Extending Net-Centricity to Coalition Operations

Niranjan Suri, Andrzej Uszok, Rita Lenzi, Massimiliano Marcon, Maggie Breedy, Jeffrey Bradshaw
Florida Institute for Human & Machine Cognition, Pensacola, FL

Yat Fu, James Hanna, Vaughn Combs, Asher Sinclair, Rob Grant, Robert Hillman
U.S. Air Force Research Laboratory, Rome Research Site, Rome, NY

Abstract

Network-centric warfare is often credited for the superior warfighting capabilities of the United States. One of the fundamental requirements for network-centric warfare is the timely exchange of information critical to mission success. However, an increasing number of ongoing military operations tend to be coalition based, and current security policies place severe constraints on information sharing between coalition forces. Bringing the advantages of network centric warfare to coalition warfighting requires a significant improvement in our ability to quickly share critical information while still satisfying security requirements. This paper explores a services-based approach to information management and argues that such an approach would improve information sharing in coalition environments.

1. Introduction

Major military operations ranging from counter insurgency operations as in Afghanistan to disaster recovery operations as in Haiti are increasingly coalition based. As the desired level of cooperation and coordination continues to increase, so does the demand for timely information exchange. Individual nations have recognized the advantages of net-centricity (Network Centric Warfare in the US, Network Enabled Capability in the UK) and are applying them within single-nation forces. However, these concepts are yet to be fully realized for coalitions, and examples from previous operations [1] have shown the inefficiencies and hazards that result from the lack of coalition information sharing. Therefore, there is a need to extend net-centricity beyond a single nation to multiple nations, in order to bring similar advantages to coalition operations.

Challenges facing net-centricity may be categorized into either technical challenges or security challenges. Technical challenges include interconnectivity, discovery, syntax, and semantics. Security challenges are primarily concerned with protecting restricted (e.g. classified) information, sources of information, and the methods used to obtain the information. These security challenges further exacerbate the technical challenges. For example, the interconnectivity problem at the level of tactical edge networks is created by differences in radio standards, frequencies, and cryptography. Therefore, interconnectivity is often possible only at designated gateway nodes. However, security requirements impose the need for a network guard that restricts the types of communication possible, complicating information discovery and sharing.

This paper explores using a services-based approach to information management (IM) to address the challenges of extending net-centricity to coalition environments. In particular, we focus on the basic service-based IM architecture, services for federation, and services for policy-based control. We also note that many of the security challenges associated with coalition IM are also present in cross-domain IM, and describe the advances made in this area. We propose that combining all of these capabilities is an effective approach to supporting net-centric operations that span coalitions. We begin by describing the current state of coalition IM.

2. State of the Practice

The process of information sharing across coalitions is complicated by the security requirements in place. Figure 1 shows the current process of interconnecting coalition networks, which involves a hardware device known as a Cross Domain Guard (CDG). The interconnection is typically pair-wise between the coalition partners. The CDG is a certified, trusted computing device that is designed to only allow certain types of information to pass through from one network to another. For example, the Radiant-Mercury system, originally developed by Lockheed Martin under a contract for the Navy, is a certified software application that runs on a trusted platform.

The Unified Cross Domain Management Office [2] has been setup in the United States to coordinate the efforts of developing and certifying CDG's and Cross Domain Solutions (CDS) in general. A survey of the currently available systems and their capabilities is presented in [3].

Note that the notion of a CDS is also used when interconnecting networks of different classification, even though they might belong to the same country. When the information flow is only from a lower classification domain to a higher classification domain, the CDS may be simpler and consist of a data pump or a data diode, which allows data to flow only in one direction. Typically, the data is checked to ensure that there is no malicious content (e.g. a virus embedded in a document) prior to transferring it from the lower classification domain to the higher classification domain.

In the example shown in Figure 1, any information passing from Coalition Partner 1 (e.g., United States) to Coalition Partner 2 (e.g. United Kingdom) will pass through a CDS. Given that each country has its own security concerns, we want to share just enough information with our coalition partners and protect our networks at the same time. In some cases, additional technical solutions or guarding devices will have to be put in place. For example, each country may have its own CDS, which means that all information will have to pass through two guards, one on the US side and one on the coalition side, before the document is sent to our coalition partners and vice versa.

The CDS will process the information differently, depending on the type of information. For example, structured information (e.g. XML) might be amenable to automated processing. An XML appliance could be used as part of the CDS to automatically manipulate

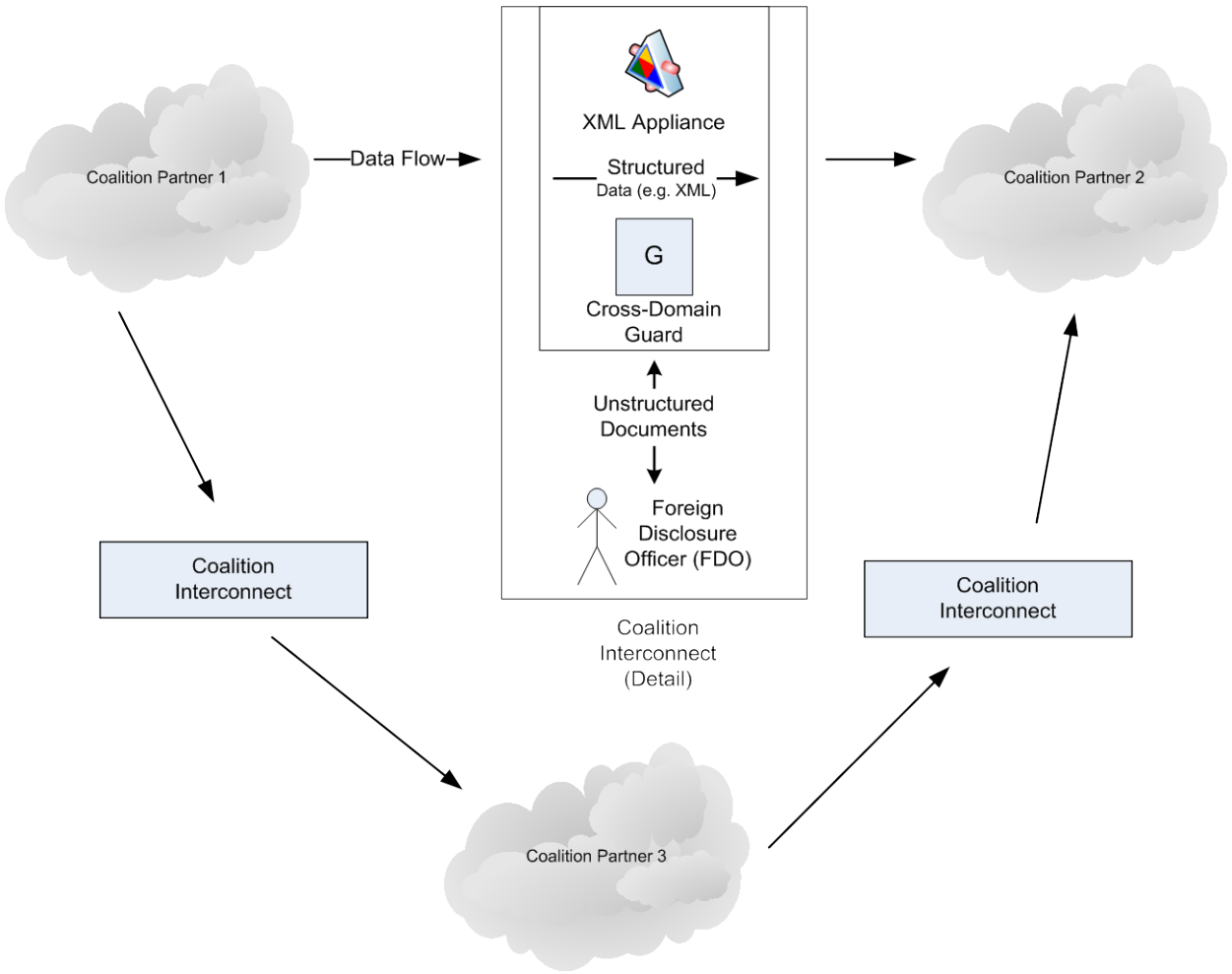

Figure 1: Current Method of Coalition Information Sharing

XML messages and pass them on to the coalition network. On the other hand, any unstructured information such as a document must undergo review by a Foreign Disclosure Officer (FDO). Depending on the criticality of the information, the processing time could vary from minutes to days.

These security and technical challenges create a barrier for us to share information with our coalition partners effectively. For some military operations, which often require agile and dynamic responses, coalition information sharing would be a nice addition, but most likely will not be considered as a requirement initially. Two primary issues are the cost of the CDS, as well as the time involved in certification. An added complexity is that changes, such as the type of information flowing through the CDS, may require redevelopment and recertification, which prevent a quick response to addressing information needs of rapidly evolving missions.

One approach to reducing the time involved in establishing a CDS is to decompose the functionality into a set of (composable) services. These services could each individually be certified. Then, deploying a new CDS, or modifying a CDS would require re-certification of only those services affected, which could be faster and less expensive. This paper addresses using a services-based approach, and decomposing the functionality into a set of services. The next section describes Phoenix, an underlying services-based infrastructure for information management.

3. Background – Phoenix -- Service Oriented Information Management

Service Oriented Architectures (SOAs) provide significant benefits when designing modern distributed systems. First and foremost, service orientation allows for the natural decomposition of business processes into a well defined and orchestrated set of services that encapsulate and export access to cohesive and modular functionality. This approach enhances efficiency through the potential reuse of services among disparate business processes and orchestrations. So called business processes may be supported by services that span quite seamlessly across multiple platforms. Further, services based approaches promote the use of service discovery mechanisms and brokering services that naturally support late and dynamic binding of applications to compatible and available services. Finally, SOA more easily enables the definition and enforcement of policy at multiple levels within the distributed system.

When attempting to support coalition operations and the federated management of information the aforementioned SOA characteristics should be considered as absolute requirements. Given the collection of disparate hardware and software environments that one is likely to encounter when engaging in coalition operations, it is very important to be able to flexibly instantiate and use collections of services across a vast array of potential deployments. In addition, there may be cases where, either based on dynamically changing policy or evolving availability of services, applications must broker for or locate services that are currently available and which will satisfy current requirements. Finally, services that support policy enforcement at many levels within the architecture can quite naturally and dynamically control the management and dissemination of information among coalition partners. For example, whether or not specific information should be shared with a coalition partner may be changed dynamically and pushed to appropriate enforcement points within the service orchestration. Furthermore, based on policy, information may be sanitized through the use of a filter so that only appropriate portions of a document are disseminated between federated collections of coalition services and applications.

The Air Force Research Laboratory (AFRL) has developed a reference set of Information Management Services to provide an essential piece of the envisioned Net-Centric IM solution for the Department of Defense (DoD). These IM Services provide mission critical functionality to enable seamless interoperability between existing and future DoD systems and services while maintaining a highly available IM capability across the wide spectrum of differing scalability and performance requirements. This effort, known as Project Phoenix, leverages existing in-house knowledge of IM, along with the expertise of research colleagues and customers, to define a SOA-based IM solution that will have significant meaning now and well into the foreseeable future. Part of this development effort focused on ensuring that the architecture and implementation aligns with the Air Force and DoD's vision of current and future network-centric operations.

We define information management as "a set of intentional activities to maximize the value of

information for achieving the objectives of the enterprise. [4]" The primary purpose of information management is to achieve effective information sharing among the many applications and users within an enterprise. We have identified three best practices as crucial to future net-centric systems:

Reduce complexity in the edge-user applications by utilizing a shared and supported infrastructure. The infrastructure will provide common necessary functions, such as authentication, authorization, prioritization, and demand-driven information flow. This will free the information provider and consumer applications from having to manage these functions. The infrastructure will provide universal services, such as publish, subscribe and query, that are information-neutral.

Increase controllability of the system by decreasing the number of places that must be modified to implement a change. By moving policy enforcement to the shared infrastructure, changes in policy can be accomplished without changing any of the edge-user applications. Similarly, when the operational environment changes, the infrastructure will be changed to compensate, and the edge-user applications will still function properly, with less application-specific adaptation.

Appropriately package information for dissemination and management. Effective management of information requires that it be characterized sufficiently so it can be interpreted unambiguously. The characterization is called metadata, while the information itself is called the payload. The information management infrastructure uses the metadata to better understand how and where to acquire, store and deliver the payloads.

The focus of the Phoenix project has been the definition and implementation of such a shared infrastructure that incorporates these best practices. The Phoenix architecture enumerates a set of services, constructs, and use cases that capture and represent the semantics and necessary functionality for managing information sharing and interoperability. The architecture also specifies how orchestrations of these services may be used to provide the basic functions of information management, specifically supporting publish, subscribe, query, and streaming metaphors. Although these use-case orchestrations are part of the architecture they by no means limit the manner in which the specified services may be used.

The Phoenix implementation provides basic services for information submission, brokering, discovery, dissemination, and query. Additional services are type management, session management, authorization, service brokering and event notification. IM services also support common information models that facilitate the management and dissemination of information consistent with client needs and established policy. The services support flexible and extensible definitions of session, service, and channel contexts that enable the application of Quality of Service (QoS) and security policies at many levels within the SOA.

4. Federation Services

The federation architecture supports seamless and secure integration of multiple information spaces, each of which is called a federate. Seamless implies that the architecture supports automatic discovery of and interconnection between federates. The process of federation is transparent to clients, which still connect to their home federate as normal. Secure implies that the federation process is not arbitrary and open. The establishment of federation and exchange of information is controlled via policies.

While the notion of federation was initially developed to support interconnection of information enclaves in a single nation, the same concept can be extended to coalition needs. Each coalition partner may be regarded as a federate, with the federation services providing the desired, controlled, sharing of information between them. The federation services support many of the needs for CDS, including policy-controlled sharing of information and dynamic information transformation.

One key aspect of the federation architecture is that all federates are peers. Each federate independently manages its connection with other federates and has its own set of policies that govern the exchange of information with other federates. This approach is particularly well suited for coalition scenarios, given that each coalition partner (and hence each federate) is a separate administrative domain.

Following the services-based approach, the federation capability is realized through a set of services that work in conjunction with each other. Figure 2 shows

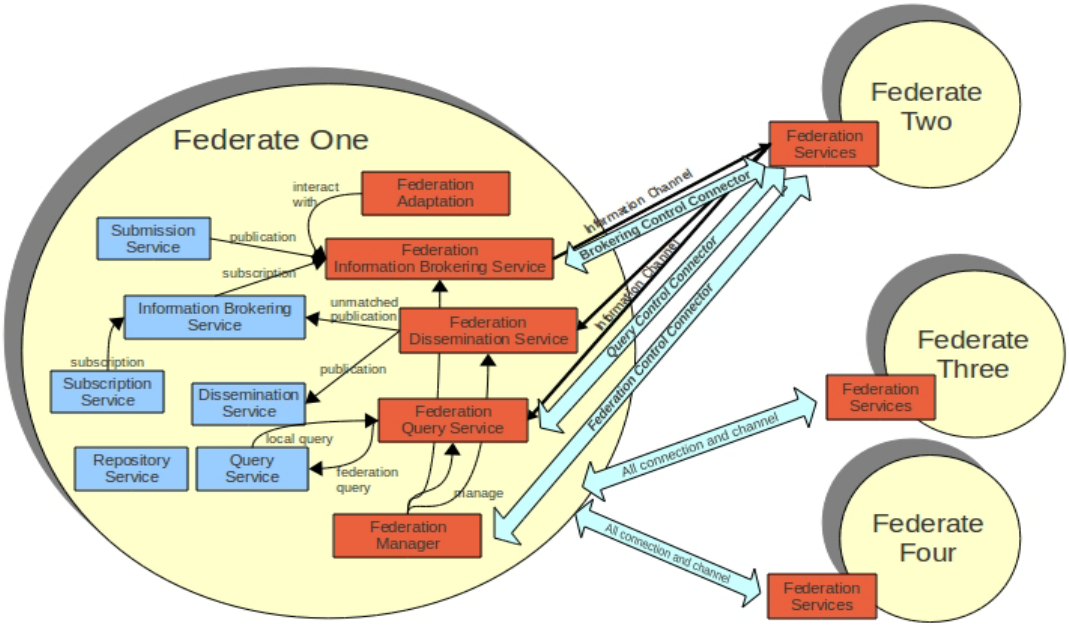

Figure 2: Federation Services and their Interaction with Phoenix Services

the key federation services and the interconnection between four federates. In this particular configuration, Federate One is independently connected with Federate Two, Federate Three and Federate Four. For simplicity, the connection and channels to Federates Three and Four are collapsed and we only show the federation services inside the first federate.

The key federation services are described below:

4.1 Discovery Manager

Federation begins when one or more remote federates are found. Federates may be found via static configuration (where the end-points are specified) or via a dynamic discovery process. The Discovery Manager component provides the discovery functionalities that are necessary to automatically find other federates in the network. The discovery process can rely either of the capabilities of the Group Manager [5] or on the Cross-layer communication substrate (XLayer) [6] for discovery and grouping support. With either of these systems, discovery relies on some variation of a broadcast or multicast at the network layer. When operating across coalition networks interconnected by a CDS, such discovery would not be possible because the network level communication would be blocked. Therefore, the role of automated discovery may need to be set aside and the system may need to resort to a pre-defined configuration.

4.2 Federation Manager Service

Once potential new federates are identified by the Discovery Manager, the Federation Manager (FM) Service is responsible for setting up the federation across the newly discovered entities. In particular the FM communicates with the new federates, negotiates contracts and informs the other federation services about the new federates. The FM is also responsible for handling disconnections and termination of federation.

4.3 Federation Information Broker

Information brokering is one of the fundamental services performed by Phoenix. Brokering involves examining new, incoming information that has been published and matching it against active subscriptions from clients. Any matching information is then forwarded to the appropriate clients through the Dissemination Service. The Federation Information Brokering Service (FIBS) extends information brokering to handle federates. It receives subscription registrations from the Subscription Service and

forwards them to the federates. It also receives the local publications from the Submission Service, brokers them locally on the behalf of the remote federates, and forwards them to appropriate federates. In particular, it forwards them to remote Federation Dissemination services (see below).

4.4 Federation Dissemination Service

Dissemination is the post-processing step that follows brokering and involves transmitting matched information to the clients. The Dissemination Service normally receives matched data from the Information Brokering Service. When federation is involved, the Federation Dissemination Service (FDS) is responsible for receiving matched information from remote federates that is destined to local clients. In most cases, when the FDS receives forwarded publications from remote federates, they have already been matched for the local clients (by the remote Federation Information Broker). In such cases, it uses local Dissemination Service to transmit the data to the clients. Otherwise, it uses the local Information Broker to publish the information locally.

4.5 Federation Query Service

Querying for archived information compliments publish and subscribe as the third primitive operation provided by Phoenix in the context of information management. Query differs from subscribe in being able to retrieve previously published and stored data. The query service permits information retrieval from the client's data stores and supports synchronous and asynchronous query execution. Data stores are managed by the Repository Service and they could be of two different kinds: repositories and archives. Repositories are low-latency high-access data stores that should support higher data read and write rates. Archives are expected to store much more data than repositories, but with a lower data access rate.

The Federation Query Service (FQS) extends the query capability to remote federates. It receives local queries and sends them for processing from both the remote federates and the local Query Service, collects the results, and returns them to the client. One of the assumptions made by the FQS is that federates do not have duplicated data, which simplifies the distributed query problem. The FQS may locally cache data that results from a remote query, thereby improving performance for repeated queries. The nature of the queries, as well as the behavior of the FQS, can be controlled via policy. For example, a query by a coalition partner being executed against a US database may be modified in order to limit the scope and nature of the query. This control is independent from the ability to control the individual objects that are a result of the query.

4.6 Federation Adaptation Service

During the course of operations, the resources available for information management are likely to change over time. For example, the network links connecting federates may become saturated, or the systems hosting federation services may become overloaded. The Federation Adaptation Service performs local adaptations to offset such shortage of resources. For example, under low-bandwidth situations, the Adaptation Service can temporarily suspend low-priority subscriptions in order to provide reasonable performance for the remaining subscriptions. The priorities of the subscriptions can be specified via the client or via policies. On the other hand, when computational resources fall short, the Adaptation Service temporarily disables local predicate processing. This causes the Federation Information Broker to send all publications to the remote federate, and for the brokering to occur on the remote federate. Subscriptions are sorted based on their hit-rate (i.e., the percentage of publications that match the predicate) and the subscription with the highest hit-rate is selected first. This minimizes the impact of an increase in the bandwidth utilization as a result of this adaptation.

For the adaptation service to perform its task, the underlying resources of the systems and networks need to be monitored. The adaptation service relies on an underlying Monitoring Service [7] to receive information about the system.

4.7 A Complete Scenario

To better illustrate the operation of the Federation Services, we will consider a complete scenario, from discovery to federation establishment to federation shutdown. To simply, we use a scenario where the federation happens only between two instances of an Information Management System (IMS), which we will refer as Federate One and Federate Two. We will also assume that the nodes where the IMSs run are provided with a lower level discovery-enabled communication substrate, such as Xlayer or Group Manager mentioned before.

Federation Establishment

When the Federation Service is instantiated along with other Phoenix IM services, the first step is the

registration with the Discovery Manager (DM). This is achieved through the use of the discovery and grouping API provided by the sub-layer, which allows for the registration of service capabilities. By registering and joining a predefined group, the IMS manifests its intention of being part of the federation. Once that happens, IMS instances are mutually notified of each other's existence. At this point a handshake phase starts. During the handshake each potential federate introduces itself to the other, sending a reference (endpoint) to itself. This contains all the information necessary to create a stub connected to the other federate, i.e. the IP address, the port number and the names for the services the federate can provide.

Eventually, a contract negotiation occurs and upon contract acceptance by both nodes, the federation is officially established. The local Federation Service (FS), Federation Information Brokering Service (FIBS), and the Federation Query Service (FQS) establish control channels with the corresponding peers in the remote federate. Publications and results of queries are transmitted over an information channel.

Subscription forwarding
When a client connected to Federate One issues a subscription with its local IMS, the request is captured by the FIBS via the local Subscription Service and the local Information Brokering Service. The subscription is forwarded to the remote FIBS. Once Federate Two obtains it, the subscription is stored in a remote subscriptions table, ready to be matched against local publications.

Publication handling
When a client publishes information to the local IMS (Federate One), such publication is intercepted by the FIBS. Under normal conditions (e.g., with no adaptation algorithms activated) Federate One attempts to execute the predicate matching locally, by comparing the publication type and metadata with the remote subscriptions it may have previously stored in its remote subscription table. Publications for which the local matching succeeds are marked as matched, and sent to Federate Two via an information channel. Federate Two receives the publication, verifies if it was already matched (and if it was not it matches it with the local subscriptions) and forwards it to the IMS. Finally the IMS takes care of the delivery to the correct subscriber clients.

Federation Termination
Federation lasts until at least one of the nodes leaves the federation group, or the connection between the two federates is lost. When the other is notified about one of these events it cleans up any references to the former remote federate, including any cached remote subscription. The system is now back in the initial state, prior to the federation being established.

Policies
All the federation operational behavior detailed above is entirely governed by policies. Before performing any step in its execution flow, FS verifies with the policy framework if and how the current operation is allowed. The following section on Policy-based Control explains in detail the contracts and policies used to dynamically control the behavior of federation.

5. Policy-based Control
KAoS [8], a set of platform-independent services, enables people to define policies ensuring adequate security, configuration, predictability, and controllability of distributed systems, including traditional distributed platforms (e.g., CORBA, Web Services, Grid Services), software agent frameworks (e.g., NOMADS, Cougaar, Luna), and multi-robot configurations. KAoS Domain Services provide the capability for groups of software components, people, resources, roles, groups, and other entities to be semantically described and structured into organizations of domains and subdomains to facilitate collaboration and external policy administration. The KAoS Policy Services allow for the specification, management, conflict resolution, and enforcement of policies within domains. KAoS policies distinguish between *authorizations* (i.e., constraints that permit or forbid some action by an actor or group of actors in some context) and *obligations* (i.e., constraints that require some action to be performed when a state- or event-based trigger occurs, or else serve to waive such a requirement).

Policies are represented in ontologies, not rules. The use of ontologies, encoded in OWL (Web Ontology Language, *http://www.w3.org/TR/owl-features/*), to represent policies enables reasoning about the controlled environment, about policy relations and disclosure, policy conflict resolution, as well as about domain structure and concepts. KAoS reasoning methods exploit description-logic-based subsumption and instance classification algorithms and, if necessary, controlled extensions to description logic (e.g., role-value maps). Unfortunately, many myths

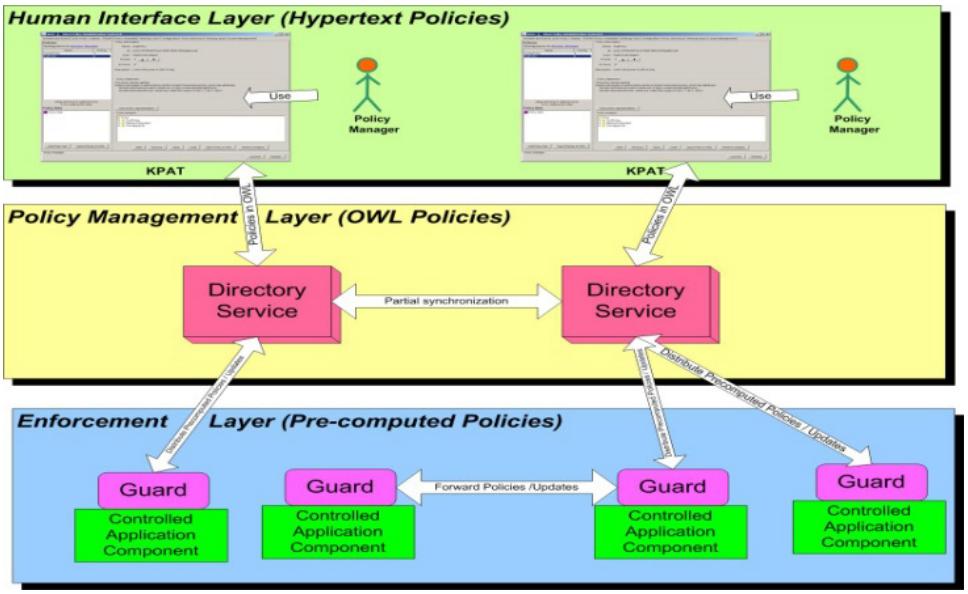

Figure 3: KAoS Policy Services Conceptual Architecture

have been propagated about the limitations of OWL for policy management—in our case, we have found it to be an extremely expressive, flexible, and efficient alternative [9].

5.1 KAoS Architecture

Two important requirements for the KAoS architecture have been modularity and extensibility. These requirements are supported through a framework with well-defined interfaces that can be extended, if necessary, with the components required to support application-specific policies. The basic elements of the KAoS architecture are shown in Figure 3; its three layers of functionality correspond to three different policy representations:

Human Interface layer: This layer uses a hypertext-like graphical interface for policy specification in the form of natural English sentences. This capability, called KPAT (KAoS Policy Administration Tool), hides the complexity of OWL from users, and provides the ability to analyze, monitor, and manage ontologies and policies. Further simplification of the policy specification task is possible through Policy Templates and Wizards. The vocabulary for policies is automatically provided from the relevant ontologies, consisting of highly-reusable core concepts augmented by application-specific ones. Unlike most other policy frameworks, changes of any kind can be made efficiently at runtime.

Policy Management layer: Within this layer, OWL is used to encode and manage policy-related information. The Distributed Directory Service (DDS) encapsulates a set of OWL reasoning mechanisms.

Policy Monitoring and Enforcement layer: KAoS automatically "compiles" OWL policies to an efficient format that can be used for monitoring and enforcement. This representation provides the grounding for abstract ontology terms, connecting them to the instances in the runtime environment and to other policy-related information (Figure 5).

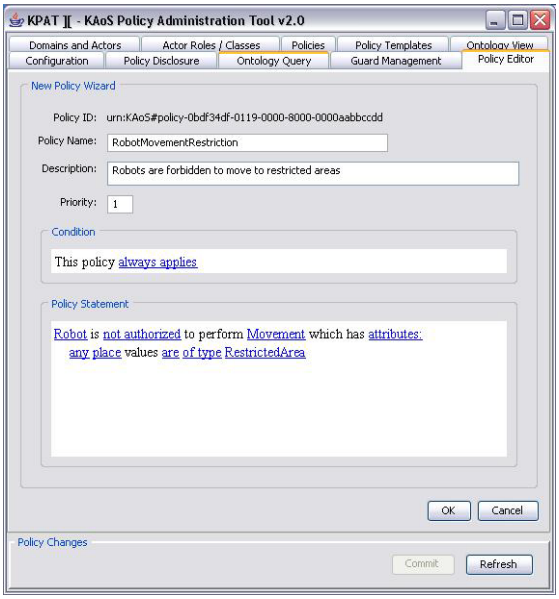

Figure 4: Authorization Policy in the KPAT Hypertext Policy Editor

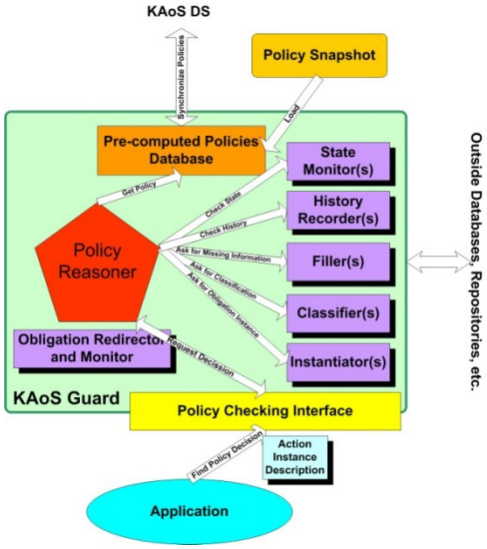

Figure 5: KAoS Guard - the Policy Decision Point Integrated with the Application

5.2 Controlling Federation

The Federation Service in each of the federates is integrated with the KAoS Guard, which stores a compact form of policies controlling the establishment, lifecycle, information exchange, and adaptation of the federations established by this federate. When a new potential federation partner is discovered and the initial connection is established, then a given federate sends the following set of information to its partner federate:

- List of its properties, such as ownership, mission, security clearance level, location, and so forth;
- List of metadata types the federate clients potentially intend to subscribe to or query about, with relative priority values attached to these metadata types;
- Matrix of values indicating preferences for using different possible adaptation methods on the connections between the federates.

Then each federate independently decides, based on its own local policies:

- Whether to federate with the remote partner;
- What priority to assign to the given remote federate;
- An estimate of the local server resources to devote to the remote federate (as a percentage

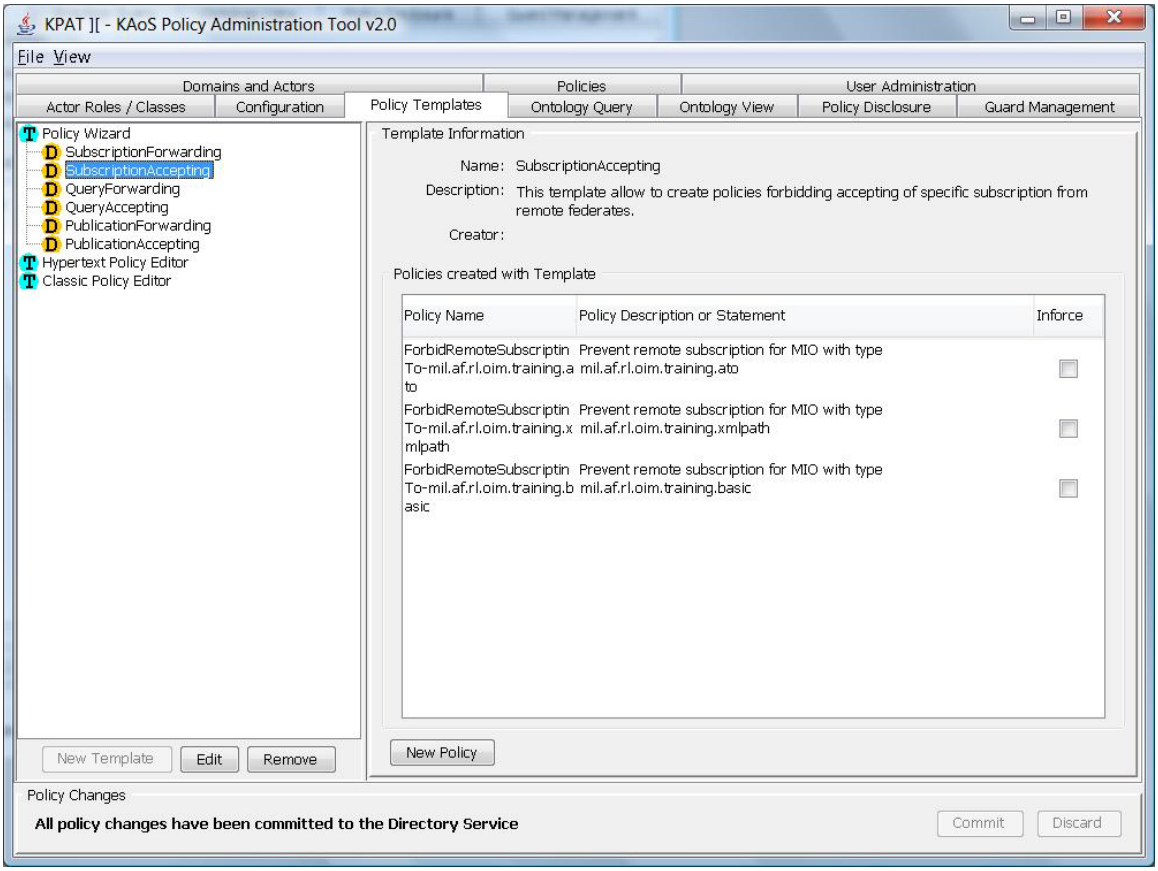

Figure 6: Federation Policy Templates in KPAT

of time), based on the current resource use for federation operations and the assigned federate priority;
- An estimate of the metadata type subscriptions and queries the local federate would be able to support for the given remote federate.

The last step above generates performance expectations that are then conveyed to the remote federate as part of the federation contract.

During the subsequent exchange of subscriptions, queries, and publications between federates, each operation is examined and analyzed with respect to the current policies. The policies can forbid the operation or modify it by changing the subscription or query predicate or trimming the metadata information in the published information object being forwarded to the remote federate.

Additionally policies and the agreed adaption matrix control when and which adaptation mechanism are activated when the share of resources used by the given federate exceeds the agreed limit.

The KAoS Guard is controlled (through the KAoS Directory Service) using KPAT, the graphical policy management tool. The KPAT configuration for the control of federation consists of sets of predefined policy templates and polices associated with them (Figure 6). These policies can be easily activated and deactivated. The policy templates are grouped into four categories:

- Federation Acceptance Polices,
- Gatekeeping Policies,
- Adaptation Policies,
- Contract Policies.

6. Cross-domain Information Sharing

We proposed a services-based approach to address the challenges with cross-domain information sharing. With the ever changing cross domain requirements, especially with coalition partners, a services-based approach would be ideal. There is a big push from the DoD and IC to do cross domain information sharing using Service Oriented Architecture, where a cross domain guard or any other Cross Domain Solution is nothing more than a service provider. The service that it is providing is to ensure only necessary information is sent from one domain to another.

The difference between the new services-based approach and the traditional CDS is that the cross domain service can and should be further divided into sub-services. That is, a traditional CDS will make an approve/deny decision at the end about whether the data or document is going to another domain. Prior to the decision phase, however, there is a sequence of processing that has to be done. For instance, the sender will have to be authenticated and authorized; the data will have to go through a virus scan, a file type check, and so on. Depending on the file type, additional checking may also be required. In the services-based approach, each of those processes will then become a stand-alone sub-service. Each of the sub-service will do its job and contribute its piece to the overall service for the guard to make a decision at the end.

One of the advantages of this approach is that each of the services can be considered as a stand-alone service and can be certified individually. One of the problems with the traditional CDS was that the Certification and Accreditation (C&A) process of the entire CDS typically takes 18 to 24 months. Any change to the CDS, (e.g., change to support a new cross-domain requirement), would require a new C&A. On the contrary, if services are certified individually and one of the services has to be modified to support a new requirement, only that service will have to be recertified instead of the entire system. This services-based approach would significantly decrease the C&A time.

Another advantage is that services can be decoupled and developed separately. For a traditional CDS, it is usually developed by a single vendor. What that means is that if there is a change of requirements or if additional tasks needed to be done, that usually requires significant engineering support from the vendor and the turnaround time could be months before the new capabilities are developed. With the services-based approach, services can be developed by others, who have expertise in a particular area, but not necessarily the CDS vendor themselves. Services developed by experts in the field would certainly improve the quality. As a result, the quality of the overall cross domain service would increase and the turnaround time would decrease.

Certified services can be grouped together in a policy to support specific cross domain requirements. As this service-based approach becomes more mature, an accredited system could have multiple policies in place. Depending on the situation or as the

requirements change, the right policy can be selected dynamically without any service interruption. For instance, a CDS could be loaded with two sets of policies, one to be used during normal operations and one could be used if the CDS senses that it is under a denial of service attack. If the CDS is under attack, the emergency policy could include an additional notification service to send an alert to an appropriate user, or reroute the data to a backup server, which none of these services are required during normal operations.

This services-based approach adds significant benefits to information sharing with our coalition partners. During operations where friend becomes foe in a split second, this approach enables CDS to be adaptive. As coalition partners come and go, predefined policies or certified services can be added or removed, depending on the situation, relatively easier and quicker than before to support new cross domain requirements. As a result, we can effectively share information with our coalition partners while maintaining the same high level of assurance.

7. Summary / Conclusions

We have described an infrastructure (Phoenix) and a series of capabilities (Federation Services, Policy Services, and Cross Domain Information Sharing Services) that together can help address the challenges of information sharing for coalition operations. In particular, the combination of the federation and policy capabilities addresses the basic need for the controlled sharing of information across administrative and security domains. Such a services-based approach increases the flexibility and reduces the time to certification and deployment of a Cross Domain Solution to support coalitions. The requirement for a Cross Domain Guard, which effectively keeps coalition networks partitioned, complicates the technical challenges of integration. Certain aspects of the proposed architecture, such as channels that interconnect services and components, would need to be extended to work across a CDG. We hope that, by following such an approach, we can quickly extend the benefits of net-centric operations to coalition settings.

8. References

[1] Nooney, M. Experiences of Combat Operations in Afghanistan. Invited Keynote Talk at Knowledge Systems for Coalition Operations (KSCO 2009).

[2] Unified Cross Domain Management Office (UCDMO) Web Site: http://www.ucdmo.gov/index.html (retrieved on July 31, 2010).

[3] Gerber, C. Dot-Connecting Across Domains. In Military Information Technology, Vol. 14, No. 1 (February 2010), pp. 6-8.

[4] Linderman, M., et. al., "A Reference Model for Information Management to Support Coalition Information Sharing Needs", In Proceedings of 10th International Command and Control Research and Technology Symposium, 2005 http://www.dodccrp.org/events/10th_ICCRTS/CD/papers/274.pdf.

[5] Suri, N, Marcon, M., Quitadamo, R., Rebeschini, M., Arguedas, M., Stabellini, S., Tortonesi, M., Stefanelli, C. An Adaptive and Efficient Peer-to-Peer Service-oriented Architecture for MANET Environments with Agile Computing. In Proceedings of the Second IEEE Workshop on Autonomic Computing and Network Management (ACNM'08).

[6] Carvalho, M., Suri, N., Arguedas, M., Rebeschini, M., and Breedy, M. A Cross-Layer Communications Framework for Tactical Environments. In Proceedings of the 2006 IEEE Military Communications Conference (MILCOM 2006), October 2006, Washington, D.C.

[7] Loyal J. P., Carvalho, M., Martignoni III A., Schmidth, D., Sinclair, A., Gillen, M., Edmonson J., Bunch, L., Corman, D. QoS Enabled Dissemination of Managed Information Objects in a Publish – Subscribe – Query Information Broker. In Proceedings of the SPIE Conference on Defense Transformation and Net-Centric Systems 2009.

[8] Uszok, A., Bradshaw, J., Lott, J. Breedy, M., Bunch, L., Feltovich, P., Johnson, M. and Jung, H., (2008). New Developments in Ontology-Based Policy Management: Increasing the Practicality and Comprehensiveness of KAoS. In Proceedings of the IEEE Workshop on Policy 2008, IEEE Press.

[9] Bradshaw, J. M. (2008). How to do with OWL what people say you can't. Invited keynote. 2008 IEEE Conference on Policy, Palisades, NY, 2-4 June.