Tools with Artificial Intelligence. vol. 2, pp. 315–322 (2008), iSSN: 2375-0197
22. Perry, M., Herring, J.: OGC GeoSPARQL-A geographic query language for RDF data. OGC implementation standard **40** (2012)
23. Sharma, N.K., Joshi, A.: Representing Attribute Based Access Control Policies in OWL. In: 2016 IEEE Tenth International Conference on Semantic Computing (ICSC). pp. 333–336 (2016)
24. Tran Thi, Q.N., Dang, T.K.: X-STROWL: A generalized extension of XACML for context-aware spatio-temporal RBAC model with OWL. In: Seventh International Conference on Digital Information Management (ICDIM 2012). pp. 253–258 (2012)
25. Zhang, Y.: Dynamic Spectrum Access in Cognitive Radio Wireless Networks. In: 2008 IEEE International Conference on Communications. pp. 4927–4932 (2008), iSSN: 1938-1883
26. Zhao, Q., Sadler, B.M.: A Survey of Dynamic Spectrum Access. IEEE Signal Processing Magazine **24**(3), 79–89 (2007)