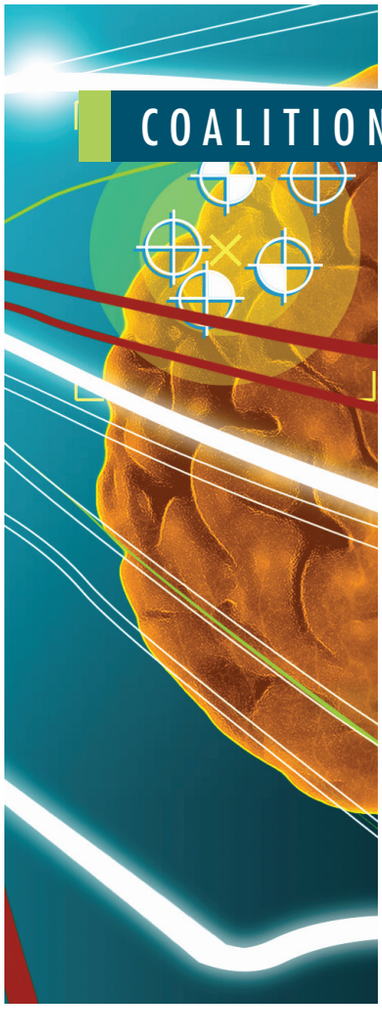

Extending Net-Centricity to Coalition Operations

Niranjan Suri, Andrzej Uszok, Rita Lenzi, Maggie Breedy, and Jeffrey M. Bradshaw,
Florida Institute for Human and Machine Cognition

Yat Fu, James Hanna, Vaughn T. Combs, Asher Sinclair, and Robert Grant,
US Air Force Research Laboratory

To bring the advantages of network-centric warfare to coalition warfighting, we must significantly improve our ability to quickly share critical information while still satisfying security requirements. Here, the authors explore a services-based approach to such information management.

Major military initiatives ranging from counterinsurgency operations such as in Afghanistan to disaster recovery work as in Haiti are increasingly coalition based. As cooperation and coordination levels continue to increase, so does the demand for timely information exchange. Individual nations have recognized the advantages of net-centricity (known as *network-centric warfare* in the US and *network-enabled capability* in the UK) and are applying it within single-nation forces. However, these concepts have yet to be fully realized for coalitions. Examples from previous operations[1] demonstrate the inefficiencies and hazards resulting from poor information sharing among coalitions. Such operations can only benefit from net-centricity if we extend it beyond a single nation to encompass multiple nations.

We can categorize net-centricity challenges as technical or security based. Technical challenges include interconnectivity, discovery, syntax, and semantics. Security challenges primarily involve protecting restricted (such as classified) information, data sources, and the methods used to obtain the information. These security challenges further exacerbate the technical challenges. For example, the interconnectivity problem at tactical edge networks occurs because of differences in radio standards, frequencies, and cryptography. So, 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.

Here, we explore a services-based approach to information management (IM) to address the challenges inherent in 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. Many of the security challenges associated with coalition IM are also present in cross-domain IM, and we describe advances made in this area. Combining all these capabilities is an effective approach to supporting net-centric operations that span coalitions.

1541-1672/13/$31.00 © 2013 IEEE
Published by the IEEE Computer Society

State of the Practice

Sharing information across coalitions is complicated by established security requirements. 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 coalition partners. The CDG is a certified, trusted computing device that lets only certain types of information pass through from one network to another. For example, the Radiant-Mercury system, originally developed at Lockheed Martin under a US Navy contract, is a certified software application that runs on a trusted platform. The Unified Cross Domain Management Office (www.ucdmo.gov/index.html) in the US coordinates the efforts of developing and certifying CDGs and cross-domain solutions (CDSs) in general. A survey of the currently available systems and their capabilities is available elsewhere.[2]

CDSs are also used when interconnecting networks of different classification levels, even though such networks might belong to the same country. When the information flow is only from a lower classification domain to a higher one, the CDS might be simpler and consist of a data pump or a data diode, which lets data flow in only one direction. Typically, the CDS checks the data to ensure that there is no malicious content (such as a virus embedded in a document) prior to transferring it from the lower classification domain to the higher one.

In the example in Figure 1, any information passing from coalition partner 1 (for example, the US) to coalition partner 2 (the UK) will pass through a CDS. Given that each country has its own security concerns, we want to share just the required

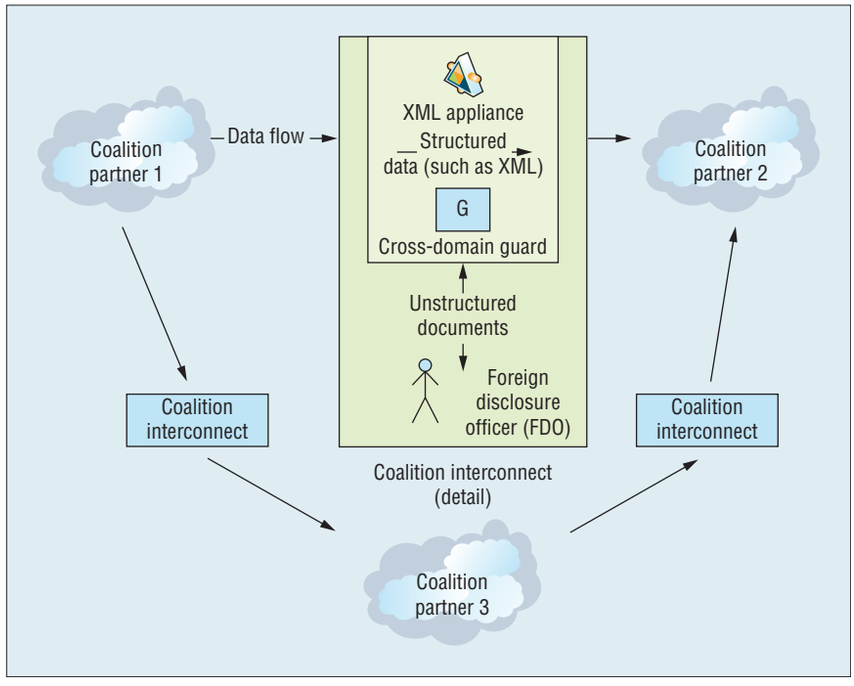

Figure 1. Current approach to coalition information sharing. This method uses a cross-domain guard (CDG) to allow only certain information types to pass from one network to another.

information with our coalition partners, while still protecting our networks. In some cases, additional technical solutions or guarding devices must be put in place. For example, each country might have its own CDS, which means that all information will pass through two guards, one on the US side and one on the coalition side, before the information is transferred to our coalition partners and vice versa.

CDSs process information differently depending on its type. For example, structured information (such as XML) might be amenable to automated processing. An XML appliance could be used in the CDS to automatically manipulate XML messages and pass them 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 information's criticality, the processing time could vary from minutes to days. Intelligent text-analysis tools can assist an FDO by highlighting areas of concern, thereby speeding up the review process. However, current security regulations demand that a human be the decision-making authority.

These security and technical challenges create a barrier to effectively sharing information with coalition partners. For some military operations, which often require agile and dynamic responses, coalition information sharing would be a nice addition, but most likely won't be available or considered. Two primary issues are CDS costs and the time involved in certification. An added complexity is that changes, such as in the type of information flowing through the CDS, might require redevelopment and recertification, which would prevent a quick response to addressing information needs for rapidly evolving missions. This, in essence, is the problem we must address before network-centric operations can span coalition forces.

One approach to reducing the time involved in establishing a CDS is to

decompose its functionality into a set of (composable) services that we could then certify individually. Deploying a new CDS, or modifying an existing one, would require recertification of only those services affected, which could be faster and less expensive. Here, we describe this services-based approach, using the Phoenix infrastructure for IM.

Phoenix: Service-Oriented Information Management

Service-oriented architectures (SOAs)[3] are beneficial to designing modern distributed systems. First and foremost, service orientation lets us decompose business processes naturally into a well-defined and orchestrated set of services that encapsulate and export access to cohesive and modular functionality. This approach enhances efficiency by letting us potentially reuse services among disparate business processes and orchestrations. Furthermore, services-based approaches promote using service discovery mechanisms and brokering services that naturally support late and dynamic binding of applications to compatible and available services.

SOAs let us integrate disparate collections of software and hardware—an important requirement with coalition operations. Services naturally support policy enforcement at many levels within the architecture to dynamically control IM and information dissemination among coalition partners. For example, administrators can change whether specific information should be shared with a coalition partner dynamically and push it to the appropriate enforcement points within the service orchestration. Furthermore, based on policy, information might be sanitized using a filter so that only appropriate portions of a document are disseminated between federated collections of coalition services and applications.

To provide an essential piece of the envisioned net-centric IM solution, the US Air Force Research Laboratory (AFRL) developed a reference set of IM services (IMS), known as project Phoenix, for the Department of Defense (DoD). We define IM as "a set of intentional activities to maximize the value of information for achieving the objectives of the enterprise."[4] These 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 a wide spectrum of scalability and performance requirements.

Phoenix is flexible in its design such that it can be deployed in both enterprise and tactical environments. Enterprise environments are resource-endowed (in terms of processing, power, storage, and network bandwidth). Tactical environments are resource-constrained, particularly with very little bandwidth over wireless communications with tactical radios. AFRL designed and implemented Phoenix as a flexible prototype that other research (such as federation) can leverage.

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 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. These services support flexible and extensible definitions of session, service, and channel contexts that let us apply quality-of-service (QoS) and security policies at many levels within the SOA.

Federation Services

The federation architecture lets us seamlessly and securely integrate multiple information spaces, or *federates*. "Seamless" implies that the architecture supports automatic discovery of and interconnection between federates. The federation process is transparent to clients, which still connect to their home federates as usual. "Secure" implies that the federation process isn't arbitrary and open. The establishment of federation and information exchange is controlled via policies.

Although the notion of federation initially supported interconnection of information enclaves in a single nation, we can extend the same concept to coalition needs. We can regard each coalition partner as a federate, with federation services providing the desired controlled information sharing between them. Such services support many needs inherent in CDSs, including policy-controlled information sharing and dynamic information transformation.

All federates in the federation architecture are peers. Each federate independently manages its connection with other federates and has its own policies governing information exchange with them. This approach is particularly well suited to coalition scenarios, given that each coalition partner (and hence each federate) is a separate administrative domain.

Following the services-based approach, we realize the federation capability using a set of services that work in conjunction. Figure 2 shows the key federation services and the interconnection between four federates. In this configuration, Federate One is independently connected to Federates

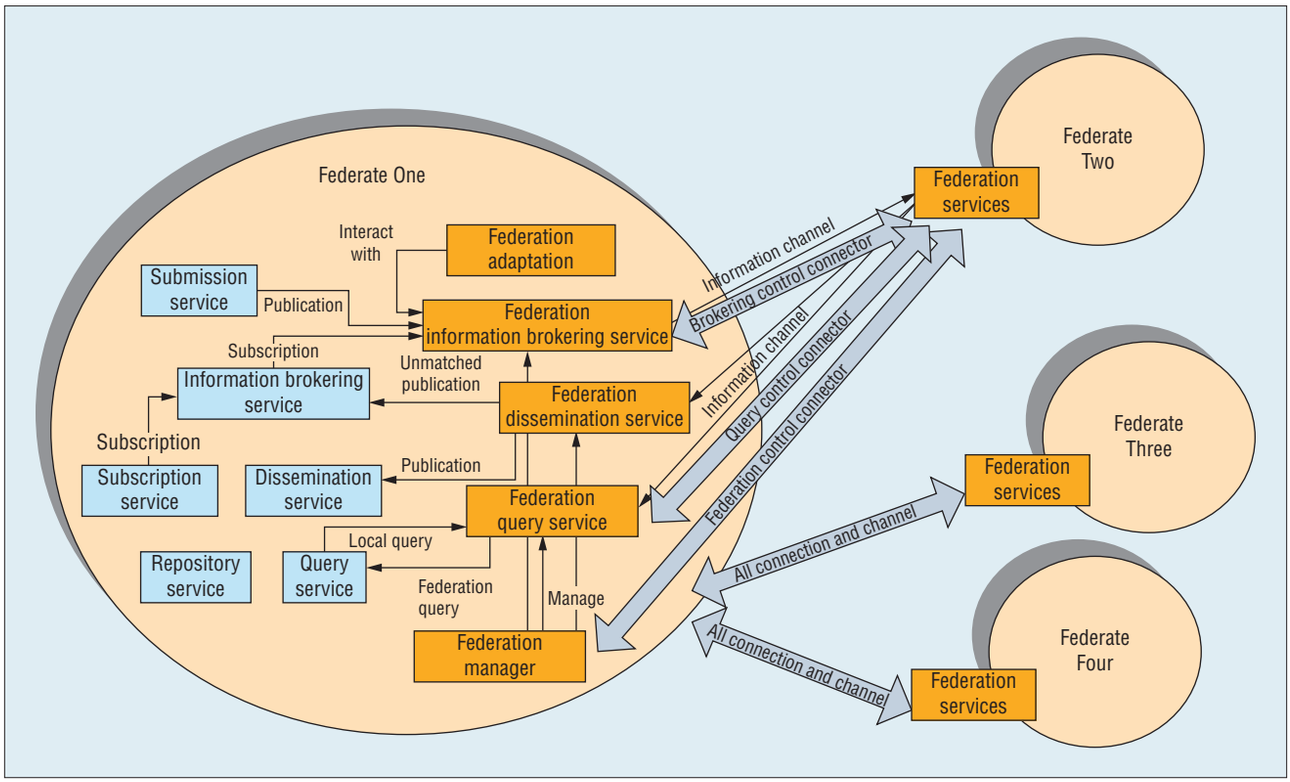

Figure 2. Federation services and their interaction with Phoenix services. Using a services-based approach provides flexibility for rapid customization and adaptation for net-centric information management.

Two, Three, and Four. For simplicity, we collapse the connection and channels to Federates Three and Four and show only the federation services inside Federate One.

Discovery Manager

Federation begins when the local federate finds one or more remote federates. Federates can be found via static configuration (where the end points are specified) or via a dynamic discovery process. The *discovery manager* (DM) provides the necessary discovery functionalities for automatically finding other federates in the network. The discovery process can rely either on the *group manager*'s capabilities[5] or on the *cross-layer communication substrate* (XLayer)[6] for discovery and grouping support. With either system, discovery relies on some variation of a broadcast or multicast at the network layer. When operating across coalition networks interconnected with a CDS, such discovery wouldn't be possible because the network-level communication would be blocked. So, automated discovery might need to be set aside, and the system will need to resort to a predefined configuration.

Federation Manager Service

Once the discovery manager identifies potential new federates, the *federation manager* (FM) sets up the federation across the newly discovered entities. In particular, the FM communicates with the new federates, informs the other federation services about them, and negotiates contracts. The FM also handles disconnections and terminates federation.

Federation Information Broker

Information brokering is a fundamental service in Phoenix. Brokering involves examining new, incoming information that's 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 other federates. It also receives the local publications from the *submission service*, brokers them locally on remote federates' behalf, and forwards them to appropriate federates. In particular, it forwards them to remote *federation dissemination services* (FDSs).

Federation Dissemination Service

Dissemination is the post-processing step that follows brokering, and it involves transmitting matched information to clients. The dissemination

service normally receives matched data from the *local information brokering service*. When federation is involved, the *federation dissemination service* (FDS) receives matched information from remote federates that's destined for local clients. In most cases, when the FDS receives forwarded publications from remote federates, they've already been matched to specific local clients (by the remote federation information broker). In such cases, the FDS uses the local dissemination service to transmit the data to clients. Otherwise, it uses the local information broker to publish the information locally.

Federation Query Service

Querying for archived information compliments publish and subscribe as the third primitive operation Phoenix provides in an IM context. Query differs from subscribe in that it can retrieve previously published and stored data. The *federation query service* (FQS) permits information retrieval from the client's data stores and supports synchronous and asynchronous query execution. The *repository service* manages data stores, which can be one of two types:

- *Repositories* are low-latency, high-access data stores that should support higher data read and write rates.
- *Archives* store much more data than repositories, but with a lower data access rate.

The FQS extends the query capability to remote federates. It receives local queries and sends them for processing to both the remote federates and the local query service, collects the results, and returns them to the client. The FQS assumes that federates don't have duplicated data, which simplifies the distributed query problem. The FQS can locally cache data resulting from a remote query, thereby improving performance for repeated queries. The nature of the queries, as well as the FQS's behavior, can be controlled via policy. For example, a query from a coalition partner that's executed against a US database might be modified to limit its scope and nature. This control is independent from a federate's ability to control the individual objects that result from the query.

Federation Adaptation Service

During operations, the resources available for IM are likely to change over time. For example, the network links connecting federates might become saturated, or the systems hosting federation services could become overloaded. The *federation adaptation service* performs local adaptations to offset such resource shortages. For example, under low-bandwidth situations, the adaptation service can temporarily suspend low-priority subscriptions to provide reasonable performance for the remaining subscriptions. Clients or policies can specify the subscription priorities. 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, where the brokering then occurs. The adaptation service sorts subscriptions based on their hit rate (that is, the percentage of publications that match the predicate), and selects the subscription with the highest hit rate first. This minimizes the impact of an increase in bandwidth usage caused by this adaptation.

For the adaptation service to perform its task, the systems' and networks' underlying resources must be monitored. The adaptation service relies on an underlying *monitoring service*[7] to receive information about the system.

A Complete Scenario

To better illustrate how federation services operate, we consider a complete scenario, from discovery to establishment to shutdown. To simplify, we use a scenario in which federation occurs between only two instances of an IMS, Federate One and Federate Two.

Federation establishment. When the federation service (FS) is instantiated along with other Phoenix IM services, the first step is registering with the DM. By registering and joining a predefined group, the IMS signals its intention to be part of the federation. Once this 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 (end point) to itself. This contains all the information necessary to create a *stub* connected to the other federate—that is, the IP address, the port number, and the names for the services the federate can provide.

Eventually, a contract negotiation occurs, and once both nodes accept the contract, the federation is officially established. The local FS, FIBS, and FQS establish control channels with the corresponding peers in the remote federate. Publications and query results are transmitted over an information channel.

Subscription forwarding. When a client connected to Federate One issues a subscription to its local IMS, the FIBS captures the request via the local subscription service and the local information brokering service. It then forwards the subscription 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), the FIBS intercepts the publication. Under normal conditions (for instance, 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 might have stored in its remote subscription table. Publications for which the local matching succeeds are marked as matched, and sent to Federate Two via a communication channel. Federate Two receives the publication, verifies whether it was already matched (and, if not, matches it with the local subscriptions), and forwards it to the IMS. Finally, the IMS delivers the publication to the correct subscriber clients.

Federation termination. Federation lasts until at least one node 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 subscriptions. The system is now back in its initial state, prior to federation establishment.

Policy-Based Control

The federation operational behavior detailed in the previous section is entirely governed by policies. Before performing any step in its execution flow, the FS verifies with the policy framework whether and how the current operation is allowed.

KAoS is a set of platform-independent services that lets users define policies to ensure adequate security, configuration, predictability, and controllability for various distributed agent-based systems.[8] KAoS domain services let the system semantically describe and structure groups of software components, people, resources, roles, and other entities into domains and subdomains to facilitate collaboration and external policy administration. KAoS policy services allow for specifying, managing, enforcing, and resolving conflicts in policies within domains. KAoS policies distinguish between authorizations (that is, constraints that permit or forbid some action by an actor or group in some context) and obligations (constraints that require some action to be performed when a state- or event-based trigger occurs, or else serve to waive such a requirement).

KAoS policy services allow for specifying, managing, enforcing, and resolving conflicts in policies within domains.

Policies are represented in ontologies, not rules. Using ontologies—encoded in the Web Ontology Language (OWL; www.w3.org/TR/owl-features/)—to represent policies lets KAoS reason about the controlled environment, policy relations and disclosure, policy conflict resolution, and domain structure and concepts. KAoS reasoning methods exploit description-logic-based subsumption and instance classification algorithms and, if necessary, controlled extensions to description logic (for example, role-value maps). Unfortunately, many myths have been propagated about OWL's limitations for policy management—for our case, we've found it an extremely expressive, flexible, and efficient alternative.[9]

Policy administration and enforcement in KAoS can occur in a centralized or distributed manner. This flexibility is important for federation because each federate might well be in a different administrative domain that requires its own policy specification and enforcement.

The FS is integrated with KAoS, which controls each federate's establishment, life cycle, information exchange, and adaptation. When a new potential federation partner is discovered and the initial connection is established, each federate sends the following information to its partner federates:

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

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

- decides whether to federate with the remote partner;
- decides what priority to assign to the given remote federate;
- estimates the local resources needed to devote to the remote federate (as a percentage of time), based on the current resource use for federation operations and the assigned federate priority; and
- determines the metadata type subscriptions and queries that the local

federate would be able to support for the given remote federate.

During the subsequent exchange of subscriptions, queries, and publications between federates, the FS examines and analyzes each operation 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.

Cross-Domain Information Sharing

With the ever-changing cross-domain requirements, especially with coalition partners, a services-based approach would be ideal. Currently, a push exists within the DoD to enable cross-domain information sharing using SOAs, where a CDG or any other CDS is nothing more than a service provider, ensuring that only necessary information is sent from one domain to another.

The difference between a services-based approach and a traditional CDS is that the cross-domain service can and should be further divided into subservices. That is, a traditional CDS will make an approve/deny decision at the end regarding whether to share the data or document with another domain. Prior to the decision phase, however, a processing sequence must occur. For instance, the sender must be authenticated and authorized, and the data must go through a virus scan, a file type check, and so on. Depending on the file type, additional checking might also be necessary. In the services-based approach, all these processes will become stand-alone subservices, each of which will do its job and contribute its capability to the overall service so the guard can make a decision at the end.

Using this approach, each service can stand alone and can be certified individually. In the traditional CDS, the certification and accreditation (C&A) process of the entire CDS typically takes 18 to 24 months. Any change to the CDS (such as a change to support a new cross-domain requirement) would require a new C&A. If services are certified individually, however, and a service needs modification to support a new requirement, only that service will have to be recertified. This services-based approach would significantly decrease the C&A time.

Another advantage is that services can be decoupled and developed separately. A traditional CDS is usually developed by a single vendor, so if requirements change or tasks are added, significant engineering support is required from the vendor. It could be months before the new capabilities are developed. With the services-based approach, services can be developed by those with expertise in a particular area, not necessarily the CDS vendor. Services developed by experts in the field would certainly decrease turnaround time and improve quality, improving the overall cross-domain service's quality as well.

Our approach enables grouping certified services 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, administrators can select the right policy dynamically without any service interruption. For instance, a CDS could be loaded with two sets of policies, one for use during normal operations and one for use if the CDS senses that it's experiencing a denial-of-service attack. If the CDS is under attack, the emergency policy could include an additional notification service to alert an appropriate user or reroute the data to a backup server, whereas none of these services are required during normal operations.

Our approach adds significant benefits to information sharing among coalition partners. During operations in which trust relationships can change rapidly, this approach enables CDSs to be adaptive. As coalition partners come and go, administrators can add or remove predefined policies or certified services, depending on the situation, more quickly and easily than other existing solutions to support new cross-domain requirements. Consequently, we can effectively share information with our coalition partners while maintaining the same high level of assurance.

We've described an infrastructure (Phoenix) and a series of capabilities that can help address the challenges of information sharing for coalition operations. The requirement for a CDG, which effectively keeps coalition networks partitioned, complicates the technical integration challenges. We would need to extend certain aspects of our proposed architecture, such as channels that interconnect services and components, 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.

Acknowledgments

Massimiliano Marcon, previously at the Florida Institute for Human and Machine Cognition, and Robert Hillman, previously at the US Air Force Research Laboratory, also contributed to this effort.

THE AUTHORS

Niranjan Suri is a research scientist at the Florida Institute for Human and Machine Cognition (IHMC) and a visiting scientist at the US Army Research Laboratory (ARL). His current research focuses on agile computing—applying opportunistic computing to distributed systems to create adaptive and resilient systems. Suri has a PhD in computer science from the University of Lancaster, UK. Contact him at nsuri@ihmc.us.

Andrzej Uszok is a research scientist at the Florida Institute for Human and Machine Cognition (IHMC). His research interests include ontologies, semantic reasoning, policy specification, agent systems, and transparent interoperability. Uszok has a PhD in computer science from the AGH University of Science and Technology, Krakow. He's a member of IEEE and ACM. Contact him at auszok@ihmc.us.

Rita Lenzi is a research associate at the Florida Institute for Human and Machine Cognition (IHMC). Her research interests include information management, federation, and service-oriented architectures. Lenzi has an MS in computer science from the University of Modena and Reggio Emilia, Italy. Contact her at rlenzi@ihmc.us.

Maggie Breedy is a research associate at the Florida Institute for Human and Machine Cognition (IHMC). Her research interests include agile computing and KAoS policy-based services for agent management, collaboration, and human-agent teamwork. Breedy has an MS in computer science from the University of West Florida. Contact her at mbreedy@ihmc.us.

Jeffrey M. Bradshaw is a senior research scientist at the Florida Institute for Human and Machine Cognition (IHMC), where he participates in the research group developing KAoS Policy and Domain Services, the Luna Software Agent Framework, and the Sol Cyber Framework. Bradshaw has a PhD in cognitive science from the University of Washington. He's coeditor of the Human-Centered Computing department for *IEEE Intelligent Systems*. Contact him at jbradshaw@ihmc.us.

Yat Fu is the lead engineer at the US Department of Defense Cross Domain Management Office (DCDMO) at the US Air Force Research Laboratory. He provides subject matter expertise on cross-domain technologies to the Defense Intelligence Agency (DIA) and other organizations within the intelligence community. Contact him at yat.fu@rl.af.mil.

James Hanna is a senior research engineer at the US Air Force Research Laboratory Information Directorate. His research interests include distributed information management, distributed policy enforcement, and advanced architectures to address the complex challenges of deploying information management in forward tactical environments. Hanna has a BS in computer science from the State University of New York (SUNY) Institute of Technology. Contact him at james.hanna@rl.af.mil.

Vaughn T. Combs is a senior engineer at the US Air Force Research Laboratory and the lead software architect and senior developer serving within the Operational Information Management (OIM) group. His research interests include distributed computing systems, resource management, fault tolerance and information management in enterprises, and tactical systems. Combs has an MS in electrical and computer engineering from Clarkson University. Contact him at vaughn.combs@rl.af.mil.

Asher Sinclair is a senior program manager at the US Air Force Research Laboratory Information Directorate, where he works in the Information Management Technologies branch (RISA). His work history includes research and development in enterprise systems management, service-oriented architectures, and cybersecurity. Sinclair has an MS in information management from Syracuse University. Contact him at asher.sinclair@rl.af.mil.

Robert Grant is an engineer at the US Air Force Research Laboratory Information Directorate and is pursuing an MS in computer science from Syracuse University. His latest research is in secure mobile application services. Grant has a BA in English from the University at Buffalo and a BA in computer science from Oswego State. Contact him at robert.grant@rl.af.mil.

References

1. M. Nooney, "Experiences of Combat Operations in Afghanistan," *Knowledge Systems for Coalition Operations* (KSCO 09), invited keynote, 2009.
2. C. Gerber, "Dot-Connecting Across Domains," *Military Information Technology*, vol. 14, no. 1, 2010, pp. 6–8.
3. T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall, 2009.
4. M. Linderman et al., "A Reference Model for Information Management to Support Coalition Information Sharing Needs," *Proc. 10th Int'l Command and Control Research and Technology Symp.*, US Dept. of Defense, 2005; www.dodccrp .org/events/ 10th_ ICCRTS/CD/papers/274.pdf.
5. N. Suri et al., "An Adaptive and Efficient Peer-to-Peer Service-Oriented Architecture for MANET Environments with Agile Computing," *Proc. 2nd IEEE Workshop Autonomic Computing and Network Management* (ACNM 08), IEEE, 2008, pp. 364–371.
6. M. Carvalho et al., "A Cross-Layer Communications Framework for Tactical Environments," *Proc. 2006 IEEE Military Comm. Conf.* (MILCOM 06), IEEE, 2006, pp. 1–7.
7. J.P. Loyal et al., "QoS Enabled Dissemination of Managed Information Objects in a Publish-Subscribe-Query Information Broker," *Proc. SPIE Conf. Defense Transformation and Net-Centric Systems*, SPIE, 2009; doi:10.1117/12.818744.
8. A. Uszok et al., "New Developments in Ontology-Based Policy Management: Increasing the Practicality and Comprehensiveness of KAoS," *Proc. IEEE Workshop on Policy 2008*, IEEE, 2008, pp. 145–152.
9. J.M. Bradshaw et al., "How to Do with OWL What People Say You Can't," *2008 IEEE Conf. Policy*, invited keynote, 2008.