

Toward a Standard Rule Language for Semantic Integration of the DoD Enterprise

Suzette Stoutenburg, Leo Obrst, Deborah Nichols, Jason Peterson, Adrian Johnson
The MITRE Corporation
1155 Academy Park Loop, Colorado Springs, Colorado 80922
{suzette, lobrst, dlnichols, jasonp, abjohnson}@mitre.org

Abstract. Historically, the behavior of Department of Defense (DoD) Command and Control (C2) systems has been embedded in executable code, providing static functionality that is difficult to change. As the complexity and tempo of the world increase, C2 systems must move to a new paradigm that supports the ability to dynamically modify system behavior in complex, changing environments. Separation of rules from executable code provides the foundation for dynamic system behavior and agile response to outside events. A Rule Language Standard is required to realize the full benefits of rule separation including the sharing of rule abstractions across disparate domains, thus enabling interoperability across the enterprise.

As engineers and researchers at MITRE, a Federally Funded Research and Development Center (FFRDC) for the DoD, we have launched a three-year effort to identify DoD requirements that need to be addressed by the Rule layer of the Semantic Web. We believe our research has implications for a broad audience interested in rules for interoperability. We are investigating the interaction between the rule and ontology layers of the Semantic Web to determine how a standard language should best express each for interaction. In particular, we are examining specification, translation and execution of the Semantic Web Rule Language (SWRL) language vs. Web Ontology Language (OWL)+RuleML to determine if the ontology and rule layers are best combined or left separate. We are focused on exploring orchestration of inferencing across layers this year. In future years, our research will address adaptable policy enforcement using ontologies enhanced with rules, dynamic rule distribution, ontological closure and rule annotation for discovery and reuse. We will demonstrate how rules can be used for agile management of information flows in complex, dynamic C2 environments. Throughout this three year effort, we continue to look ahead to anticipate future requirements for the emerging standard. We will share our research plan and preliminary findings at the workshop.

1 Purpose

This paper presents the plan and initial findings of the MITRE Sponsored Research effort, **Toward a Standard Rule Language for Semantic Enterprise Integration**. This is a three year effort to investigate the interaction between the rule and ontology layers of Semantic Web languages to determine how a standard language should best address the needs of our DoD sponsors. To date, this effort has identified a number of compelling DoD use cases and requirements for a standard rule framework. We are examining issues such as orchestration of inferencing across layers and adaptable policy enforcement using ontologies enhanced with rules. We will determine how rules can be used for agile management of information flows in complex, dynamic C2 environments, allowing identification of DoD specific requirements for the evolving Rule language standard. In later years, we will explore dynamic rule distribution, ontological closure and rule annotation for discovery and reuse. Finally, we will look ahead to anticipate future requirements for the emerging standard.

2 Motivation: Potential DoD Use Cases

The behavior of C2 applications has traditionally been embedded in executable code, offering static functionality and slow response to changing real world events. As the world becomes ever more complex, C2 systems need to move to a new paradigm in which system behavior can be dynamic, agile and easy to change. Separation of rules from executable code supports the ability to dynamically modify system behavior in complex, changing environments, allowing agile response to outside events. In this section, we identify DoD use cases to which a standard rule framework could be applied.

2.1 Dynamic Information Sharing and Mediation

Managing information flows across a Coalition Battlefield is a complex problem. Diverse sources provide C2 information in different formats and languages, with varying security levels and disparate metrics. Rules can be used to facilitate dynamic information sharing in several ways. First, information sharing rules can be applied to enforce security policy as information is exchanged across multiple security boundaries. Next, rules can be applied for syntactic and semantic mediation between sources. This is

especially important when exchanging information across multiple organizations, especially coalition partners. In these cases, contextual differences must be considered and mediated. For example, altitude may be measured from the center or surface of the earth and may be measured in meters or miles. By applying rules as information is exchanged, translations and transformations can occur automatically. Rules could also be applied for discovery during the sharing process. Since coalition partners may contribute different capabilities to a theatre, multiple facts could be exchanged and reasoned over, resulting in enhanced situational awareness for all partners. Also, information sharing rules based on usage and capability of the receiving source can ensure that a particular node in the theatre receives only that information that it can process and display in a meaningful way. Finally, information sharing rules can be based on periodicity or events, so that updates will be sent to multiple partners on a dynamic basis.

2.2 Rapid Enterprise Integration and Reuse

When a new set of C2 requirements is identified or a new C2 system is built, functions tend to overlap, and are often “re-invented”, greatly slowing the fielding of new mission capabilities. Integration of new nodes into the Enterprise must occur more rapidly. With the separation of rules from execution engines, new nodes can be more rapidly integrated into the Enterprise, since rules can be swapped and reused, as can generic engines that perform a specific function. For example, Figure 1 shows how rules developed for classification and characterization could be reused to meet other program needs. Also, a generic Classification Engine could also be reused by other nodes in the Enterprise.

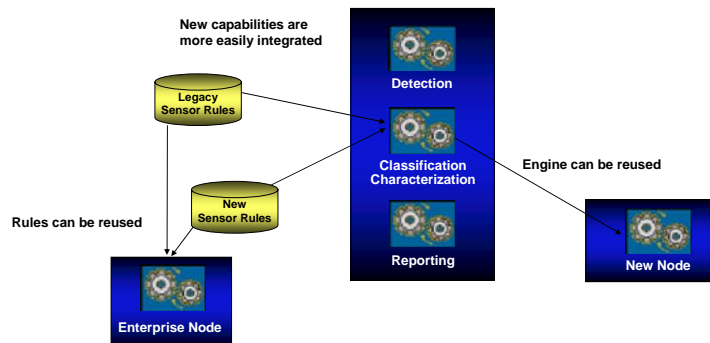


Figure 1. Rules for rapid enterprise integration

2.3 Dynamic Service Oriented Architectures

Rules can be applied to achieve *dynamic* Service Oriented Architectures, in which web services could behave differently under varying circumstances. Service behavior could be structured according to belief, desire and intent rule filters that provide a specific context for the interpretation of service semantics (expressed in both supporting ontologies and rules). As the needs of the enterprise change, rules of engagement can be modified to handle new situations and provide new results, as show in Figure 2 below.

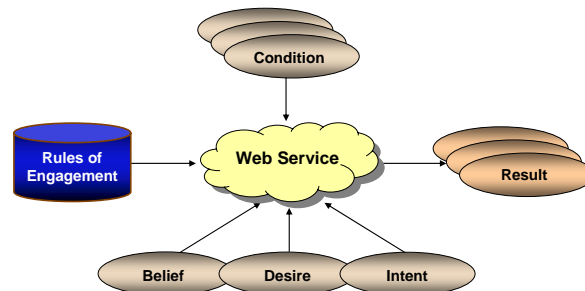


Figure 2. Rules for dynamic service oriented architectures

2.4 Complex Semantic Integration

A standard rule framework will also support complex semantic integration since rules can expand on the semantics captured in ontologies. While semantic web ontology languages specify some semantic context,

rule languages can apply implication and logical consequence rules for more powerful and detailed specification of semantics. Figure 3 shows an example of complex semantic integration in which disparate information across multiple enterprises could be managed by the orchestration of synthesis and semantic rules.

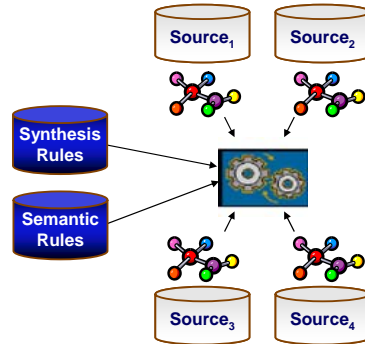


Figure 3. Rules for complex semantic integration

2.5 Machine to Machine (M2M) Interactions

A standard rule framework built to operate over ontologies can enable machine to machine interfaces in a number of DoD environments. Dynamic C2 systems could employ M2M interactions for “asset allocation” of battlefield capabilities. For example, Unmanned Aerial Vehicles (UAV) could be dynamically tasked for reconnaissance and surveillance. Requests for backup and troop reallocation could also be automatically triggered during times of distress. Requests for clarifying information between components could be exchanged by applications while monitoring battlefield events, resulting in richer alerts and recommendations to the Warfighter.

2.6 Computer Vision

Computer vision provides a rich domain for rule and ontology application. There is an increasing need for the capability to process real time streaming video and derive meaningful information for use in situational awareness. By comparing successive images using complex rules, visual information can be derived and integrated into a comprehensive knowledge base. Rules for adaptive visualization of the battlespace is another example of where to apply a standard rule framework.

2.7 Current Research Focus

Our research is focused on a scenario in which fusion of data using ontologies and rules contributes to enhanced situational awareness. We show how Machine to Machine (M2M) capabilities can be provided by reaction rules. Our initial-year experiment models a supply convoy moving through an unsecured area. The convoy has access to reports of moving objects in and around its route as well as previously reported intelligence information. A new report of an unknown moving object near the convoy triggers rules to interpret the operational situation combined with the intelligence information. Further application of reaction rules enables a UAV to be automatically deployed for a closer look at the unknown object. The results of the UAV observation are integrated into the growing knowledge base, resulting in further recommendations to the convoy commander.

3 Research Plan

Though we see advantages to segregating ontologies and rules, our hypothesis is that rules should be captured in a single integrated language with ontologies. To determine if this is true, we are building an experiment to compare an integrated versus segregated approach to applying ontologies and rules. For this experiment, we have selected the Convoy mission use case described above. This example will require a rich ontology with a broad set of well-defined rules of different types.

To capture this use case, we are specifying the ontology and rules in two ways: first, using SWRL as the integrated language and second, using OWL + RuleML 0.87 for the segregated approach. We are translating each into an executable language, Prolog, then executing to compare results. We plan to measure the expressivity of each approach in terms of effectiveness, efficiency, difficulty, and degree of

representational impedance. We will evaluate translatability for each approach. Finally we will evaluate the suitability for deployment by comparing complexity, flexibility, determinism and performance of each approach.

The rule base, rule engine, and code will remain constant in both the integrated and segregated approaches and only the languages in which the rules are expressed will be varied. Also, we intend to explore the full range of rule types. We will consider the impact of rule types on the ontologies (assertions about concepts) and the knowledge bases (assertions about instances of those concepts) referenced by the rules.

4 Early Findings

Our research, still in progress, has yielded some interesting early findings. Though related to DoD requirements for a standard rule language, these findings may be significant for other uses of rules for interoperability of systems.

4.1 Uncertainty

First, very little is definitive in the world that DoD applications support. The need to express levels of confidence for facts and relationships is paramount. Uncertainty needs to be expressed about facts and class membership of objects, and perhaps even about relationships and properties. Further, to support the vision of rule discovery in a semantic network, a standard way of expressing uncertainty is required. We do not necessarily suggest that the ontology or rule standard support uncertainty; at the very least, however, a standardized “plug-in” will be needed.

4.2 Dynamic Classification

With regard to orchestration of the ontology and rule layers, we anticipate issues in handling dynamic classification. Clearly, classification axioms and rules should be executed first; but what about dynamic classification? How do we ensure that dynamic classification triggers the firing of rules already executed? We can't be sure that the executable language will trigger all rules until exhausted (i.e., a forward chaining or backward chaining system cannot be assumed) since the standard rule framework must be translatable to any language, including Java-like languages.

4.3 Translation

Two kinds of translation problems emerged, some originating in conceptual differences between OWL and Prolog, others in using XSLT as our translation tool. We have had some difficulty in translating OWL constructs to Prolog, our runtime engine. Description logics in general do not map directly to the Horn subset of First Order Logic employed in logic programming, as [1, 2, 3] describe, which has led to the emerging research program called Description Logic Programming. In general, equivalence assertions can be problematic, since equality in the head of the resulting Prolog rule may be necessary, which violates the Horn rule specification. Also, no existential quantifications can occur in the head of a rule. Additionally, most Prologs use “negation by finite failure” and the Closed World Assumption (the knowledge base is fixed and closed at any given time) rather than true logical negation. Cardinality restrictions therefore are difficult to enforce over a Prolog knowledge base, since equality, existentials, and negation may be necessary. Applying inheritance downward across all classes also results in complex Prolog rules. We plan to optimize the inheritance of properties down the transitive closure of classes by computing the closures only when necessary and caching the result, work that is still in progress. Similarly, we are investigating using complex structures in effect as tables to capture individual classes and their properties. Since we are currently using XSLT as our mechanism to translate from input OWL files to output Prolog files, we have found the need for multiple passes over the OWL source for optimization. A future consideration is to implement the translations in Java, Perl, or within Prolog itself. Finally, we are primarily using backward-chaining, the typical reasoning style in Prolog, but are investigating using a limited forward-chaining mechanism for taxonomic reasoning and the handling of some dynamic rules, the latter of which may require new inferences to be drawn.

4.4 Ontology Design

Ontologies are powerful and extensible when complete, but specification is hard. We have followed a hybrid approach, combining a “bottom up” method that bases concepts on existing data sources, with a “top down” approach in which we design against a more general representation of the domain to capture key concepts beyond current applications. We have found that to specify ontologies, knowledge of the operational environment and underlying database structures is necessary, along with an understanding of

what is possible with ontologies. Therefore, an operational expert with vision, database experts, and an ontologist are the minimum team needed to develop a robust semantic model for a C2 use case. It can be difficult to assemble such a trained set of experts.

4.5 Tools

Finally, for DoD to embrace a standard rule language, we believe that an integrated yet interoperable framework of tools, languages and standards needs to evolve. This framework must support specification of ontologies and rules, validation, translation to executable environments, and execution. It must also be extensible to allow for integration of future layers of the Semantic Web. Integrated engines which operate over ontology and rules are also required, to allow for “one stop” querying from the executable engine.

5 Interesting Questions

Below we identify a list of questions that we view as unanswered as of yet. We hope to contribute to answers in our continuing research.

- What are the advantages of an integrated vs. segregated approach for specification of ontology and rules?
- Should rules be combined with ontologies into a single standard language?
- If a single language is best, then how should it be structured to express different rule types and different logics?
- If a layered approach is preferable in which rules and ontologies are separate, then how should these layers interact? Can the rules be separated from the concepts? If so, how?
- How should inferencing best be orchestrated?
- Would the use of metadata about the ontologies and rules help in automating the orchestration process?
- Do the approaches lead to deterministic systems?
- Are there certain properties of ontologies that will make them more or less tractable with certain execution environments?
- What are the rule types needed for DoD applications, and how should they be layered?
- How do we account for the mechanics of executable languages?
- Should OWL, RuleML and SWRL be translated to executable environments? Or should integrated SWRL engines be developed and integrated into DoD systems?

6 Conclusion

MITRE has initiated a three year effort to explore the interaction between the rule and ontology layers to determine how a standard language should best address the needs of our DoD sponsors. To date, this effort has identified a number of compelling DoD use cases and requirements for a standard rule framework. In this paper we presented some of these use cases and provided initial findings regarding DoD requirements for a standard rule language. C2 applications will need a standard method of expressing uncertainty, and dynamic classification may need to be managed differently, depending on the execution environment. Development of ontologies requires a team of operational and technical experts; training for such expertise should be expanded in the DoD. Finally, there is a need for integrated, robust tools to support ontology and rule specification, translation and execution.

In order for DoD to embrace these technologies, a number of compelling questions remain to be answered. We identified some of these in this paper and hope to contribute to answers in our continuing research.

7 References

- [1] Grosz, Benjamin N.; Ian Horrocks; Raphael Volz; Stefan Decker. 2003. Description Logic Programs: Combining Logic Programs with Description Logic, WWW2003, May 20–24, 2003, Budapest, Hungary.
- [2] Volz, Raphael; Stefan Decker; Daniel Oberle. 2003. Bubo - Implementing OWL in Rule-Based Systems, WWW2003, May 20–24, 2003, Budapest, Hungary.
- [3] Volz, Raphael. 2004. Web Ontology Reasoning with Logic Databases, PhD dissertation, University of Karlsruhe.