Responsive Decision Making through Automated Policy-Enabled Systems

Anne-Marie Buibish
Amy Lange
Michael Woitalla

Raytheon Company
Network Centric Systems
1010 Production Road
Fort Wayne, IN 46808-4106
260-429-6624

1 Abstract

Policy is an essential part of military decision-making. The use of policy is growing in both importance and complexity. Some examples of military policy are Commander's Guidance, Rules of Engagement, security restrictions, legal and international policy. Commanders use policy to allow or restrict missions, pair weapons with targets, allocate airspace, and request coordination, among others. A policy-enabled system is one that can make decisions based on policy and control the decision-making behavior of the system. In its most general form policy defines when an action should be taken, or not taken, based on a specified condition or criteria. Present day policy-enabled systems embed the policy-related decisions and procedural software code within the business logic of the application. This approach is sufficient when the policy set is fairly simple and static. However, this approach does not scale well to increasingly complex and dynamic policy environments, such as those in the military battlespace. The increased operational tempo, uncertainty of the situation, and fluidity of the contemporary operational environment influence this complexity.

For these reasons, policy-based management of automated systems has become an active area of research. Central to this research is to capture domain knowledge in the form of an ontology, [a machine-interpretable definition of objects or concepts], the relationships between them, and the constraints and axioms about those relationships. Ontologies define more than syntax and structure; they describe the semantic information needed to reason and infer. Ontologies coupled with rule-based programming become a powerful technique to achieve an adaptable and extensible solution. Based on the cited research and military policy examples, this paper will demonstrate how such extensibility and adaptability can be achieved.

Additionally, this approach lays the foundation for the evolving network centric battle environment. Furthermore, it does not preclude the use of agent-based learning systems to further enhance automated decision-making combining these technologies.

2 Motivation

The need to adapt and survive drives the military battlespace to evolve very quickly. For example, during a course of operations, conditions may create opportunities to utilize weapons and munitions in new ways to meet emerging threats. The following operational scenario illustrates this point.

For a moment examine a scenario where a series of mortar attacks plagued a Division's Main Command Post. The counter-mortar radars assigned to monitor the area were able to acquire the mortar rounds and provide a point of origin. This point of origin was always in a built-up area or surrounded by protected areas. The Division was able to direct collection assets to confirm the mortar team locations and reduce the Target Location Error (TLE) within acceptable standards. However, it was unable to dispatch forces quickly enough to engage the mortar team before they left the area and evaded aerial surveillance.

Commander's Guidance (policy) in this situation dictates that fires into a built-up area must have a TLE of less than 10 meters and be engaged by a precision-guided munition. Additionally, the Rules Of Engagement (ROE) dictate that collateral damage be minimized.

All available indirect fire assets were determined to be incapable of satisfying the ROE because they were not precision-guided and did not possess the required minimum safe distance, increasing the risk of collateral damage. To further frustrate the situation, any appropriate air support options would take too long from request to delivery to be effective against this mobile target.

However, the team had an ample supply of an 155-mm artillery round called Copperhead. This round is an artillery fired, laser guided munition, with a small minimum safe distance. It is employed against light armored vehicles and tanks with a terminally guided laser capability.

The system did not recommend Copperhead for employment due to its typical engagement profile. Due to the specialized nature of the Copperhead munition, the business logic intentionally precluded the selection of Copperhead unless specifically requested by the operator. So even though the Copperhead had all the characteristics needed to engage the target, it was not automatically recommended to the team.

Although the system overlooked recommending the Copperhead, Commanders realized that they could leverage this munition to satisfy immediate strike needs against buildings, vehicles and other emerging targets on the complex battlefield. Unfortunately, they couldn't adapt the system to recognize this newly realized relationship and so be able to provide this as a recommendation to the operators. In this sense, the system proved inflexible.

With ontologies and policy driven decision-making, this new relationship could have been modeled in the knowledge base as a munition that possessed not only anti-armor capability, but also precision guidance and small minimum safe distance characteristics. Copperhead would then have been recommended in the scenario. Even if not originally modeled in the knowledge base, it would be far easier to update this knowledge base than to update and test new, added software code.

Beyond this scenario the changing landscape of the battlefield combined with geo-political factors require systems to be flexible and adaptable. To satisfy this dynamic environment, a policy-enabled system needs to be developed that can handle increasing complexity without lengthy software development cycles, and can automatically (without changes to the procedural logic) interpret and act on new relationships.

3 Current Force Policy

Policy is an important factor in the current military domain. It enables emerging opportunities to be identified (for example, new targets, new weapon capabilities), and enables the commander to influence the automated decision behavior of the system. It also allows the commander to ensure that the changing rules and regulations are considered before every mission. Also, a policy-enabled system handles mundane tasks, allowing the commander to manage by exception.

In its most general form policy defines when an action should be taken, or not taken, based on a specified condition or criteria. In the military battlespace domain, policy is known as Commander's Guidance (CG), Rules of Engagement, and Course of Action. Other types of policy include security management, legal, and international policy.

The Advanced Field Artillery Tactical Data System (AFATDS) is a policy-enabled system currently deployed with the Army and Marine Corps. It has a rich set of Commander's Guidance to allow the commander to influence the automated decision-making behavior. A few examples of CG are:

- Attack Methods Tables - allows the commander to specify the type of munition and fuze combination to be used for a specified target type
- Munition Restrictions – allows the commander to indicate a specific type of munition that must not be used for a specified target type
- Intervention Criteria - allows the commander to specify that the selection be first approved through the commander for a given mission type or selected fire unit
- Target Selection Criteria - guides the process to determine if a target nomination should result in a target
- Alerts – triggers and tripwires

In addition, AFATDS allows for policy combinations that enable the commander to define a complex policy. For example, "restrict High Explosive (HE) when a personnel target is within the Named Area of Interest AB300." Collectively, these policies are used to influence the automated selection of weapon-target pairs and intervention display behavior of the system.

Despite the automated application of policy, manual application of policy is still prevalent. During Operation Iraqi Freedom, it was evident that legal policy had become an increasingly important factor in effects employment. Legal and international policy is in constant flux; a valid target in today's battle may be an illegal target in tomorrow's battle. To address this challenge a legal team works with the commander and participates in the approval process for each mission request. Jointly they apply the legal and international policy for each mission manually, because the automated systems have not been designed to make those types of decisions and cannot be easily modified to do so. These factors reduce the mission throughput and commander's effectiveness. It has become clear that the rapid evolution of policy cannot be accommodated by today's automated decision-making systems.

4 Current State-of-the-Technology

Interest in policy is growing in many technical communities. In the area of networking, research is being done to more easily apply policy to enhance system performance. New areas of research, such as, pervasive computing, are looking to policy-enabled techniques to realize their vision of seamless semantic interoperability. In the military domain, forms of automated policy have been employed for many years, however, current systems cannot adapt to the dynamic battlespace with the current, more static implementations of automated policy.

Table 1 outlines a few key policy needs in the Network Community and their corollaries in the Pervasive Computing and Military Battlespace domains.

Network Community	**Pervasive Computing**	**Military Battlespace**
Preferential Service to select few	Prevent access to preserve privacy	Preferred Target or Resource
Simplified device and service management	Simplified device and resource management	Simplified resource management
Fewer Engineers for configuration	Limited operator configuration	Fewer Operators for operation
Define behavior of network	Define interactions of systems	Define behavior of systems
Manage increased complexity of programming devices	Manage dynamic resource availability	Manage increased complexity of resources

Table 1 – Comparison of Domain Concerns

4.1 Network Community

Policy-Based Network Management (PBNM) has been used since the 1990s and was initially characterized as Quality of Service management. The network community spearheaded the research on policy-enabled systems to specify network protocol. More recently, PBNM evolved into an approach for handling business rules and applying business processes. The networking community formed a working group, developed a policy model, and published a draft of the first policy model for the Internet Engineering Task Force.

A goal of this working group was to generalize the policy model such that it could be applied to a variety of domains. Unfortunately, this policy model contains network-specific language.

Strassner (Strassner, 2004) has taken the network policy model as a basis, and generalized the terminology. The resultant model can now be applied to many other domains, including the military domain.

The key definitions of this model are as follows (Strassner, 2004):

Policy-Based Management *is the usage of policy rules to manage the configuration and behavior of one or more entities.*
Policy *is a set of rules that are used to manage and control the changing and/or maintaining of the state of one or more managed objects.*
Policy Rule *contains data that define how the Policy Rule is used in a managed environment, as well as, a specification of behavior that dictates how the managed entities that it applies to will interact.*
Policy Condition *defines the necessary state and/or prerequisites that define whether the associated Policy Actions should be performed.*
Policy Action *represents the necessary action that should be performed if the Policy Condition is met.*

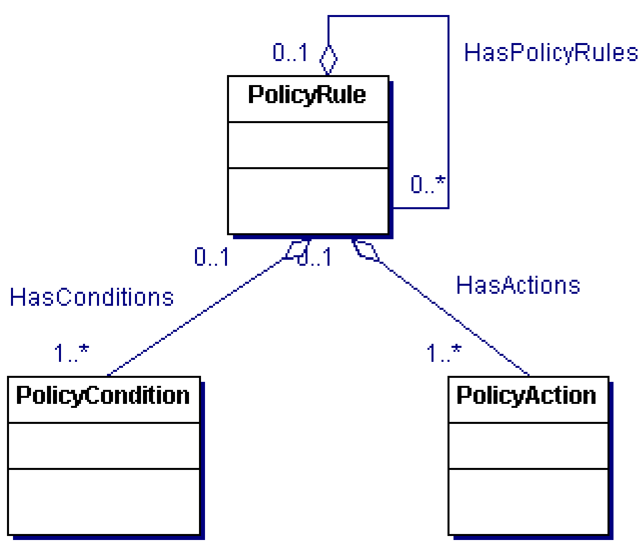

Figure 1 Policy Model – Primary Classes

4.2 Pervasive Computing

Pervasive computing is another current area of research that has embraced automated policy to overcome the challenges of security and privacy in this open and dynamic environment. Researchers at the University of Maryland, eBiquity Lab, envision a pervasive computing environment where computer systems will seamlessly integrate to share knowledge, reason about their environment, and interoperate. Key to this vision is a dynamic and flexible policy mechanism. The Standard Ontology for Ubiquitous and Pervasive Applications (SOUPA) project (Chen et al, 2004) has defined an ontology and architecture that allows a user to specify high-level rules that control low-level system behaviors. The Me-Centric Project (Kagal, 2002) has applied and demonstrated the SOUPA approach to solve a printer resource allocation example.

As a result of the current successes and promise of flexibility and extensibility, more research is being done to create a generalized definition and architecture for policy-enabled systems. The Rei Ontology Specification (Kagal, 2002) is one such research project that has abstracted policy to a level where it can be applied to many different domains, including the military domain.

4.3 Current Military Applications

Today's military automated policy systems provide some useful tools when dealing with weapon systems and battlefield conditions that are known well in advance. Operators define rules that allow the system to make recommendations based on the operator specified criteria. In this way, the operator delegates to the automated system routine decisions that are highly deterministic. This results in faster and more consistent decision-making.

AFATDS has employed automated policy-based decision-making since its inception in the 1980s. However, new approaches to automated policy have become an area of interest as users discover the increasingly complex and dynamic nature of the policy environment.

The technical approach for current military automated policy in systems, such as AFATDS, is simplistic and static. The policy definition, policy relationships, policy model, and execution points are all embedded in tightly coupled business logic. For example, the policy rule "restrict HE when the target is of type personnel" requires a highly coordinated development effort between the user interface used to define the rule, the database structure used to store the rule, and the decision logic used to process and respond to the rule. Any changes to add a more complex rule or to derive policy beyond what was originally designed, results in a ripple effect across the system, adding lengthy development time and risk. Adding a new munition or weapon, for example, is not easily accommodated, and must often wait until the next product development cycle.

A simple policy, for example, "Use least expensive munition when selecting munitions" can be coded into the software logic by simply comparing the cost of each capable munition during the selection process. However, as the rules become more complex, the procedural logic also gets very complex, increasing the risk of programming error or solution infeasibility. A more complicated rule, such as, "Use the least expensive munition, unless a Navy asset is available and the Navy asset has sufficient inventory to support its planned missions" would result in very complex procedural logic.

Another problem is that the relationships between managed entities must be explicitly stated in the procedural logic. For example, if a rule states that all rocket missions must be approved by the Air Operations Center, a relationship must be defined between each rocket model and each rocket system. This relationship is made explicit in the procedural logic, and must therefore be defined in the software code before a relationship would be automatically understood by the system. Changes to this relationship, therefore, cannot be made ad hoc as new relationships are discovered.

A complicating issue is that the embedded nature of today's policy systems does not allow for knowledge sharing or reuse. Each application must actualize policy as a very specific implementation to that application. Embedded policy logic for coordination of fires, for example, cannot be directly applied to allocation of airspace, even though the essential elements of policy (event, condition, action) are identical.

5 Automated Policy Solution for Net-Centric Warfare

The Department of Defense is transitioning to a Net-Centric warfighting environment. Future Net-Centric Military Applications must be flexible and adaptable to recognize and utilize battlefield systems that are dynamically entering and exiting the theater. Many systems currently in development, such as, Future Combat Systems (FCS), Distributed Common Ground System (DCGS), DD(X), among others, have recognized this and identified the need for a sophisticated policy application.

5.1 New Approach

The proposed policy system architecture allows an operational expert to define knowledge about the domain. The commander defines policy rules for use in conjunction with the domain knowledge. The policy consumer can retrieve policy rules applicable to a set of criteria. This architecture separates policy from its implementation by such that it is possible to change policy dynamically without the extensive ripple effect through these key components.

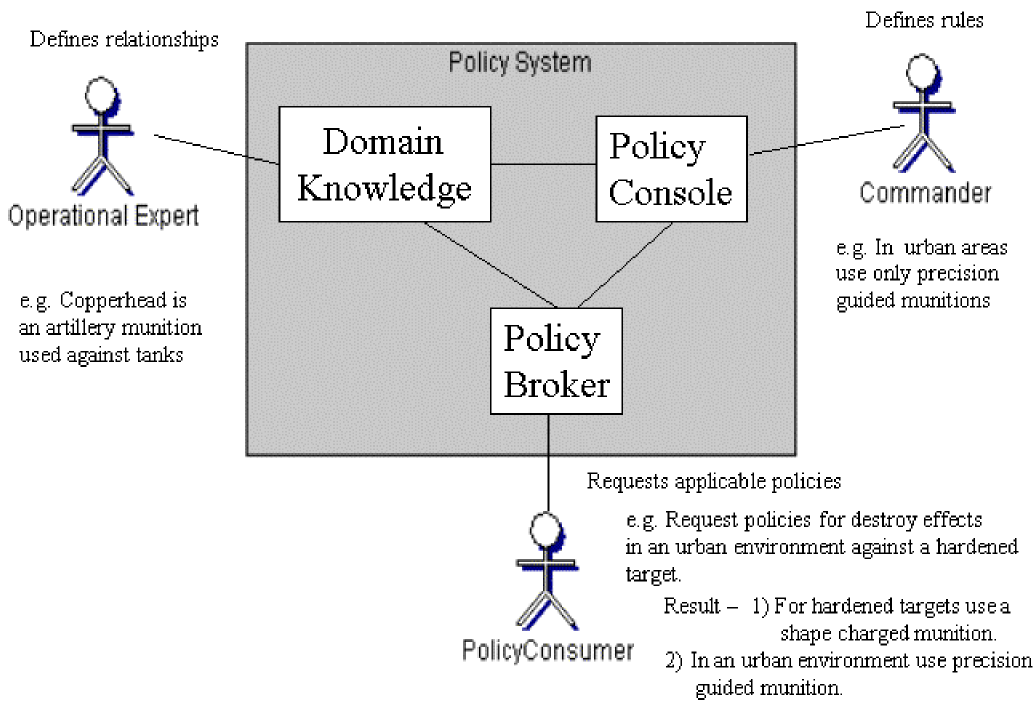

Figure 2 Policy System

5.1.1 Domain Knowledge

In the Policy System, Domain Knowledge is a formalized and machine interpretable representation of the problem space. It is captured directly from the operational expert and includes entities as well as relationships within the domain.

The Domain Knowledge is used by the Policy Console to constrain the types of rules that may be defined. It is also used by the Policy Broker to help determine applicable rules based on the relationships defined in the Domain Knowledge. The Policy Consumer may also use this knowledge to assist in developing solutions as governed by the relevant rules. As we learn about the environment we can improve the Domain Knowledge to better represent the relationships and rules, resulting in better solutions utilizing this newly acquired knowledge.

5.1.2 Policy Console

The Policy Console captures the commander's intent in the form of policy rules. This is typically a user interface, however, other types of input are possible. The commander defines the rules, which consist of conditions and actions. The type and complexity of the rules are constrained by the complexity of the domain knowledge.

A simple domain representation will allow for only simple rules, whereas, a complex domain can accommodate complex rules.

5.1.3 Policy Broker

The Policy Broker is an expert system that stores the policy rules as defined through the Policy Console. The Domain Knowledge is used to relate rules to the request criteria received from the Policy Consumer. For example, a request for rules applicable to the criteria of a fire mission with Mortar Team target type in an urban environment would return all rules that are directly or indirectly associated with that criteria. Indirect associations are defined through the domain knowledge. For example, the commander has defined a rule "The preferred munition for Fire Support (FS) type targets is Improved Conventional Munitions (ICM)." Using the domain knowledge, which relates a Mortar Team as a specialization of FS target, this rule is returned, even though the rule does not explicitly state the Mortar Team target type. Additionally, the rule "Precision Guidance System is required for targets in an urban environment."

It is possible for the Policy Broker to return rules that are in conflict. Policy conflict detection and resolution is a current area of research, where strategies are being developed to address this problem.

5.1.4 Policy Consumer

The Policy Consumer interacts with the Policy System by making requests for a given set of criteria and acting on the response. The Policy Consumer is responsible for both the Policy Decision and Execution Points (Strassner, 2004). The Policy Consumer may exist in a variety of forms:

1. Expert System – this consumer uses the rules and the domain knowledge within its own knowledge base to reason and infer solutions
2. Conventional Software System – this consumer applies the rules within its pre-programmed understanding of the domain
3. Human – this consumer simply views the rules and manually applies the policy

A Policy Consumer in the form of an expert system provides the most flexibility because decisions can automatically adjust to the changing domain knowledge. While conventional software systems may only act upon pre-programmed relationships, transforming domain knowledge into internal system knowledge can augment the decision-making ability of the Policy Consumer.

5.2 Key Technologies

The key technologies to implement this architecture for flexibility and scalability are an ontology and an expert system.

Ontologies are used to capture domain knowledge. An ontology is a specification of a conceptualization (Gruber, 2003). In other words, it is a formal description of concepts, relationships, constraints, and axioms that exist for a specified domain. An ontology defines a common vocabulary along with the semantics, and is in a machine-interpretable form to enable software agents to reason about them. It explicitly states assumptions by clearly defining relationships between entities. An ontology has the advantage of separating the domain knowledge from the implementation, such that operational experts are able to define the ontology, without having programming expertise (Noy and McGuinness 2001).

If rules are stored in the knowledge base about the FS target type and a request for a mission against a Mortar Team is received, a software agent using the relationships defined in the ontology can infer that Mortar Team is a specialization of FS target, and thus retrieve and apply all rules concerning FS type targets.

The ontology is used in several ways:

1. Policy and guidance are defined as a set of rules within the scope of the ontology representing the domain knowledge.
2. A user interface may use the ontology to prompt for available selections for defining rules.
3. The Policy Consumer uses the ontology to understand relationships between objects in the domain.

Ontologies, however, cannot work alone. When combined with reasoning systems, such as a rule engine, an ontology provides the relationships, constraints, and axioms necessary to reason over the data (Davies et al, 2003). As defined in "JESS In Action (Friedman-Hill, 2003)" a rule-based system is a computer program that uses rules to reach conclusions from a set of premises. Rule-based programming, also known as declarative programming, is well suited for the challenges of the automated policy system. Rules describe what the computer should do, rather than how to do it. The two basic components of a rule-based system are:

- Inference Engine – controls the process of applying rules to data
- Rule Base – contains all rules within the system

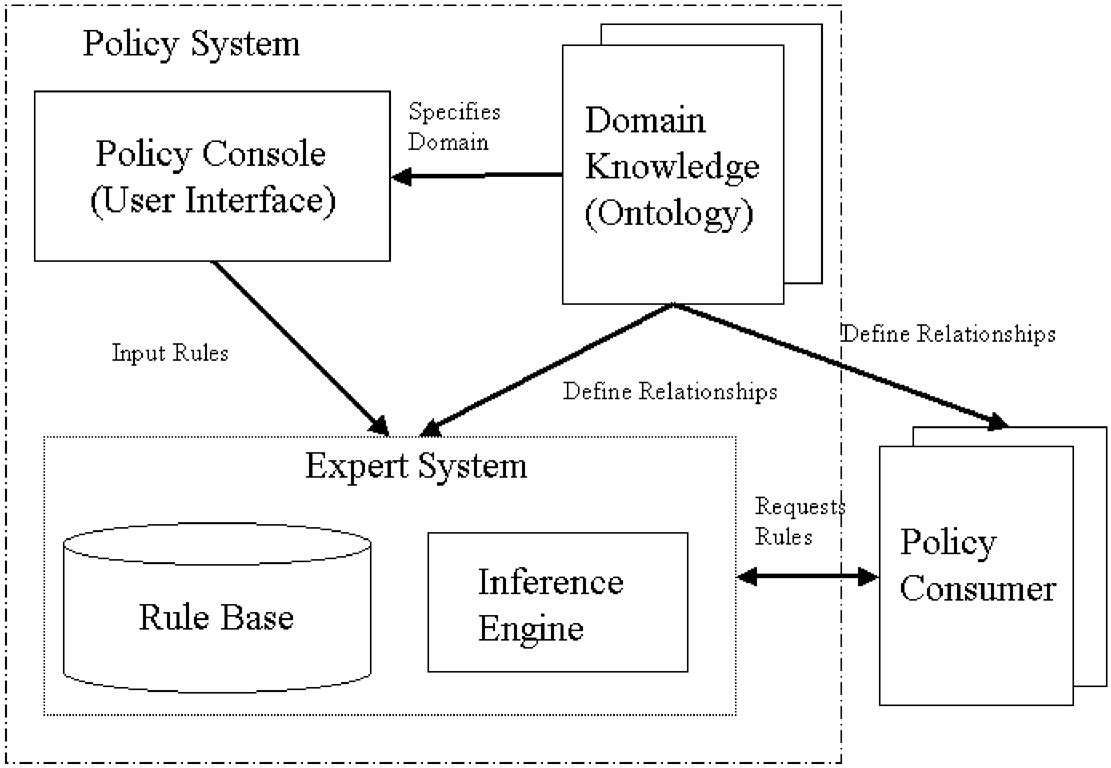

Figure 3 General Policy Architecture

5.3 Example

Returning to our scenario, the example ontology in Figure 4 contains the initial relationships defined in our example domain – Artillery, Target, Precision Guidance System, and Terrain. Each of these ontological relationships becomes a rule in our rule base. This ontology shows the hierarchical and dependency relationships between entities in the domain.

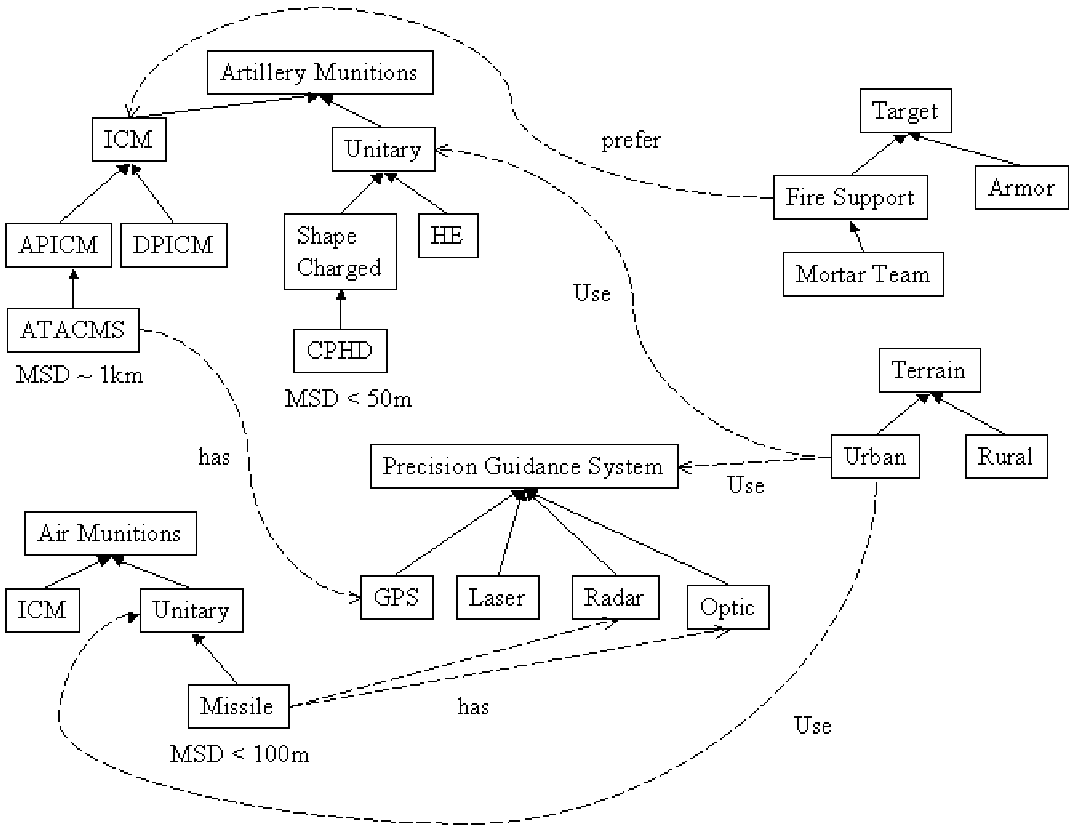

Figure 4 Example Ontology

Assume that the commander has defined the following policy rules.

1. "**PREFERRED** Munition for FS type targets is an ICM Munition"
2. "Targets in an Urban Environment **MUST** use Munition that has a Precision Guided System"
3. "Targets in an Urban Environment **MUST** use Munition that has a Unitary charge"

The ontology had defined the following relationships:

- ATACMS is an ICM Munition with Precision Guidance System
- Air Munitions of type Missile have a Radar and Optic Precision Guidance System
- CopperHead (CPHD) is a Unitary Shaped Charge.

Policy Request Criteria	Rules Returned	Policy Consumer Recommendation	
Target Type=Mortar Team	1	Any munition (ICM preferred)	Any munition available
Target Type=Mortar Team and Environment=Urban	1,2,3	Precision Guided Munition with Unitary Charge	Air Missile

During the scenario, the Commander realizes that the Copperheads available to him are not being recommended, even though it has a Precision Guidance System. A quick examination of the ontology reveals to the Commander that a relationship between CopperHead and Laser Guided is missing. He quickly modifies the ontology to define this newly recognized relationship. Now with the same mission requests, the results are

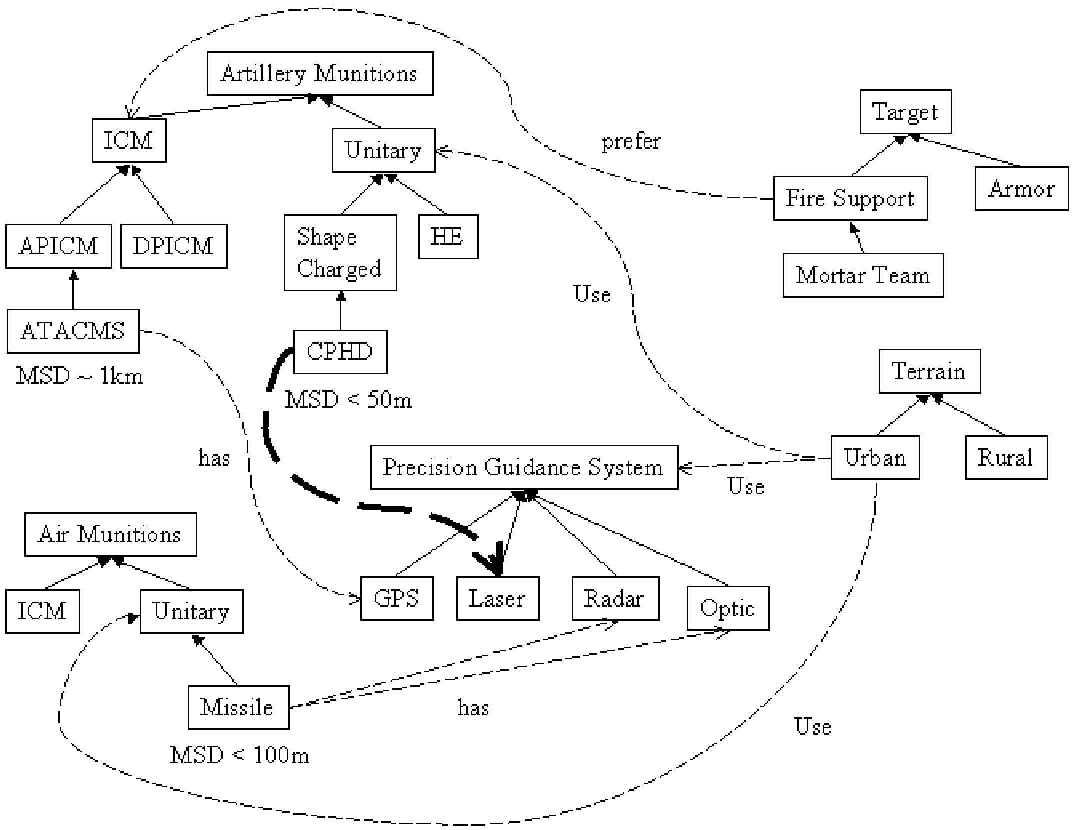

Figure 5 Updated Example Ontology

Policy Request Criteria	Rules Returned	Policy Consumer Recommendation	
Target Type=Mortar Team	1	Any munition (ICM preferred)	Any munition available
Target Type=Mortar Team and Environment=Urban	1,2,3	Precision Guided Munition with Unitary Charge	Air Missile, CPHD

By modifying the ontology, a previously unused munition is now utilized in a new capacity; with an improved response time while achieving the same desired effects. This happened by simply defining a new relationship in the ontology. This type of flexibility is required for today's dynamic battlespace.

5.4 Cognitive Extensions

The proposed approach of coupling ontologies with rule-based programming can be extended to enable cognition. Such extensions could include,

- Learn from decision patterns, analyze results, and learn from mistakes
- Recognize uncertainty in the situation and task assets appropriately
- Identify conflicts in policy
- Discover patterns in policy decisions to recommend new policy

In the military domain, a policy enabled command and control system could learn from the situation and engagement results. This would enable the system to refine and optimize the recommended solutions. The system could also manage uncertainty by recognizing that there is missing information and make recommendations for collecting the missing data. For example, a potential target is detected but the requisite TLE is not met due to sensor inaccuracies. Before making a recommendation for engagement, the system determines that a UAV should be re-tasked to collect better imagery and that the corresponding information then be mensurated to reduce the TLE to within acceptable standards.

Furthermore, to increase efficiency and optimize attack options the system could learn from Battle Damage Assessment (BDA). For example, initially, the system recommends 54 rounds of DPICM to destroy a tank. However, during operation, empirical data is collected showing that it takes far fewer rounds to destroy a tank. Automatically, the system would update the ontology with this new knowledge. Future recommendations, would then consider this newly discovered information to recommend fewer rounds against tank targets.

5.5 Multilevel Security

Multilevel Security is the concept of processing information with different classifications and categories that simultaneously permits access by users with different security clearances and denies access to users who lack authorization (ATIS). The proposed approach can be used to solve these concerns, by applying policy rules and ontological relationships between data, operators, and computing environment to control access to information. This access can be further shaped to provide role-based tailoring to operational functions inside the system. These roles can be dynamically narrowed or expanded by adding, removing, or modifying policy rules.

6 Next Steps

Research in the area of generalized policy has uncovered several topics that need to be researched (Kagal et al, 2004) such as performance and processes. Issues of concern specific to the military domain in the context of the Joint Battlespace InfoSphere were identified by researchers at the MITRE Corporation (Herinka et al, 2003) and include the following:

a. Guaranteed Service – Can we guarantee Quality of Service and Timeliness?
b. Support for changing situational environments – The battlespace is a very dynamic domain. Is policy able to keep up?
c. Support for policy evolution – How can we validate policy rules and understand their inter-relationships?

Additional areas of research include

a. Trust – Will the user have confidence in the solution?
b. Knowledge elicitation – How is domain knowledge captured?
c. Training – How do commanders translate Commander's intent into policy rules? Will the ontology be maintained to accurately represent the domain knowledge in a dynamic environment?
d. Conflict Resolution – How are conflicts between rules detected and resolved?
e. Supportability and Testing – Can problems be easily identified and resolved? What new testing methodologies will be needed?
f. Performance – Can this approach be responsive enough for time-critical threads?

This proposed policy approach does not preclude cognitive tools. Using an expert system combined with ontologies we can further develop a policy engine that can learn from the actions of its users through pattern recognition. If the system recognizes that a commander's response always follows a particular pattern, then the response can be anticipated. Based on these repeated actions, the system may recommend a new policy rule to the commander for inclusion in the rule base.

Another extension is to maximize reusability by loading the domain ontology at run-time. This approach allows the relationships in the domain to be updated or refined very quickly through updates to the ontology.

7 Summary

This paper has outlined an evolving operational need for automated policy. Over the past twenty years policy has been applied both automatically and manually to achieve commander's intent. Current capabilities cannot keep pace with the contemporary battlefield environment. This paper presented a realistic scenario demonstrating the need for flexibility and adaptability.

Additionally, this paper outlined an architecture and approach coupling ontologies with a rule-based system to provide an adaptable and flexible solution to challenges of the automated policy problem.

Policy-enabled systems have become a growing area of research in many technical communities, including the military battlespace. Emerging areas of research include the generalization of policy so that it may be applied to many domains. Leveraging the successes realized in the networking and security communities, new domains for automated policy have been identified. Models, terminology, and architectures have been developed to better understand and transition this technology into the mainstream.

Further research is encouraged to explore the feasibility and application of this approach into the military domain.

References

ATIS Committee T1A1, "Alliance for Telecommunications Industry Solutions", ATIS Telecom Glossary 2000 at http://www.atis.org/tg2k/

Chen, H.; Perich, P.; Finin, T.; and Joshi, A. 2004. "SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications". In Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous 2004), 258-267. Boston, MA.

Strassner, J. 2004. "Policy-Based Network Management; Solutions for the Next Generation": Morgan Kauffman.

Gruber, T. 2003. "What is an Ontology?" at http://www-ksl.stanford.edu/kst/what-is-an-ontology.html.

Ernst, J. 2003. "What are the differences between a vocabulary, a taxonomy, a thesaurus, an ontology, and a meta-model?" at www.metamodel.com/article.php?story=20030115211223271

Davies, Fensel, Van Harmelen, 2003. " Towards The Semantic Web: Ontology-Driven Knowledge Management," John Wiley & Sons.

Noy, N. and McGuinness, D. March 2001. ``Ontology Development 101: A Guide to Creating Your First Ontology''. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880 at http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html.

Friedman-Hill, 2003. “JESS In Action,” Manning Publications Co.

“REI Ontology Specifications, Ver 2.0” at http://www.cs.umbc.edu/~lkagal1/rei/.

Kagal, L, September 2002. "Rei : A Policy Language for the Me-Centric Project", TechReport, HP Labs.

Kagal L., Finin T., Joshi A., Oct 2004. “Declarative Policies for Describing Web Service Capabilities and Constraints”. In the W3C Workshop on Constraints and Capabilities for Web Services.

Herinka, Wild, Allen, Smith, Zhang, Panek, Semy, 2003. “Towards Policy-based Information Management for the Joint Battlespace InfoSphere”. In Proceedings of the 4th International Workshop on Policies for Distributed Systems and Networks.

Jackson, P. December 1998. “Introduction of Expert Systems,” 3rd Edition: Addison Wesley.

Acknowledgment

The foundation of this paper was based on the 2004 Graduate Independent Study course: “Cognitive Enablers” for Dr. Kenneth Modesitt, Professor of Computer Science and Associate Dean in the School of Engineering, Technology and Computer Science at Indiana University-Purdue University Fort Wayne (IPFW).