13th ICCRTS: C2 for Complex Endeavors

"An Architecture for Policy-based C2 Decision Support Systems"

Jim Jacobs(Raytheon), Amy Lange(Raytheon), John Watts(Raytheon), Robert Sedlmeyer (Indiana University-Purdue University at Fort Wayne), John Bryan(Raytheon), Kathie Irk(Raytheon)

Topic 4: Cognitive and Social Issues

Topic 9: Collaborative Technologies for Network-Centric Operations

Topic 1: C2 Concepts, Theory, and Policy

Point of Contact: Jim Jacobs

Raytheon Company

1010 Production Road, MS A3-02

Fort Wayne, IN 46808

+1.260.429.6069

Jim_Jacobs@Raytheon.com

An Architecture for Policy-based C2 Decision Support Systems

Abstract

At the 11th International Command and Control Research and Technology Symposium (ICCRTS), an architecture was proposed supporting a more flexible, policy-based approach to command and control systems.[1] Such a system would be adaptable to ever-changing warfighter needs and different phases of modern conflicts. Policy in its many forms is an essential part of military decision-making with examples such as Commander's Guidance, Rules of Engagement, security restrictions, legal and international policies. Additionally, commanders use policy to shape missions, pair weapons with targets, allocate airspace, and request coordination.

As policy use grows in importance, so does the need to flexibly adapt policy and to select the most relevant for the situation at hand. Systems that attempt to support the use of policy must move from hard-coded, predefined structures to more flexible representations expressed in the warfighters' terminology.

A demonstration-of-concept architecture was built to explore policy-based decision support, combining a situational awareness ontology, a flexible policy engine with externally defined policies, a context manager, and intelligent agents. These architectural components were integrated with an existing Service Oriented Architecture (SOA)-based targeting tool called the Joint Target Manager and exercised in an experimental environment. This paper discusses this work including lessons learned and next steps in development.

Keywords: policy-based, targeting, ontology, architecture, cognitive systems, decision-making, decision support, intelligent agents

Introduction

The challenge of military decision-making has always been significant. The tremendous cost of a poor decision has always led to a strong emphasis on training and preplanning, and the use of policy in its various forms has long been a method for providing guidance to decision-making – guidance developed with the leisure of time and shaped to the expected situations. Modern command and control (C2) systems such as the Advanced Field Artillery Tactical Data System (AFATDS) are using business logic rules to provide policy guidance. However, the effective use of policy in such automated systems is limited sharply by the difficulties associated with keeping them current and in tuning their behavior to reflect the shifting realities of the modern warfighting environment.

Using ontology-based policy reasoning systems, we developed an architecture for coupling dynamic semantic reasoning tools to existing C2 systems using a service-oriented architecture. This architecture allows for semantic representation of policy knowledge in a highly dynamic form that can be adapted to the evolving needs far more rapidly than traditionally developed systems. This approach enables strong use of intelligent agents within the local reasoning context.[2] As an additional evaluation of the policy-based reasoning approach, we also developed a second application in which the

architecture was used to provide oversight and guidance for distributed control of mobile ad-hoc networks (MANET).

The Problem

The term 'decision-making' often conjures up an image of an intelligent and rational, highly trained individual deliberating about several options and carefully selecting the one they feel is most likely to result in an advantageous outcome. In reality, warfighters are rarely afforded substantial amounts of time for planning before they are deeply embedded in critical situated action. Most of the features of complex decision-making that one might have expected of generals and admirals two centuries ago are now faced by junior officers and non-commissioned officers (NCOs) heavily engaged in the conflict of the moment.

The environment for making these complex decisions continues to change as well. Instead of a stable war room, with plenty of information and subordinates to track down details, the decision-making must be done in the heat of the moment with a paucity of information and with extreme time pressures. Uncertain and missing information has always been a part of the decision landscape, often simply referred to as ‘the fog of war’. Modern technology has changed the picture, but not always for the better. Modern intelligence tools and organizations are using highly sophisticated means to develop an overwhelming mountain of data. Often the most difficult decisions is to select which pieces to ignore and which pieces of information are the most significant for the decision at hand. All too often the critical piece of information is overlooked in the flood of data that is tangentially relevant at best.

Mission Need Trends

These challenges can be summarized into two broad categories of trends that are problematic for today's military decision maker at any level of the command hierarchy. The first of these concerns the changes in the military mission itself. The complexity of the decision-making has continued to grow both in terms of the number of elements and the difficulty of each decision. Additionally, substantially less time is available in the increasing tempo of the modern battlefield. Another element in these mission need trends is the increasing variety of military operations. In today's military, the diversity of operations is greater than ever before. This means that while each mission may not be completely novel, there is likely to be operational aspects and characteristics of each that are unique in themselves or in combination. The old adage, ‘train like you're going to fight; fight like you've trained’ is increasingly difficult to carry out. You can train for a wide variety of circumstances, a wide variety of operations, and with a wide variety of tactics; and still be surprised by novel circumstance or rapidly evolving tactics of a very dynamic and agile enemy.

Operational Context Trends

The operational environment itself has seen a number of additional trends that are challenging. One of these is increased distribution of assets, information, authority and action. The fairly static or a least well-defined front lines of the century or so ago, have

long passed. Actions today are not defined between territory held and enemy territory. In the last two engagements of the United States (US) military in Iraq and Afghanistan, there were relatively few large action battles; those that did occur were early in the campaign, and each was followed by a sustained period of small unit actions across a wide geographic area. Not only were the actions distributed geographically, the warfighters themselves and their equipment were sourced from a much wider range than in the past. From long-range bombers based in the continental US to real-time satellites; the logistical, strategic and tactical coordination issues are dramatically more complex than ever before.

All of these operational context trends as well as the trends in mission needs, lead to an increased need for careful high-quality decision-making. To aid warfighters at all levels, we must provide very high quality decision support tools and frameworks.

A model of 'policy'

As a concept, 'policy' is multiply defined. The American Heritage dictionary defines policy as "a plan or course of action, as of a government, political party, or business, intended to influence and determine decisions, actions, and other matters." [3] An alternate definition gives policy as "a course of action, guiding principle, or procedure considered expedient, prudent, or advantageous." [3] From the Wiktionary definition, we get "a course of action thought to be prudent or advantageous".[4] An alternate definition gives "a statement of requirement, often used in an organisation to instruct personnel as to a specific course of action." [4] Another aspect of policy becomes clear in these definitions—policies are made a priori. They are not decisions made in the moment, or as a part of situated action. Rather we see policy as something that is deliberative in origin and typically developed in counsel with authorities and subject matter experts.

In the context of military decision making, policy comes in many forms – all of which are determined long before the actual engagement in which they are needed. Commander's guidance, rules of engagement, and standing orders are all common forms of policy. Additionally conventions, treaties, and international law all form constraining policies that must be considered at the tactical level as well as during strategic decision-making. The doctrines studied and taught at various levels and in war colleges across the country could also rightly be considered as forms of policy. Throughout military training materials, we find a large body of knowledge, corresponding to heuristic rules and established models, to support complex action selection. The a priori deliberative authoring of policy maps well to the idea of capturing knowledge, heuristics and accepted models for reasoning in support of contextual decision-making.

Policies are more than a simple collection of 'if-then' production rules. A heuristic learned by a particular helicopter pilot may be quite valuable to him and his colleagues, but it is clearly quite a different thing from guidelines established in formal policy or even international law governing the operation of his helicopter. A useful trick learned by a particular operator or even shared among a group of practitioners is not considered to be weighted the same as formal policy, commander's guidance, or established rules of engagement. The word *policy* connotes an authoritative rule or established guideline with an originating authority that shapes the operation of those under its command. This

means that policy guides the selection of lower-level rules and reasoning types; policy guides the reasoning behavior of the individual rational agent. This is true for humans and should also be true for automated reasoning systems or decision support systems. Effective management and dissemination of policy improves the performance of actors, human or automated, and reinforces the value of providing policy-based decision support.

The Approach

The team's strategy was to apply leading research in policy-based systems to existing C2 applications. The objective was to create a 'demonstration of concept' system that would become a growing infrastructure for additional research. The initial project focused on a fielded decision support system for artillery targeting built on an SOA. The second major effort shifted domains to MANET and explored the potential of policy-based reasoning to support other techniques from ongoing research.

Policy-based Decision-making Tools

As described in a paper presented at the 11th ICCRTS, we sought to explore the value of policy-based reasoning systems within the C2 environment. The initial focus was on the challenges of the targeting officer working in a divisional artillery role. We also wished to explore the possibilities emerging from policy-based system research rather than working with the reduced semantics of rule-based systems. We sought a flexible framework to support rich interaction between an operator and a policy-based decision support system. Such a system would provide for substantial capabilities and support varied forms of reasoning within a shared context.

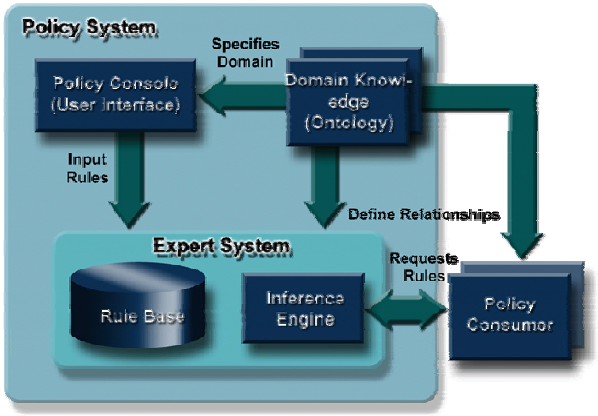

Figure 1. General structure of a policy-based reasoning service.

Our approach was to use a leading policy-based system from ongoing research, having a design and architecture for close integration with existing decision support tools and C2 systems, and providing a platform for intelligent agents to support the needed reasoning. We selected the Knowledgeable Agent-oriented System (KAoS), developed by Bradshaw and others at the Institute for Human Machine Cognition (IHMC)[5] which provides rich support for policy-based reasoning. Its modular design includes a standard rule engine, a policy console interface for user interaction and the KAoS policy administration tool

(KPAT) for policy development and editing. As shown in Figure 1, the basic architecture of a policy system includes domain knowledge in the form of ontology that captures the semantic information used to express and reason about policy. The policy console is a user interface where policies are defined. As the state of the domain is updated, the expert system uses standard inferencing and delivers the relevant rules to the policy consumer.

Applying the KAoS policy service

KAoS is an IHMC-developed tool, delivered as a service, to specify and apply policy in terms appropriate to the application domain. KAoS provides a policy management framework based on a model comprised of *actors* which are *(permitted| forbidden| obligated)* to perform *actions* with *constraintProperties* under *qualifiyingConditions*. KAoS is delivered with a core ontology which defines these basic terms and their relationships and extends the basic terms with common specializations (subclasses).

Applications extend (subclass) the core terms with additional ontology definitions which express the domain concepts needed to express policy in application-specific terms. Once the application-specific ontology extensions have been made available to KAoS, those terms are immediately available in the graphical user interface (GUI) policy editor for specifying policies.

Policies are applied by 'agents' which register with KAoS, identifying themselves by name, class, and domain. KAoS selects all policies which relate to the registered agent and compiles them for execution efficiency. An agent must 'understand' the terminology used for the policies to communicate with KAoS, using strings to specify the action and conditions for which authorization or obligation is sought. A utility is provided to read the ontology and generate Java String constants and String functions for the necessary terms.

Agents construct Action Instance Description (AID) structures which are delivered to KAoS to apprise KAoS of a proposed action and the conditions under which it will occur or of the status of some ongoing action. KAoS provides several methods for querying about policies depending on what type of guidance an agent wants.

The development of an ontology begins with the careful consideration of the key terms in the domain and their relationships. This phase of ontology creation results in a specific domain ontology that is linked to the standard ontology elements used in the KAoS system for representing abstract concepts related to policy-based reasoning. Once the domain ontology is defined, it is integrated with the existing KAoS ontologies and incorporated into an ontology server. This publishing step makes the ontology available to the KAoS policy consumer and other reasoners. At this point, the developers can use the KPAT policy editor and analysis tool to include the domain policy information into the correct formats for reasoning. These definitional and development steps form the basis for the operation of the policy-based service and, during runtime mechanisms within KAoS, respond to a request from client agents that make use of the service.

Phases of policy development:

- Ontology creation – extend KAoS Actor, Action, and Property classes and save the result as a web ontology language (OWL) file
- Ontology publication (for KAoS access) – strategies for varying levels of external connectivity
- Policy creation – KPAT editor and snapshot mechanism
- Policy application – mechanisms used by agents to determine what actions the policies direct

In the general form, a policy is stated as is stated as '(*specificAgent|classOfAgent)* is (*permitted|forbidden|required)* to perform *classOfAction* with conditions *actionProperties* under conditions (*stateOfAnAction|etc)*.'

Obligation Policies

Obligations 'trigger' action upon detection of a condition which satisfied an obligation policy's condition. The type of obligation policy demonstrated is triggered by the status of some other action. KAoS provides a set of default action states with an implicit ordinal arrangement. An obligation policy may be defined so that a request for authorization of the trigger action will cause the obligated action to be triggered. A triggered obligation may be dealt with by the agent requesting authorization for the triggering action (if that agent is registered with KAoS as the class of agent tasked to perform the obligated action). Alternatively, another agent – which has registered as the obligated class – may be asynchronously notified to perform the obligation.

The KAoS implementation detects when an obligated agent is also the agent requesting authorization for the trigger action. In this case, KAoS does not invoke the agent asynchronously, even if that agent is registered under a second class name. Instead, KAoS expects this agent to use a query which simultaneously confirms authorization and delivers a list of obligations to the agent.

Authorization Policies

Authorization policies apply to *actors*. For policies to be useful they need to prescribe a permitted/forbidden 'space' with a larger 'space' of the opposite permitted/forbidden condition. (e.g., a policy that authorizes an action when the temperature is between 50 and 80 degrees is meaningless if that action is also authorized without any conditions. For that policy to be useful, the action must be forbidden by default so that the action is only authorized in the 50-80 degree range).

Domain is the construct used by IHMC to control the default authorization. An agent registers with KAoS specifying its name, ontological class, and domain. Domains can be defined via the same KPAT GUI used to create/edit policies. If the domain does not exist when the agent registers, it is created and assigned a default *positive* authorization.

Shared Semantic Framework – A Context for Reasoning

In addition to the policy services, other forms of reasoning are typically needed to support the desired intelligent behavior. All of these elements must work together in a context of shared meaning. The basic architecture for incorporating semantic-based reasoning tools and policy-based systems in particular, within an SOA uses a two layer approach. The lower layer is the information management layer where standard SOA tools and techniques are applied. As shown in Figure 2, adapters are used to take information in common formats and schemas from the information management layer and convert the information into assertions within a semantic reasoning framework, sometimes known as 'the cognitive layer' or 'the context mechanism'. As assertions are made in the shared knowledge space, reasoning agents interact with that space using Semantic Web services. In this way, collaborative reasoning is enabled by the use of a common ontology representing the semantic information needed to generate the desired intelligent behaviors. It is expected that a wide variety of reasoners and intelligent agents may share this same local context. For policy-based systems, the policy services exist as a module that is accessible to agents using the same shared context mechanism.

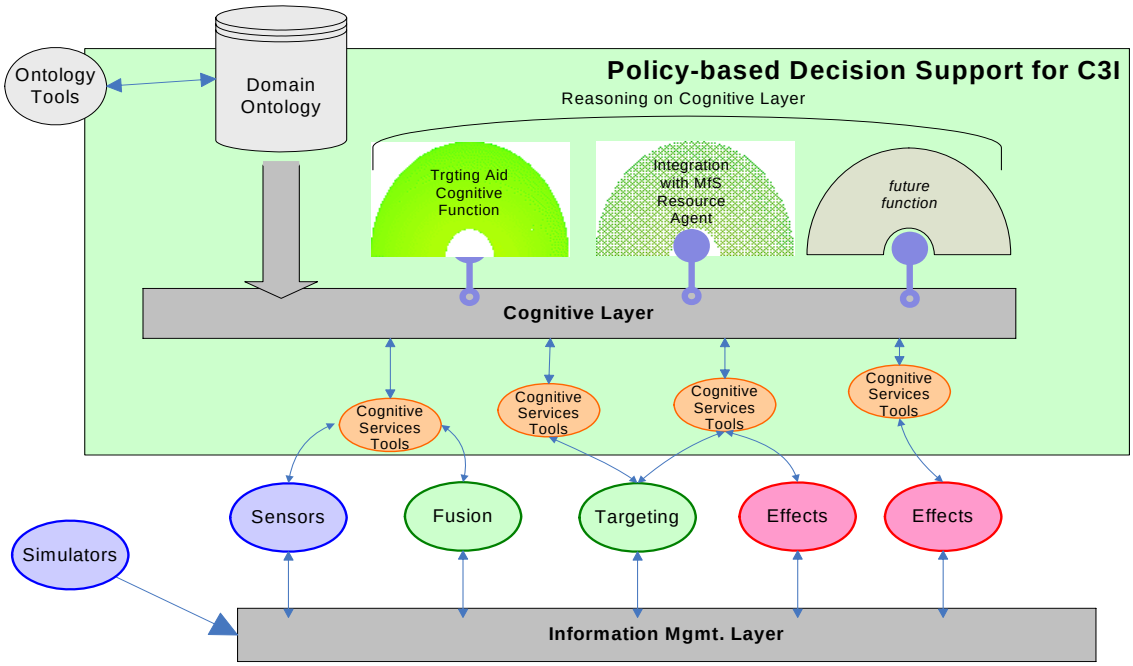

Figure 2. General Architecture for Integrating Policy-based Reasoning with an SOA

In our architecture, we used the Jena, a semantic web framework for Java, as the 'cognitive layer' that has a store of current assertions and as a reasoning environment. Jena, developed at Hewlett-Packard (HP) labs, is an open source framework for semantic technology applications. It provides a variety of inference engines and middleware components running in JBoss, an enterprise application platform. An instantiation of this architectural pattern typically has a number of intelligent agents within a shared framework, each with its own domain knowledge and functional responsibilities. In a policy-based system, at least some of those agents will be querying the policy service for guidance about which policies and rules are applicable at any particular time. The

collection of agents forms a collaborating team that works together to generate the intelligent behavior desired from the overall system.

Towards System of Systems Integration

The use of adapters between more conventional service oriented systems acts as a form of ‘impedance matching’ that adapts existing systems to the new architecture for semantic tools and policy-based reasoning. In our applications, real-time data into the system is generated by a system stimulator. This adapter brings data from the existing communication bus - as a stand-in for the Global Information Grid (GIG) - into the shared reasoning context. Use of adapters in this fashion has been called "ontological normalization".[6]

The Results

Our initial system applied policy-based decision support to the challenges facing a targeting officer (see Figure 3). In the existing tool, the Joint Targeting Manager (JTM) provides decision support for the targeting officer through a data fusion mechanism that collects related data from a wide variety of sources and provides an integrated environment for working with this data. Working within this tool, the targeting officer nominates targets for prosecution and passes the vital information to effects management tools through the standard service interfaces. A key concern for users of this and similar systems is the difficulty in handling the large volume of data. By applying the policy-based approach, we were able to use intelligent agents to draw on policy services within this reasoning framework to intelligently filter the data presented to the officer. The flexible nature of this architecture allowed us to bring this added value to an existing tool with very limited modification to the existing system. Specific agents within the reasoning framework were used to handle the interactions to the targeting tool, while others worked with the policy services and other reasoning tools.

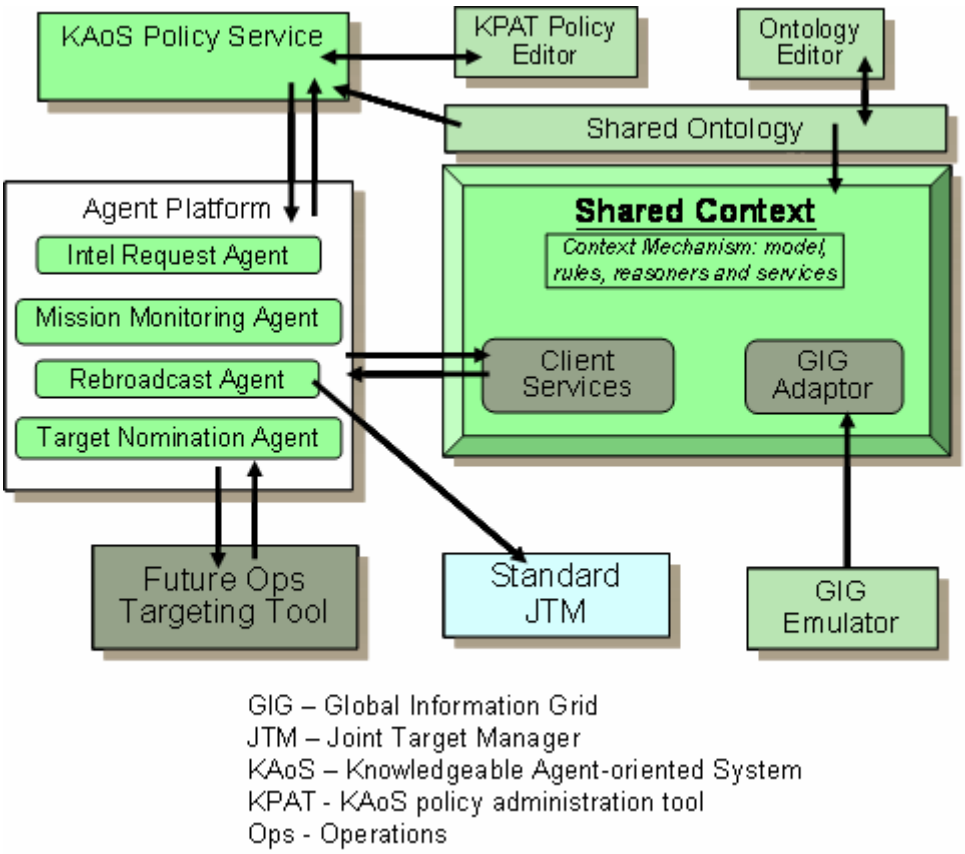

Figure 3. Architecture for a Policy-enhanced Decision Support Tool as Demonstrated

Applied to MANET

Policy-based systems running on each node of a mobile ad hoc network can very flexibly guide the configuration and use of network resources.[7] In this 'demonstration of concept' project, we used a testbed that simulated a mobile ad-hoc network with a set of servers, each with dual network interfaces. One of the network interfaces was used to control the system and the other acted as the simulated network. The network controller managed movement information and other network behavior dynamics according to the specific scenario script being studied.

Each node had its own reasoning framework, its own collection of agents, and typically its own policy service. If there was reliable connectivity to a dedicated node that provided policy services, it would not be necessary for every node to have its own policy server. This would allow for very lightweight nodes to benefit from advanced reasoning about their use of network and other resources by using remotely hosted policy services.

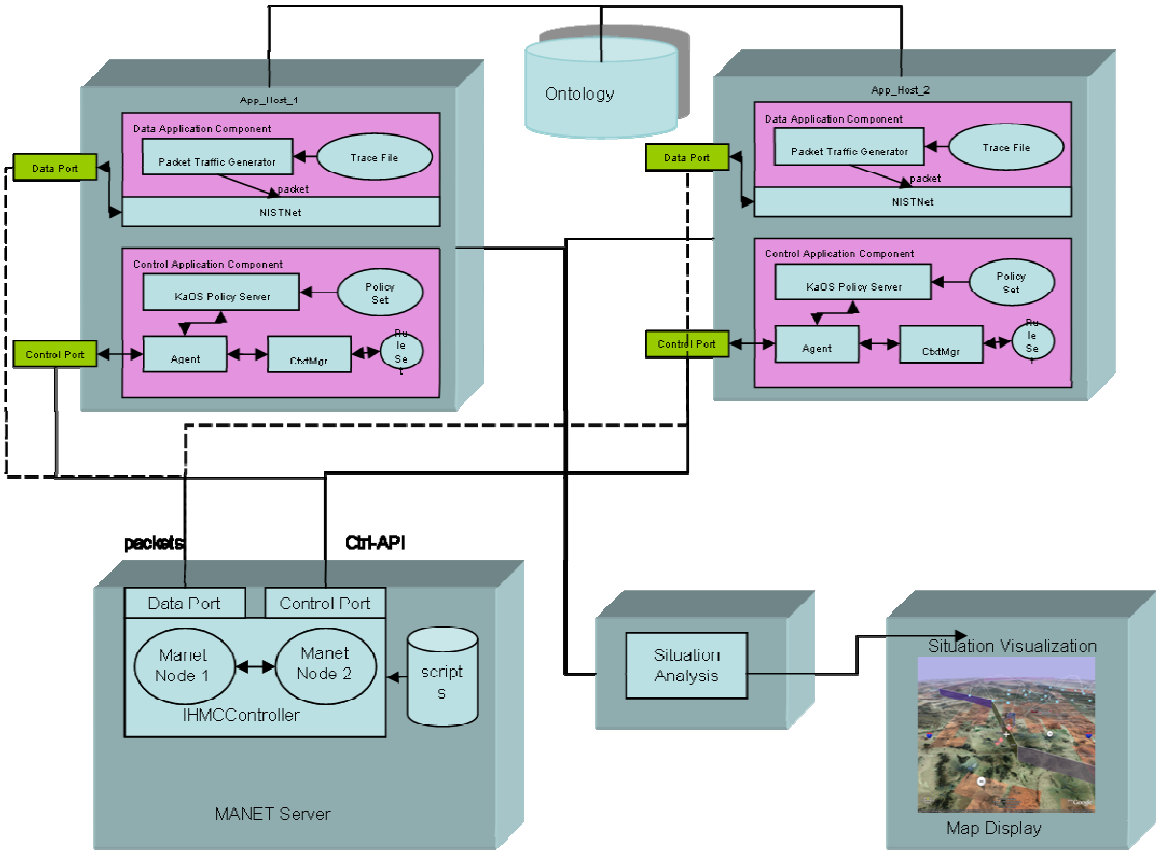

Figure 4. Integrating Policy-based Reasoning to Support the Operation of a MANET.

Figure 4. shows a simple two node architecture coordinating through a simulated MANET. We used a battlespace situation assessment tool to provide visualization of the simulated network, both its performance and its simulated geographic location. The shared ontology shown in the diagram represents the build-time sharing of semantic representation, not a runtime shared knowledge store that spanned all of the nodes.

For the focus of this particular demonstration of concept, we worked primarily with policies regarding regulatory compliance over varying geographical spaces. For example, as an unmanned aerial vehicle (UAV) travels across regulatory boundaries, the intelligent agents attached to its network node made use of the policy server to ensure the frequency and power settings in use stayed in compliance with all applicable regulations. While lower level details of packet formation and routing were handled by the MANET algorithms, the application-centric issues of data shaping and context issues of regulatory compliance were handled by the policy service.

Conclusion

The essential role of policy in military decision-making is clear. Likewise, so is the need for advanced decision support systems. Policy-based reasoning systems promise a higher degree of flexible, intelligent support for decision makers than previous frameworks could allow. This effort has established the achievable value of such systems and shown

that extant research has resulted in viable architectures and policy service frameworks that are ready for transitioning to operational systems.

There are many significant issues to address in moving this technology to higher levels of technology readiness. The lifecycle of knowledge must be addressed for policy systems, as it must be a consideration in any reasoning system. The software implementing policy research is maturing, but requires substantial investment to lift it to the operational reliability requirements demanded of fielded systems. Additionally, as policy reasoning moves closer to the forms of human problem solving, they form a tremendous opportunity for highly dynamic collaboration between warfighters and their supporting cognitive systems environment.

Policy is ubiquitous in military decision-making with an abundance of forms and subtle entanglements of authority, obligation and permission. Modern policy-based services such as KAoS provide an opportunity to make significant progress towards realizing the promise of this research at the critical point: mission assurance for the warfighters of today and tomorrow.

References

1. Buibish, Anne-Marie, Amy Lange, and Michael Woitalla. 2005. Responsive Decision Making through Automated Policy-Enabled Systems. Paper presented at11th International Command and Control Research and Technology Symposium, June

2. Bradshaw, J.M. "An Introduction to Software Agents", in Software Agents, Bradshaw, J.M. (ed.), Cambridge, MA: MIT Press, 1997.

3. Policy. *Bartelby.com.* Retrieved March 17, 2008 from http://www.bartleby.com/61/20/P0412000.html

4. Policy. Wiktionary.org. Retrieved March 17, 2008 from http://en.wiktionary.org/wiki/Policy

5. Bradshaw, J.M., S. Dutfield, B. Carpenter, R. Jeffers, and T. Robinson. 1995. "KAoS: A Generic Agent Architecture for Aerospace Applications", in Proceedings of the CIKM'95 Intelligent Information Agents Workshop. Baltimore, MD.

6. Schneider, T. (2006) Personal Communication.

7. Niranjan Suri, Marco Carvalho, Jeffrey M. Bradshaw, Maggie R. Breedy, Thomas B. Cowin, Paul T. Groth, Raul Saavedra, Andrzej Uszok, "Enforcement of Communications Policies in Software Agent Systems through Mobile Code," *policy*, p. 247, Fourth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'03), 2003.