

Federation as a Service

Author(s)

Address(es)

Email address(es)

ABSTRACT

Federation of information systems is a critical capability for service-oriented computing systems that support information distributed across heterogeneous domains. Although service-oriented architectures provide data services that represent information systems such as database systems, federation architectures provide services for integrating heterogeneous and distributed information systems to provide a single-system view of the information to a federation. At the same time, a federation maintains the anonymity and autonomy of each administrative domain and its clients, and clients interact with their local information system and services to produce and consume information. The storage and dissemination of information is managed through cooperation of the federation and the individual information systems (federates) participating in the federation.

In this paper we present a services-based architecture for information federation and describe how the components and capabilities of that architecture participate in and interoperate with service-oriented architectures (SOAs). In particular, we discuss approaches to incorporating federation as a service for SOA, integrating heterogeneous SOA-based enterprise systems using federation, and integrating SOA-based systems with non-SOA systems through federation. We analyze the pros and cons for distributing federation services and examine a use case integrating systems with different performance characteristics. Finally, we discuss a number of services that could be used in conjunction with federation to enhance information interoperability. Throughout we motivate the need for federation as a distinct service in services-based systems of systems.

Categories and Subject Descriptors

D.2.12 [Software Engineering]: Interoperability;
H.2.5 [Database Management]: Heterogeneous Databases

General Terms

Design, standardization

Keywords

Information federation, data service, SOA

1. INTRODUCTION

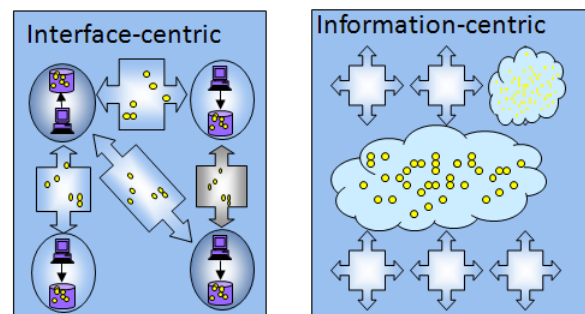
Service-Oriented Architecture (SOA) is the newest instantiation of *interface-centric* interoperability, building upon the foundations of distributed object computing (e.g., CORBA [1] and distributed components (e.g., CORBA Component Model, CCM [2]). In interface-centric systems, interoperability is based on point-to-point referencing in which a client locates “who” it wants to talk to (e.g., through service discovery and URIs) and then accesses an exposed interface (e.g., service invocation). The interoperation is based on a reference or address and operations, not on the information provided or needed.

A complementary, but alternative, view is information-centric interoperability such as that supported by information services [3], as shown in Figure 1. In information services, clients that have information make it available (e.g., through *publish* operations) and clients that need information request it (e.g., through *subscribe* or *query* operations) based on the information’s type, attributes, and content. Interoperation is based on active management to locate, broker, and administer information.

In SOA environments, a core set of information services provide critical functions for publishing, brokering, and disseminating information between decoupled clients. However, these basic services only handle information management within an administrative domain, i.e., for clients under the control of a single information system. To support information that is distributed across heterogeneous, autonomous domains, such as those of disjoint organizations, coalition partners, or security domains, information systems need a set of services for *information federation*.

This paper describes a set of federation services we have developed to enable information exchange between autonomous information systems. These services provide the ability to establish information exchange between systems (i.e., create a federation of information systems), provide federation visibility to information requests (subscriptions and queries), identify systems or services in the federation that might service requests, and communicate published or archived information in response to requests.

The focus on *information* discovery and management is the major contribution of federation to service-oriented systems. In particular, to maintain the autonomy of its constituent systems a federation will leave decisions of how, and how much, information is shared in the hands of the policy makers and managers for the local information systems. This leads to the following key contributions of federation:



(a) Interface-centric systems interoperate based on address and operations

(b) Information-centric systems interoperate based on active management to locate, broker, and administer information

Figure 1. Information systems provide an alternative, but complementary, interoperation pattern to service-oriented architecture systems.

- Information publishers and consumers can be decoupled and anonymous from one another. That is, a publisher does not necessarily need to know who is interested in the information and a consumer does not necessarily need to identify a specific information source.
- Clients of an information system do not need to know whether their system is part of a federation. That is, they can produce and consume information with their local information broker (whether or not it is part of a federation) and exchange information with clients of other federates without being aware of federation.
- Policies for information management can be provided by system managers or control authorities, and information service agreements can be negotiated between federates to control their interaction.
- Information systems can control access to their own information, e.g., security services provide access control and information protection for federates.

Indeed, in the federation model an information system is itself treated as a service that requests (subscriptions and queries) and provides information to the federation (publications and query responses).

In the next section we describe an architecture for federation services and a core set of services supporting information interchange, and identify additional services useful for federation. In Section 3 we discuss how the addition of federation and key services to a SOA can be used to integrate information in multiple systems in an enterprise. We extend this to cross-organizational federations in Section 4. In Section 5 we discuss service distribution and describe how federation enables combinations of systems with different performance characteristics. Finally, in Section 6 we summarize our contributions and discuss future directions.

2. INFORMATION FEDERATION ARCHITECTURE

An *information system (IS)* [4] supports *clients* who produce and consume information; it helps clients communicate anonymously with other clients and may store information persistently. A system *manager* monitors and controls the system. Note that an IS may, in fact, be a collection of services that support information management and thus IS could just as easily refer to information services. As it turns out, these definitions are equivalent in our model for information federation, and we generally use the term information system to recognize that federations may include legacy information systems (such as databases) as well as data or information services.

A *Federation* is a system of autonomous, yet cooperating information systems. An extensible, modular, and composable architecture of services for coordinating, configuring, and managing behavior of the federation is outlined in Figure 2. In this architecture each system (i.e., federate) is an independent administrative domain with its own processes, state, and configuration, and the federation is a separate administrative domain. A federation has rules and procedures for federates' behavior when participating in the federation, and provides *federation services* that guide and assist federates' behaviors within the federation. The services shown at the top of the figure are examples of services that might be provided by a federation.

Services required for federation include those that support communication of information between information systems and services that support administration and management of the federation as a collection of information systems, i.e.:

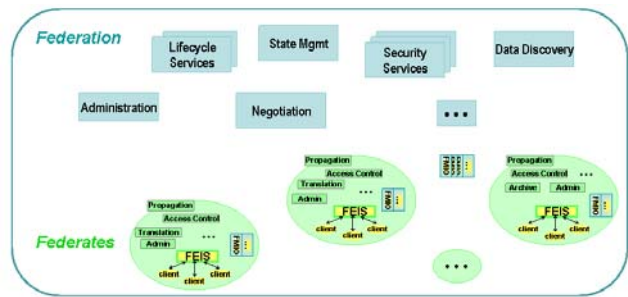


Figure 2. Services Architecture for Information System Federation

- *Data Discovery services* for federates to find information and other federates within the federation. These services are the analog of service discovery in a SOA.
- *Lifecycle services* with capabilities to configure, initiate and dissolve federations, and manage the membership of federations.
- *State Management services* for managing information about the state of the federation such as its services, membership, members, and active policies.

Other services provide valuable capabilities that may or may not be required by a particular federation.¹ For example, Security services for federation could authorize the participation of individual federates, authenticate the origin of information and requests, or define and authorize specific interactions between federates. Query and Translation services supporting information semantics and interoperability are also expected to be needed in many federations.

Other required federation services provide the capabilities needed by a federate to participate in a federation. At minimum, each federate is associated with a Propagation Service that handles interactions between the federate and other components of the federation (such as other federates' Propagation Services, or federation services). Other services might be provided for local or federation use. For example, a federate could provide an Archiving Service that could be used by the federation for storage. We discuss other useful services in Section 2.2.

Federates and federation services are designed to be transparent to existing information systems' clients, maintain the publish/subscribe/query paradigm of information exchange, and enable each information system to maintain controlled access to its own information.

A *Federation-Enabled Information System (FEIS)* is an IS that is aware that it is part of a federated information system and thus exposes an interface for interacting with federate services, in particular for interacting with a Propagation Service. By separating IS-specific activities from the federation, the interface can be used in a compatibility layer (or "wrapper") for an existing IS (i.e., a legacy system) to allow it to participate in a federation.

The information objects managed by an individual IS are moved throughout the federation as *Federation Managed Information Objects (FMIOs)*. FMIOs maintain the integrity of local information objects by tagging them with metadata used for federation purposes. This includes metadata such as an ID of the information source and a federation-assigned unique identifier for the information object. This metadata allows the object to be

¹ Note that any of these services could be provided as centralized services or could be distributed among the federation members. We discuss distribution of services in Section 5.

managed within the federation. For example, the information ID helps each information system distinguish between its own information and objects it received from elsewhere in the federation. Federation metadata also provides a vehicle for the federation to maintain information provenance, i.e., information about who has handled or changed the information (perhaps for security purposes). For example, the federation could prohibit movement of FMIOs that do not have authentic source IDs. The federated metadata does not modify the local information object so Translation services may be necessary for one information system to make use of an information object from another system.

2.1 Core Services: Data Discovery and Propagation

We implemented a prototype federation with a basic set of interactions using a CORBA implementation for managing federation-level service interactions (distributed method calls). The basic interactions are illustrated in Figure 3 and listed below.

- FEIS ↔ Propagation: Each Propagation Service provides a unified view of the federation to its FEIS.
- Propagation ↔ Data Discovery: The Data Discovery Service advises the Propagation Service as to which federates should receive requests or information produced by its FEIS.
- Propagation ↔ Propagation: The Propagation Services associated with individual federates interact to process data and requests in the federation.

Our FEISs were built by adding, to an existing information system, the capabilities needed to interact with a Propagation Service. Some capabilities were added via IMS clients and some access the IS interface – in other words, we “wrapped” the system. This is described in more detail in [5].

Data Discovery is used to determine where information or requests can be sent or retrieved; it is the federation analog of service discovery. The complexity of Data Discovery varies depending on the characteristics of the data models of the different federates and on the performance desired for federated operations. For example, Data Discovery can be as simple as providing to a requesting federate the addresses of all other federates in a federation. On the other hand, it could entail matching operation predicates with detailed information about the information available at other federates to identify precisely which federates can satisfy an operation. A Data Discovery service will typically work with a type system for matching information with requests, so may require Translation services to interpret requests or convert disparate type descriptions.

The Propagation Service associated with a federate communicates with a federation Membership Service to provide information about the federate that will be used by Data Discovery, and communicates with Data Discovery to identify other federates for disseminating requests or information. When a Propagation Service receives a publish, subscribe, or query

(P/S/Q) request from its FEIS, it interacts with Data Discovery to determine which other federates or services it should contact to process the request. The Propagation Service then contacts the other services directly, establishing a session with each other federate to process the request. As a requestor, the primary function of the Propagation Service is to disseminate requests from its federate to multiple other federates and manage the collection of responses from possibly multiple federates to provide a coordinated response to the request to its local FEIS. As the recipient of a request from the federation (i.e., another Propagation Service in the basic system), a Propagation Service moves the request into its federate and, for queries and subscriptions, provides a response to the federation. Note that a Propagation Service must be able to manage multiple simultaneous requests from local clients presented through the local FEIS as well as concurrent incoming requests and responses from remote federates.

2.2 Additional Services

Extensibility is an important aspect of any services oriented architecture, since it is unrealistic to anticipate that a set of comprehensive services could be defined for all existing and envisioned federate, configuration, and administrative domains. The information federation architecture described above was designed to be extensible, both to support alternative implementations of services and to incorporate additional services. In this section, we briefly describe some additional services useful for federation that could be exposed for discovery and use in an enterprise services environment.

Security services validate information, sources and services. These include different access control description languages and associated evaluation engines, and different authentication models. Our prototype implements services for federate and service authentication (the Data Discovery and Propagation services maintain trustworthy identities that must be presented for all message and information exchange), for access control (XACML [6] policies define allowable inter-federate behavior based on federate identities, information types and/or operation types) and for information integrity (messages and information are hashed and signed when moved between services).

Policy services provide mechanisms for defining and enforcing federation policies for inter-federate and inter-service interactions. Examples of policy types include policies for access control that define how federates are allowed to interact or what services they may use, policies that govern quality of service for particular federates or interactions, and policies that control communication and network use. Our prototype currently manages security policies.

Quality of Service (QoS) management services include capabilities to dynamically monitor, control and differentiate federation operations to achieve predictable performance of federated operations. QoS management services also adapt federation configurations to particular platform, resource, or performance characteristics and needs, and thus provide introspection into federated components to gather information on performance, resource usage, and configuration options.

Query services manage the distribution and execution of requests and the aggregation of results. They provide a uniform and configurable way to manage federated queries, performing operations such as distributed join, distributed merge, and enforcing global properties such as sorting and aggregation of results. Our prototype includes a simple query service distributed to all federates that distributes queries to all federates and merges (but does not aggregate) results.

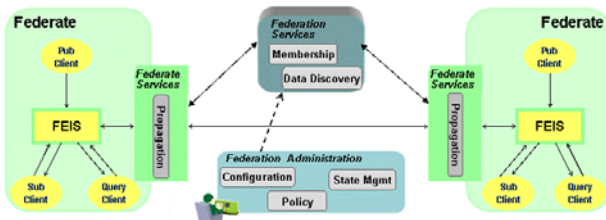


Figure 3. Basic service interactions.

Translation services mediate syntactic and semantic differences between the schemas understood by different federates. This is especially useful in cases in which federates are significantly different administrative domains, such as coalition partners or security domains. These services need the ability to discover the schemas that a federate and other translation services support and to incorporate and interpret semantic rules governing translation.

3. FEDERATION AND SOA

The federation architecture presented in Section 2 is a services-based *Information-Oriented Architecture*. Like a Service-Oriented Architecture it focuses on well-defined and partitioned services that can be invoked separately to achieve desired functionality and composed to achieve composite functionality. However, whereas a SOA focuses on distributed services, our federation architecture defines the services and service interactions needed to specifically support storage, access and dissemination of distributed information.

There are a variety of potential relationships between federates, clients, and a federation in the context of an Enterprise Service Bus (ESB) and Service Fabric underlying an instance of an SOA. For example:

- Information space federation can be implemented using features of a SOA environment, as shown in Figure 4a.
- Federation and federate services can be services offered in a SOA environment, as shown in Figure 4b.
- Federation and federate services can be used to federate different ESBs (and thus separate enterprises); see Section 4.
- ESBs and non-enterprise IMSs can co-exist via federation services; see Section 5.

In Figure 4a, an ESB and enterprise services are used in implementing a federation. In such an implementation, each federate registers itself with the ESB via its Propagation Service, clients connect to federates, and the federation services register with the ESB. The ESB in this case provides the connectivity between the federation and federate services, and the federation Data Discovery Service provides the discovery within the federation. In addition, enterprise services provided by the SOA

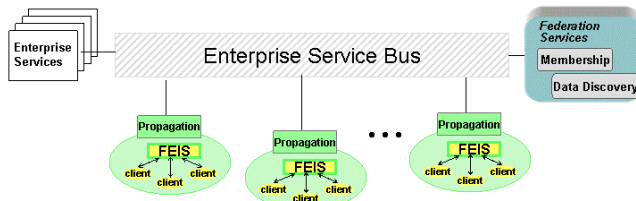


Figure 4a. Federation can be implemented using features of a SOA environment.

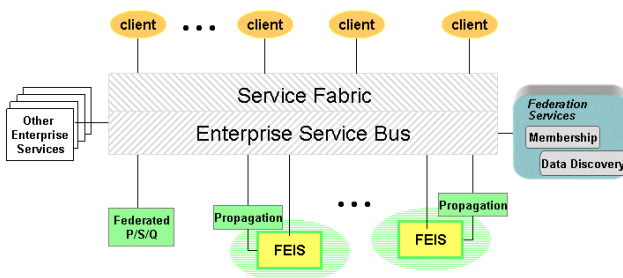


Figure 4b. Federation services augment a SOA environment.

could be used in the implementation of the federation services. For example, service discovery components of the SOA could support federation Membership by locating potential federates (i.e., locating Propagation services); conversely, Propagation services could use service discovery to locate the federation services. Similarly, State Management for the federation could be supported by the enterprise service registry. In addition, available ESB translation capabilities could assist in mediating interface and semantic differences in federates.

Figure 4b illustrates our federation architecture as an overlay to a SOA – i.e., federation is a service integrating information (data) services in the SOA treated as information systems in the federation. In the figure we show two enterprise data services “wrapped” as FEISs with Propagation services to connect them to the Federation services on the ESB. Each data service could participate only in the enterprise, only via federation, or in both ways as shown in the figure. Although a data service can itself provide an integrated view of underlying data sources to applications [7], federation offers a more dynamic approach to integration that relies on services rather than programmers to effect the information integration.

Within an enterprise services architecture, the federation architecture consists of federation and federate services, as described in Section 2, and additionally includes a Federated P/S/Q service to support enterprise information publication, subscription, and query. Recall that in our federation architecture we assume that all clients deal directly with an information system. The Federated P/S/Q service shown in Figure 4b plays this role in the SOA by exposing an IS-like interface for use by enterprise clients (or other enterprise services) to perform publish, subscribe or query operations in the federation. The service also exposes a Propagation-like interface for interacting with Federation services and other federates to execute those operations. Note that these interactions are exactly the role of a federate in the federation architecture; that is, the Federated P/S/Q service interacts as a federate within the federation. Unlike most federates, though, the P/S/Q Service does not store information so cannot respond to query or publish requests from the federation (i.e., from Propagation services). The federation registration process needs to provide the information Federation Data Discovery requires to determine that the service will not process P/S/Q requests from within the federation. On the other hand, the enterprise service discovery process would provide the Federated P/S/Q Service to clients, applications, and other services that are not part of the federation for use in interacting with the federation.

The federation, as a service to the enterprise, presents a single, virtual information system view of the multiple federates to the enterprise services environment. Enterprise clients will not know they are interacting with a federation; they have simply requested a service to support their publication, subscription and query requirements – i.e., they requested and are interacting with a data service. Although not shown in the picture, each federate may also have private clients, such as those shown in Figure 4a, who interact directly with the federate’s information system. In other words, these clients are not participating within the enterprise services environment. They may benefit from federation, though, if their FEIS chooses to federate their actions.

4. FEDERATING ESBs

Federation can also be used as a mechanism for interaction between enterprise systems hosted on different ESBs, as illustrated in Figure 5. In the figure, one enterprise is hosting a federation and each other enterprise (with ESBs 2 through n) presents itself to the federation as a federate. Each enterprise does

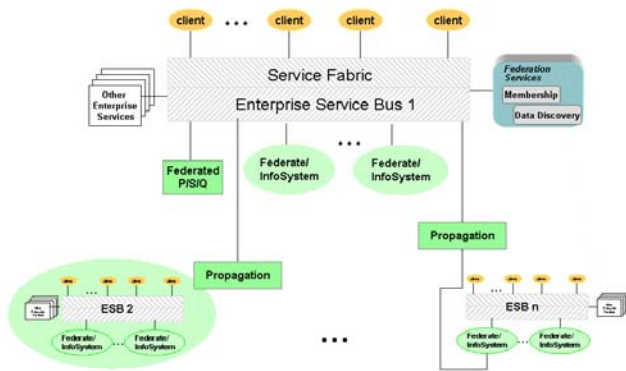


Figure 5. Federation can integrate different enterprises and information systems.

this through its interaction with a Propagation Service registered with Enterprise 1.

The way in which each Propagation Service is configured to work with its host enterprise depends on how that enterprise wishes to interact with the federation. For example, the Propagation Service can connect to the ESB (as shown in ESB-2) and present itself to that enterprise as a data service or other Information System that can accept and process P/S/Q requests. In addition, if the host enterprise is willing to process requests from other enterprise systems, the Propagation service could also register itself as a client who would make publication, subscription or query requests. In addition, if the enterprise hosts a federation (and thus its own federation services – not shown in the figure) the Propagation Service could register with that federation and use its membership in both federations to move requests between them.

The figure also illustrates that federation can be used to integrate an individual information system with multiple enterprises. In this example, an information service connected to ESB-n also interacts as a federate in Enterprise 1 via its FEIS interface to the Propagation service.

Although these examples present federation as a centralized set of services, it should be noted that the organization of the services implementing federation can be decoupled from the logical view of a single federation. In particular, in this example, the federation consists of all of the enterprises shown and we show the federation services centralized at Enterprise 1 for ease of explanation. Other implementation models are possible for federation services, including hosting the federation on a separate system from all of the enterprises (federates), or distributing the federation services. The former is similar to the federation prototype described in Section 2; we discuss the latter in the next section in the context of federating systems with disparate capabilities.

5. DISTRIBUTING SERVICES IN A FEDERATION

Scalability and performance requirements for federations, as well as the number of federates expected to participate in a particular federation deployment, will motivate design and configuration choices for federation services. For example, a small federation might have centralized federation services, whereas a federation with federates distributed across a large network might need multiple, distributed services. A configuration where every

federate has all federation services would be similar to a peer-to-peer system with a high degree of uniformity among the members.

Whereas a centralized set of services could become a bottleneck as the number of federates increases, a fully distributed (peer-to-peer) model also breaks down as the number of federates increases and synchronization overhead for the services becomes unmanageable. For example, when the Data Discovery Service is distributed, the degree of accuracy needed when discovering information locations must be balanced against available resources. Frequent synchronization of the distributed service instances would be needed for accurate request and data routing decisions, but could have a significant negative impact performance. Less frequent synchronization could result in better performance, but could also reduce the accuracy of routing requests and information. The tradeoffs increase as the number of federates in the federation and the amount of inter-federate communication increases.

In the remainder of this section we discuss some of the choices that can be made in determining the distribution of federation services. We examine the federation of enterprise systems with tactical systems because the different performance characteristics of the two types of systems illustrate a number of the issues that need to be considered in service distribution. We also believe this mix of systems illustrates many of the concerns that would need to be considered in federating Web-based services.

5.1 Characteristics of Tactical Systems

Tactical environments are characterized by two features in particular that distinguish them from enterprise environments. First, computation resources of any type are often limited in the tactical environment; most important are network bandwidth and physical memory, but disk storage, memory and processor capabilities may also be limited. Second, network connectivity or even the nodes themselves may be unreliable. In a tactical environment a node may become unavailable due to transient failure or may fail completely and not return to the federation; it may be unavailable due to a planned outage and may leave permanently or be expected to a return at a later time (e.g., reconnaissance). Transient failures could result in the loss of state information at a node, or a node could return with full knowledge of its state at the time it was disconnected. In comparison, in an enterprise environment, outages are likely to be less frequent and more predictable. In addition, if an enterprise component does fail, we can usually expect it to retain its state.

The characteristics of tactical systems favor designs in which services are distributed, both for reliability against communication or node failure and to allow for computational load to be distributed according to availability of relevant resources.

5.2 Federating Enterprise with Tactical Systems

Federations consisting of both enterprise and tactical networks must support access by enterprise-scale client loads to data originating from a tactical network without overloading the capabilities of components on the tactical network and without inhibiting local access to tactical network resources by tactical network clients. Conversely, the tactical environment should be able to take advantage of the resource richness of the enterprise environment for processing it does not have the resources to perform or that can be performed more efficiently or effectively in the enterprise environment.

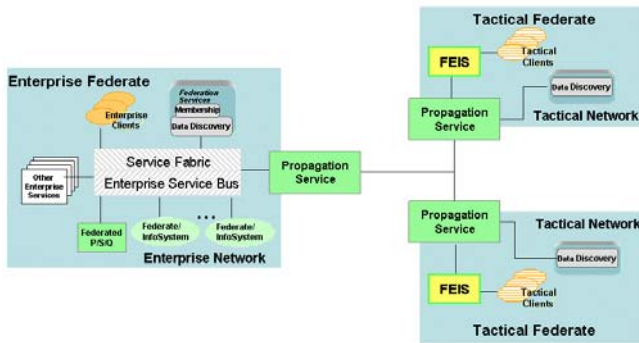


Figure 6. Distributed services combine enterprise and tactical federates in a single federation.

One design pattern federating enterprise and tactical system networks has each network as a federate in the federation. This is illustrated in Figure 6. As described in Section 2, each federate connects to other federates in the federation via Propagation Service interactions. In this design we distribute Data Discovery services so that each federate has a Data Discovery Service instance.

Although synchronization of state information in distributed Data Discovery Service instances can potentially improve the accuracy of request routing, in this environment we do not expect complete synchronization between Data Discovery Service instances and instead accept a “best effort” model in which the federates identified by a local Data Discovery Service may be a subset of the federates that potentially could participate. We believe this trade-off between accuracy and performance is more acceptable in the mixed environment because it resembles the effects of intermittent connectivity of tactical federates. This environment will also require error handling mechanisms for situations when a federate’s Propagation Service attempts to contact a federate that is no longer responding.

One advantage of mixed federations is that an enterprise federate could provide services to tactical federates that they cannot perform for themselves or that can be performed more efficiently in the enterprise environment. For example, an enterprise federate could be configured such that it archives the data published by tactical federates that cannot archive their own data. This could be accomplished, for example, by configuring a subscription client in the enterprise to receive all data produced from a particular federate and republish that data to its local FEIS for archiving. This configuration has the difficulty that the republished data could duplicate those data received by other federates directly from the tactical federate. One solution is for the tactical federate to only propagate publications to an enterprise federate; this would be configured in the tactical federate’s Data Discovery Service instance. In this case, the federation might also maintain relevant configuration information in the tactical federate’s Data Discovery Service instance to route query requests within that federate to the “closest” archival store.

6. SUMMARY AND FUTURE WORK

In this paper we presented a services-based architecture for information federation and examined how federation can contribute to information integration in SOA-based systems. In particular, the services and service interactions needed to specifically support storage, access and dissemination of distributed information can augment those services and interactions provided in a SOA that support service interaction. In addition to incorporating federation as a service for SOA, we

discussed how federation can be used as an integration mechanism for heterogeneous SOA-based enterprise systems and for integrating SOA-based systems with non-SOA systems.

We have developed federation prototypes with software services that address the technical aspects of federated interoperability, including processing of publish, subscribe and query requests and exchanging information across federate boundaries, information protection and security, and configuration and dynamic state management. As future work we plan to develop additional services supporting federation, and to incorporate our federation services within SOA-based systems.

As new services are added to a federation, we will want to compose them in different combinations or in different locations within the federation to give different behaviors for particular actions. For example, a Propagation Service composed with a Security Service and a Translation Service could enforce a particular access control on requests transmitted between a pair of heterogeneous federates. In a SOA environment we need to ensure that we don’t assemble an inefficient or ineffective orchestration of services, e.g., one that introduces information cycling through the federation. Additional work is needed to develop a set of useful interaction patterns and define effective orchestrations of services for the exchange of information and requests.

7. ACKNOWLEDGMENTS

TBD

8. REFERENCES

- [1] Object Management Group, “Common Object Request Broker Architecture (CORBA) Specification, Version 3.1,” Jan. 2008. <http://www.omg.org/spec/CORBA/3.1/>
- [2] Object Management Group, “CORBA Component Model Specification, Version 4.0,” Apr. 2006. <http://www.omg.org/technology/documents/formal/components.htm>
- [3] M. Linderman, B. Siegel, D. Ouellet, J. Brichacek, S. Haines, G. Chase, and J. O’May, “A Reference Model for Information Management to Support Coalition Information Sharing Needs,” *The Tenth International Command and Control Technology Symposium (ICCRTS)*, 2005.
- [4] V. Combs, R. Hillman, M. Muccio, and R. McKeel, “Joint Battlespace Infosphere: Information Management within a C2 Enterprise,” *The Tenth International Command and Control Technology Symposium (ICCRTS)*, 2005.
- [5] self reference, “TBD.”
- [6] “OASIS eXtensible Access Control Markup Language (XACML) Version 2.0,” Feb. 2005. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml
- [7] M. Carey, “SOA What?,” *Computer*, vol. 41, 2008, pp. 92-94.