Enterprise Architecture Special Interest Group (EA-SIG) Mediation Working Group An Analysis of Mediation in the GIG Enterprise Services

Version 1.0

February 20, 2004

Prepared for the

Defense Information Systems Agency
NCES Program Office

By the

Open GIS Consortium, Inc. (OGC)
Enterprise Architecture Special Interest Group

Contributors

Paul Lunceford - FGM

Steve Matney - SRA

Tom Huggins – Polexis

Chuck Heazel – Lockheed Martin

Disclaimer: This document represents the consensus view of the members of the EA-SIG Mediation Working Group. The contents of this document do not necessarily represent the views or policies of the Department of Defense nor of the employers of working group members.

Change Log

Date	Author	Description	Version	Affected Pages
12/02/03	S. Matney	Near-final draft	0.3	
2/11/04	C. Heazel	Added change log Added discussion of mediation as an integrated capability of a service Added discussion of mediation standards Deleted Issues section Populated recommendations Cleaned up graphics Misc. cleanup	1.0	pg. 2 par. 3.2.5 par. 3.3 par. 5 par. 5 par 4.2
2/20/04	EA-SIG	First delivery to DISA	1.0	

Table of Contents

1 Introduction

The sole purpose of an IT enterprise is to support the execution of mission operations. Due to the great variety and criticality of the mission operations supported by the GIG, a single one-size-fits-all IT infrastructure is not appropriate. It is clear, then, that the GIG will consist of a large number of mission focused, independent IT environments. These autonomous environments must still integrate into a single GIG enterprise, a federation of systems.

To achieve interoperability, we need a "set of negotiated agreements for interaction between components that enable those components to work together to perform a task." These agreements, defined through standard interface and data interchange specifications, form the framework around which an interoperable enterprise can be built.

In a large enterprise of autonomous systems, such as the GIG, the definition of a single set of standards that are suitable for everyone is nearly impossible. Each system has its' own unique requirements that developers must address. Individual system environments will be built to the set of standards, data models, and technologies that best address those requirements. To participate in the greater enterprise, there must be a way to bridge the incompatibilities between these individual IT environments. Mediation services provide that capability,

2 Scope

Figure 1 depicts the broad scope of GIG Enterprise Services (GES). As the enterprise services component of the Global Information Grid, GES is the infrastructure on which DoD computer applications (e.g., C2, Combat Support, Medical) rely. GES in turn relies on the GIG transport services such as the Defense Information System Network (DISN) and tactical communications systems. DISN and tactical communications systems consist of transmission systems, distribution/switching systems, Video Teleconferencing (VTC) and packet and other support infrastructures.

While GES relies upon the GIG transport services for the exchange between the Core Enterprise Services (CESs) and the Community of Interest (CoI) capabilities, transport is not an inherent component of GES. There are nine CESs – Application, User Assistance, Storage, Messaging, IA/Security, Discovery, Collaboration, Mediation, and Enterprise Service Management (ESM) services. These core services will provide a common IT infrastructure to provide reliable, secure and efficient information delivery to decision makers and the war-fighter.

This document focuses on the goals, objectives, capabilities and recommendation for the Mediation Core Enterprise Service.

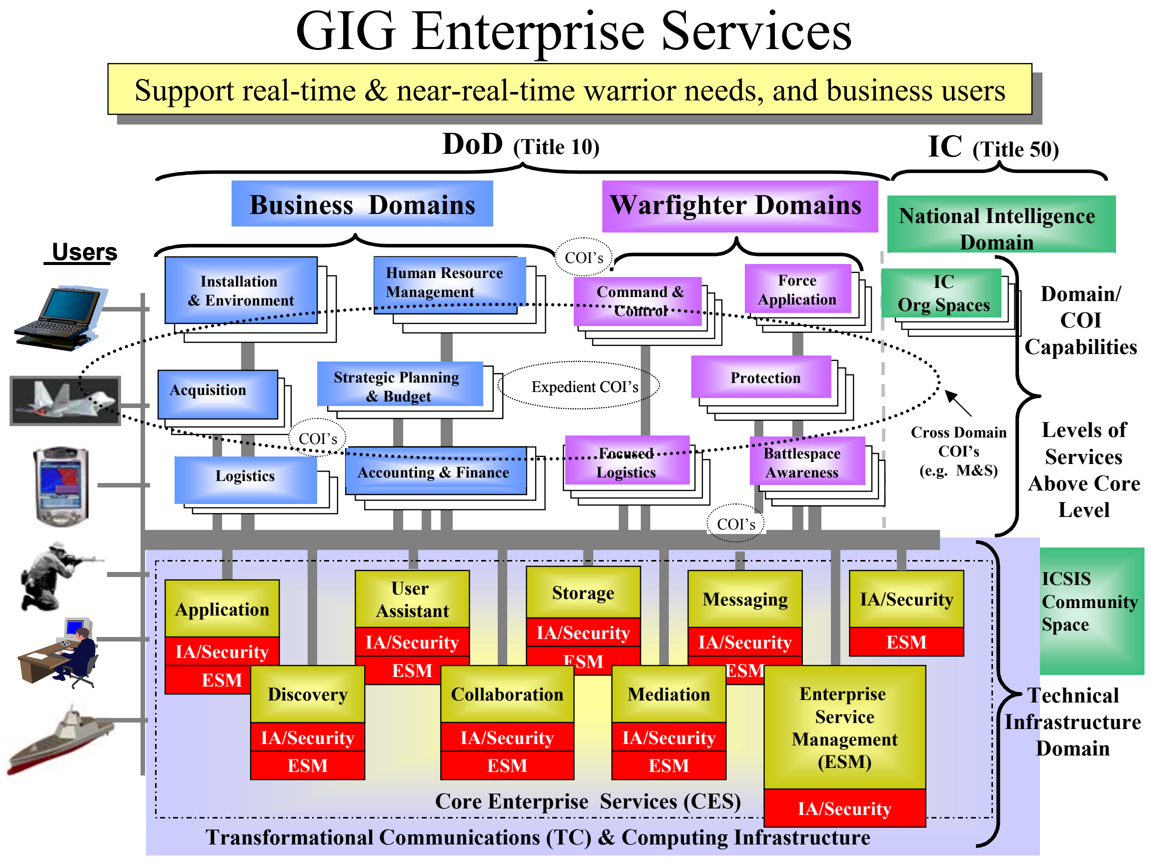


Figure 1 – GIG Enterprise Services

3 Technical Analysis

3.1 Mediation Concepts

3.1.1 Adaptation

Today, software developers are often confronted with the issue of integrating their systems with other systems. This is especially clear when developing enterprise systems or integrating with business partners. The fact that an activity has happened on my system requires me to pass information to my buddy's system. Consider the commercial procurement world. I want to order a box of pens. I enter into my ERP system the order. That system now needs to pass the invoice information on to the organization from which I placed my order. As such, my system may be sending out a comma separated file. Great, except the supplier's system only receives EDI documents. I have a problem. The invoice requires "adaptation" to the receiving system. When an invoking application can not communicate directly with an outside service, an adaptor must be employed. Adaptors provide service mediation when systems need to communicate point to point.

It is not always the information packet that needs to be adapted, but sometimes the transport protocol by which that information is received. Assume the caller is utilizing an ftp client tool to transfer information, yet the destination point for that file only has an sftp daemon. Different encryption schemes or handshaking may need to be done between the client and server processes. The ftp client may use an ftp to sftp mediator to handle the transport. This mediator interacts with the sftp server on behalf of the ftp client handling all the sftp protocol demands, and forwards the exchange status back to the ftp client but utilizing the ftp handshaking techniques.

Adaptors may provide enhancements to the service it performs. If the destination service failed, it may notify the calling service of the last time the mediator successfully integrated with that service. Another example might be that a person in the United States places an order for an item in Japan. Not only does the mediator translate dollars to yen, but it may also provide information to the service it is invoking specifying what the original payment was in dollars, what the exchange rate was that was used, and the date and time of the conversion to validate the exchange rate.

3.1.2 Orchestration

What happens when a service request triggers a whole chain of events? The USS Battleship has just reported its new position. That information needs to be disseminated to a number of receivers that are tracking that ship's location. If the ship is close to port, the port needs to be alerted to be ready to receive that ship. If the ship is coming into dangerous waters, the ship may need to be alerted of its predicament. All these actions require "orchestration." Mediation services are employed to provide these hook points between multiple service operations. The chain of events is outlined within the mediator to determine how these services interact and under what condition certain services are invoked. This depicts a smooth orchestration of services. This orchestration is defined as a workflow because it defines how information or instructions received will flow between the separate services that it needs to invoke, and also defines the set of rules by which a service is invoked.

Workflow means different things to different people, yet it is truly a generic term. Workflow orchestration can be used to determine how an electronic transaction is processed with very immediate results, or it may be used to define how a business process is performed where the process lifecycle is measured in days. Workflows can be utilized for a number of functional needs from distributed transaction management, to business practice management, to error handling. Therefore, there are many different devices that can be used to provide service orchestration mediation. Web Services orchestration is a good example of using multiple generic web services to achieve a specific service request (getting weather conditions in Fahrenheit with barometric pressure may require a request to a temperature service followed by a request to a Celsius to Fahrenheit converter service followed by a request to a barometric pressure service). Business Process Managers (BPM) also provide workflow orchestration mediation by ensuring that information is routed throughout the organization's resources (computers and people) to accurately achieve the business need in the most optimal fashion.

Since elements of the workflow need to interact with numerous other parties, other mediation services may need to be utilized by the workflow mediator. Suppose we are using a BPM as a mediator service and we need to interact with different systems. If we are dealing with the tracking location of a ship, different receivers may want different formats. The BPM may need to employ an adaptor mediator to get it to the different services. Likewise, it may need to use information mediation to get information to assist it with its routing decisions.

There are also different workflow mediation services that can be provided. These techniques are predefined, templated, and ad-hoc. All deal with how the choreography is defined. A predefined workflow has the choreography all ready set and is immediately ready to use. Templated workflows have the flow specified, but not the specifics of the destination hooks. In other words people may provide a customization of what the end-points are that the flow hooks to at runtime, or may simply download the choreography to be customized and used locally. Ad-Hoc workflow services are those where the mediation service is available, but the choreography and destination points need to be configured or loaded before it is usable.

3.1.3 Transformation

When an application requests information, it is quite possible that the source by which it requests information is not providing it in the fashion that the requestor desires. That information may need to undergo a "transformation." This is very similar to the adaptation provided by the service mediator. The difference is that while the adaptor conforms the initiator's information packet to the receiver, the transformation has information requested in a specific format, it receives the information and transforms it before passing it back to the initiator. I may want to do an ODBC query to get a relational view of forecasted temperatures in the DC Area for the next 3 days. I may use an information transformation mediator to interact with a web service that provides 7 day forecasts. The mediator strips off the last 4 days and repackages the XML for the first three day predictions in a relational, tabular data package.

In the above example, the information transformer mediator provided an XML to ODBC translation service, thereby repurposing the web service by extending its usability to SQL clients. It may even augment that service by providing the capability of translating from Celsius to Fahrenheit for me.

3.1.4 Aggregation

One of the biggest needs for data mediation is where the requestor does not know who the keeper of the data is, but desires a central point of interaction when requesting information to be served to it. In enterprise architectures, there may be multiple points for information management, yet the users do not wish to spend time navigating the cyberspace for the best (or authoritative) source of the information. Content Integrators are employed to provide the single point of interaction, mediating to the separate information repositories. ETL tools can also be used to build data marts or data warehouses for centralized locations of disparately managed information. Other techniques such as OLAP or ROLAP can be used to summarize information and store that data or information cubes centrally for utilization. Often, a combination of techniques can be used to provide information mediation services.

When dealing with information that is dispersed, consideration must be given for who the authoritative provider is or to the stagnant nature of the data. If information is rather stagnant (what is the name and traditional operational strength of unit X), it might be highly desirable to have the integrator access the fastest available source even if it is warehoused data. However, if the information is very fluid, you may want to go to the “authoritative” source (what is the current location of unit X). Often when providing information mediation services for a generic community, the aggregation of information from multiple sources on the fly is desirable. I may want to provide a data construct that dynamically pulls information from my local repository for getting a units name and their traditional strength, yet has to go to the central army repository to get its current strength, and may need to go to my tracking system to get its current location. Therefore, one single construct can serve the requests mentioned above without the consumer needing to worry about the optimum location for information retrieval. Likewise, a third question could be formed that requests information from all three sources at once, integrating the information together in one comprehensive unit object for an application’s consumption.

Another scenario is one in which there may be no specific single authoritative source. In the case of intelligence and C2 systems, it may be quite clouded as to who authoritatively knows the answer based upon the intended purposes of the underlying information sources. An NSA or CIA system may specify an enemy unit to be in a specific location based upon confirmed sources, yet a C2 system may say that that unit is in another location based upon recent supposition. The C2 system is used to insure that flight plans are developed to minimize risk in the flight path. The intelligence system is used to track last “known” location. So, where is that unit? The consumer may want only confirmed data, or recent supposition at the AOR, or may want tracking by ALL sources with a tag as to who is telling me what. In this case, rather than building one concrete object from multiple providers, I want a separate object from every information system that knows about this data construct.

One final information aggregation technique uses choreography much like the service workflow mediator. In this scenario, multiple sources are used to provide data cleansing or to “focus the picture.” A logistics officer may want to know what the current state is of his/her order. A query is placed to the data aggregation mediator. That mediator queries the local fort system and discovers that the order was sent to the National Inventory Control Point. Querying the Control Point’s repository, it was discovered that a material release order was sent to depot A. Querying the depot, it was found that a shipment was sent using Acme Trucking. Acme Trucking’s system specifies that the order was received and signed for by Sgt Smith back at the fort. By following a chorography to construct the information object, the data was cleansed by each system in turn and an accurate picture was returned such that the logistics officer can now go to fort receiving and pick up the order.

Finally it should be noted that much like the service workflow mediator could employ other mediators, the data aggregation mediator can use data transformation mediators. In the case of the unit object constructed from multiple sources, the location information may be recorded on an X Y plane. A translation may need to be done to transfer that orientation to a latitude longitude coordinate before returning the information to the requesting application.

3.2 Industry Concepts

3.2.1 Enterprise Application Integration (EAI)

Adaptation, Transformation

One of the early problems with eCommerce implementations was that businesses wanted their internal applications to integrate seamlessly with their suppliers, customers and trading partner's applications. As such, they wanted only specific access to those applications. For example, a supply chain management system would send purchase requests to a specific supplier's procurement system when inventory levels ran low without exposing their internal systems or databases. As different organizations utilized different applications with application specific interface calls, there was no common ground by which these systems could integrate. Integration points were difficult to code, non-reusable, and expensive to maintain.

Enterprise Application Integrators took this challenge to provide specific hooks between applications. They would offer translations between several of the predominant Enterprise Resource Planning (ERP) systems. For example, if a buying organization is running an SAP supply chain management system and using CommerceOne as their procurement system, the EAI software might supply translations between the SAP BAPI calls and Electronic Data Interchange (EDI) formats accepted by CommerceOne. Since BAPI and EDI are protocols and do not specifically outline the contents of the package, the EAI package might identify the translations. The EAI software might even allow the developer to map the contents of the BAPI package to the requested EDI format specified by the supplier, or derive new fields of information using a graphical editor. Current EAI solutions also assist organizations in developing web services interfaces to their systems, generating Web Service Definition Language (WSDL) documents, and publishing via Universal Description, Discovery, and Integration (UDDI).

3.2.2 Extract/Transform/Load (ETL)

Transformation, Aggregation

In the 1980's, organizations started to move away from mainframe systems and centralized computing to decentralized applications running on personal computers (PC) and PC servers. This decentralization led to the development of information pockets across the organizations systems. As the systems became decentralized, the organization's valuable information resources also became decentralized. Applications could no longer efficiently leverage all of an organization's information resources. For example, procurement needed to know what vendor products the organization was utilizing in inventory. Therefore, certain information that was gathered and managed by one set of applications needed to be placed into another set of applications. IT organizations then needed to periodically dump this information from its managed location, and discover ways of uploading it into other systems that needed to utilize that data. This led to the creation of what is commonly referred to as data warehouses and data marts. A data warehouse provides centralized access to an organization's vast data while a data mart provides access to smaller subsets of relocated information by secondary applications. Gathering and managing these data warehouses and data marts became very burdensome and difficult to manage. In many cases, the information needed to be transformed in structure, model and/or management system.

This need led several non-database vendors to come out with offerings that allowed IT shops to develop snippets of code to automate certain management functions. These functions included:

- Specialized database extractions based on changes in values or other value based criteria
- Transformation of the information in content and structure
- Migration of the information to the target system
- Rapid loading utilizing characteristics of the target database engine to construct the data warehouse or mart

Also, vendors provided business rule triggering of these actions by allowing scheduling at non-peak times, outside invocation, or by what is referred to as "log sniffers". A "log sniffer" performs monitoring of a databases log file to identify when changes to data have occurred and when data warehouses or data marts need to be refreshed. These functions could either augment or replace the data in the target store based upon some sort of business logic. These offerings became known as Extract/Transform/Load or ETL tools.

By providing ETL from multiple source systems to a centralized target, data from different organizational departments became aggregated into large data warehouses. ETL tools were used to extract detailed information and construct summarized or new information stored in unique structures such as data cubes. This data could then fuel online analytical processing tools (OLAP) which enabled organizations to base decisions on statistics gleaned from the historical information.

Historically, command and control systems have utilized data warehouses and marts. Commonly, intelligence databases produced at DIA in the form of IDB and later MIDB have been warehoused in Targeteers and other systems to assist mission planning. This information has needed to be refreshed as new versions have been published. Also, standards boards have maintained lookup tables for status information and country codes. These tables have needed to be disseminated as they have changed. Management of this information flow has been the job of ETL tools. It should be noted that though MIDB has been historically warehoused by other agencies, it is not necessarily the recommendation for all cases. As we will discuss next, EII provides another means of information dissemination.

3.2.3 Enterprise Information Integration (EII)

Transformation, Aggregation

Data warehouses and marts provided huge successes for organizations by enabling application access to data managed outside its domain. However, it also introduced new problems. The management of the warehouse or mart by log sniffing was not always possible based on the storage management software or political environment. Many governing agencies would simply not allow additional software to be installed on their systems because of the impact on either the source or target system. This is especially the case when dealing with high volume online transactional processing (OLTP) systems. As such, the data in the warehouse or mart became stale. It was only as accurate as the last time the source system was polled. Therefore applications utilizing the data might present a false picture based on out of date information. This inaccurate data would then facilitate poor decision-making. Rather than looking at warehoused information, users required access to the information from the authoritative source instead of an out of date copy. Many times, this information came from multiple sources.

Database vendors tried to solve this problem by creating federated databases. They basically allowed the creation of table constructs that referenced database tables in another implementation of their own product. This would force customers to centralize all of their data management needs on the chosen vendor's proprietary product. However, it became apparent that this did not necessarily solve the problem due to the investment cost and time needed for the transition. Also, as is the case within government entities, centralizing database platforms at all computational centers to one vendor was not even feasible, much less desired. Therefore, these vendors needed to stand up gateways to interface their proprietary protocols to their competitors' protocols. An example would be Oracle's Oracle Gateway for Sybase to translate SqlNet calls to OpenClient. Finally, to develop aggregated information sets, they required the development of SQL Views and stored procedures. This solved the need of accessing information in a timely fashion. However it did not allow the integration of non-relational information. Another problem was that the integration by procedural code was unwieldy and difficult to manage.

Enterprise Information Integration arose to solve these problems. Vendors created environments by which IT shops could develop centralized logical data models of how the enterprise's information related to the overall business. Then they provided the capability of mapping the physical information sources to the portions of the model that they fulfilled. They also developed strategies to join or create unions of information sets across physical domains where more than one source mapped to the central model. Finally, they developed means of drawing information from non-relational sets into their model. Examples of non-relational sets include structured and unstructured text, and data from web services. Finally, they provided standards interfaces that allowed all organizational information to be brokered through their engine to enterprise applications, the enterprise portal, business processing engine and/or the enterprise application integration engine.

As these solutions first evolved, they commonly were thought of as offerings in the EAI space. This mischaracterizes the functions these products perform and clouds the purpose of EAI in general. Recent market analysis has extracted EII from this area and created a new category. The key differentiator between EAI and EII is that one provides access to the application layer or business logic and the other provides access to the information layer. This is important to note because EAI and EII provide completely different capabilities. The question of which to use involves whether the requirement is to invoke an application and consume its output or to leverage organizational knowledge assets. They also are commonly found in different market spheres. EAI enables organizations' ERP systems to interact in an eCommerce space, while EII is an infrastructure approach to allow enterprise applications full access to the organization's knowledge base. EII reduces the risk of conflicting information commonly brought about by redundant storage of data.

3.2.4 Business Process Management

Orchestration

With the infrastructure in place to unify organizational information and provide application-to-application interchange, there was one last business layer that required automation. This business layer required the coordination of automated and manual tasks to control business practices. Business automation had always been governed by business policy. Organizations needed to put in rigid constraints to insure practice reflected policy. Too often action items fell between the cracks. Competitive businesses needed to perform efficiently with zero mistakes. This led to the introduction of business process managers or BPM software. BPM allowed the flowcharting of how an operation was to be performed. Exception handling was mapped into the flow. Automated and manual applications would notify the BPM software when its action was complete. This allowed the BPM to mark the task as complete, determine what the next operational task was that needed to be performed, alert that person or system of the request, and send the necessary data package for processing.

Some EAI solutions have offered some aspects of application coordination. These solutions have been confused with the next breed of orchestration, business process management. The difference is that EAI is about application integration, which provides a common ground for information exchange. As a secondary feature EAI solutions may offer some BPM functionality. However, this functionality is commonly synchronous alerting for short rapid workflows with no informational archive to retain the state of the flow. BPM software is positioned to track those processes that might occur over long periods of time, be able to retain the state of a process flow should the system go down, and provide the ability to monitor the state of the process flow.

Lately, BPM vendors have extended their offerings to include business activity monitors (BAM). BAM software is fairly new in the industry. It provides the capability of monitoring the efficiency of businesses automation systems. BAM works in conjunction with BPM software to monitor the overall process management to determine bottlenecks and constraints within the process. This allows an organization to streamline its business by fine tuning its practices.

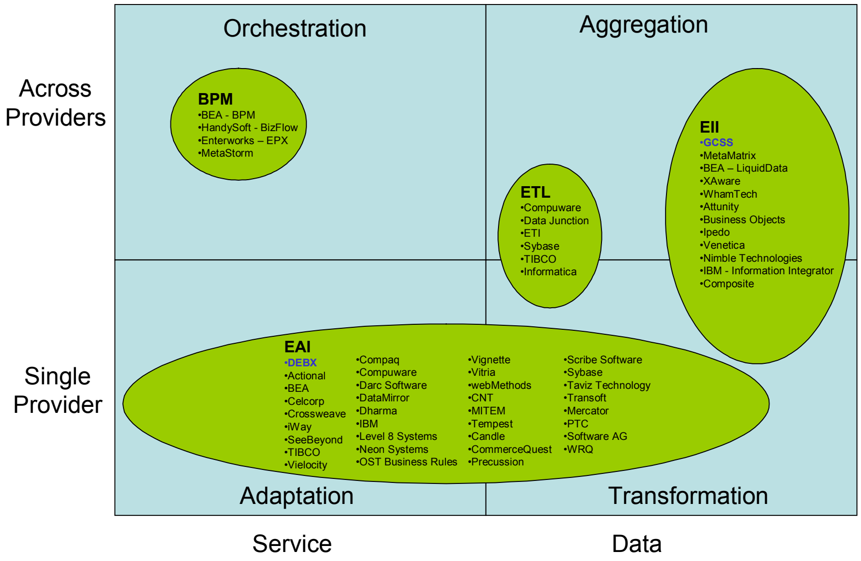


Figure 1 – Mediation Technology

3.2.5 Embedded

While the discussion of Mediation has focused on Mediation services, in many cases mediation capabilities are built into other services. For example, some managed storage services allow the client to specify the format to use in packaging the requested data for delivery. This is an example of an embedded transformation capability. Cascading Web Map Servers such as the FEMA Multi-hazard Mapping site (http://www.hazardmaps.gov) provide aggregation as in integral part of a storage and presentation service. IONIC Software (http://www.ionicosoft.com) fields a suite of products that through a single service supports both web and J2EE interfaces (adaptation). Many other examples exist in the market.

The mediation industry focuses on building and selling mediation products. For GES, these stand-alone products may not be the best solution. The embedded mediation concept positions mediation as a capability to be supplied, not just as a product to be purchased. It leads to an approach of first defining explicit requirements for mediation, then determining if they are general enough to justify a stand-alone service or if they should be bundled within another service.

3.3 Mediation Standards

There are a number of organizations promoting standards for mediation. Few if any of them can be considered ready for operational use. Activities worth monitoring are:

- Business Process Execution Language (BPEL) at OASIS
- W3C Semantic Web (RDF, OWL, etc.)
- Federal Geographic Data Committee (FGDC) Framework modeling and data effort and the supporting efforts to develop data servers that incorporate transformation.

4 Mediation Use Case(s)

4.1 Adaptation Use Cases

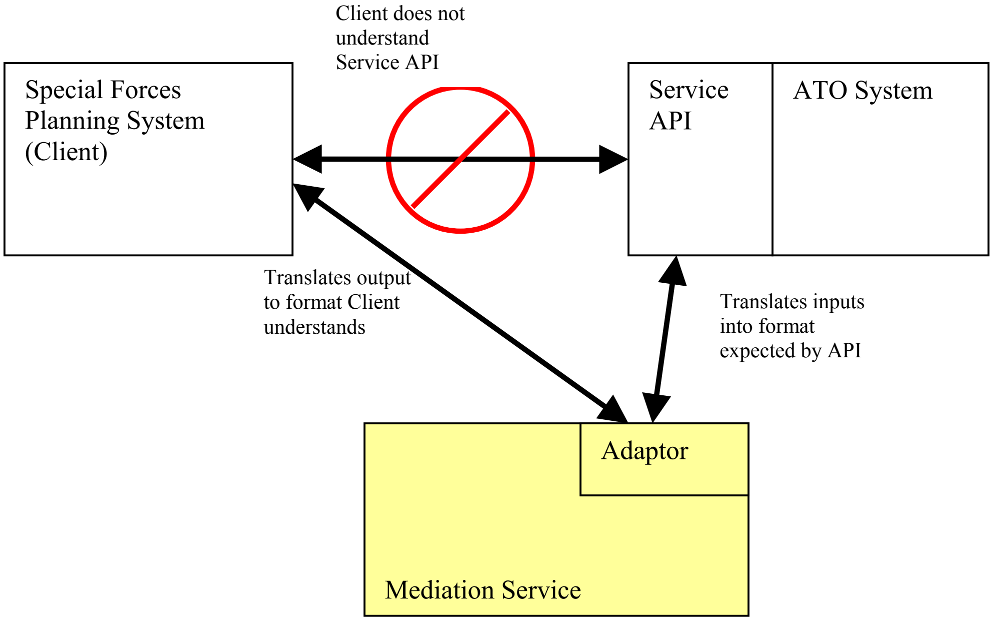
4.1.1 Use Case: Parameter translation

Description:

Parameter translation is required when a service or application wants to initiate an API call to a service whose inputs or outputs are incompatible with the calling service or application. For purposes of this use case, we'll define the calling service or application as the Client. The Mediation Service is responsible for converting inputs from the Client to the format that the Service expects. The Mediation Service is also responsible for converting the output from the Service to a format the Client understands.

Example:

A Special Forces unit is planning a reconnaissance mission and wants to know if any air strikes are planned along the unit's route. However, the Special Forces planning system does not know how to communicate with the system that holds the Air Tasking Order (ATO). The planning system sends a request to the Mediation Service to contact the ATO system and request any ATOs that are near the unit's route. The Mediation Service has information on the web service interface that the ATO system supports and translates the planning systems request into the correct interface format. The ATO system returns the ATOs that are near the unit's route to the Mediation Service, which translates the return into a format the planning system can understand.


Client does not understand Service API
Special Forces Planning System (Client)
Service API
ATO System
Translates output to format Client understands
Translates inputs into format expected by API
Adaptor
Mediation Service

Actors

Client – initiator of call to a mediation service

Service – mediation service that performs some processing on behalf of the client

Pre-Conditions

Client has a service it wants to call but the inputs and outputs for that service are incompatible with the data formats implemented by the client.

Flow of Events

1. Determine conversions necessary to adapt available data for call to service
2. Transform appropriate available data into necessary format using above conversions.

Post-Conditions

- Client receives data in the proper format.

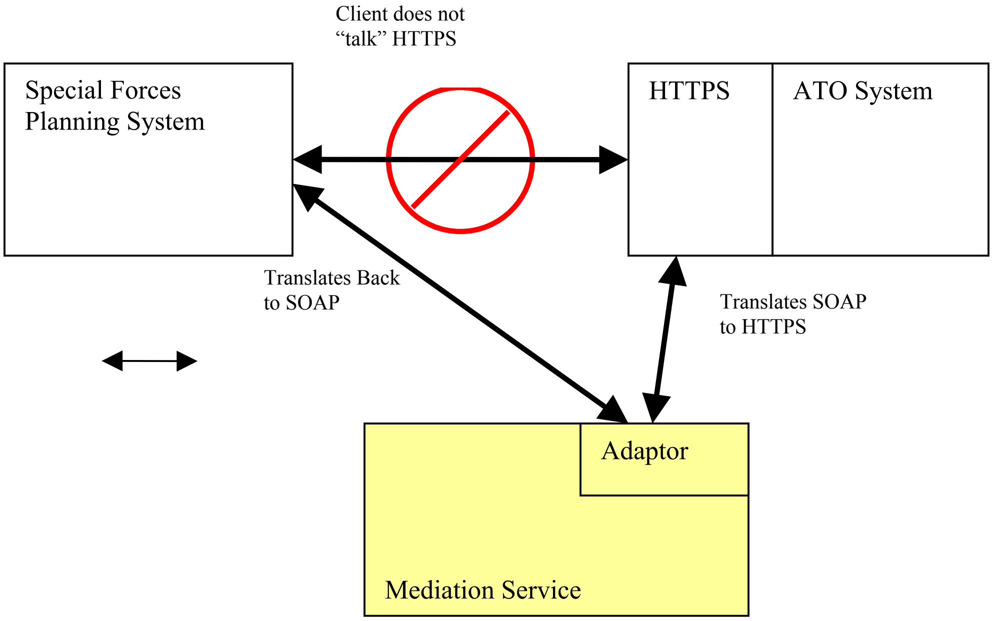
4.1.2 Use Case: Changing protocols

Description:

Changing protocols is required when a service or application wants to initiate an API call to a service that uses a protocol that the calling service or application does not understand. For purposes of this use case, we'll define the calling service or application as the Client. The Mediation Service is responsible for converting the protocol used by the Client into the protocol used by the Service and back again.

Example:

A Special Forces unit is planning a reconnaissance mission and wants to know if any air strikes are planned along the unit's route. However, the Special Forces planning system does not know the protocol that is required to communicate with the system that holds the Air Tasking Order (ATO). The planning system sends a request to the Mediation Service to contact the ATO system and request any ATOs that are near the unit's route. The Mediation Service has information on how to translate the SOAP request from the planning system into the HTTPS protocol that the ATO system supports. The ATO system returns via HTTPS the ATOs that are near the unit's route to the Mediation Service, which translates the return into the SOAP protocol that the planning system can understand.



Actors

Client – initiator of call to a mediation service

Service – mediation service that performs some processing on behalf of the client

Pre-Conditions

Client has a service it wants to call but it cannot because it does not know the protocol necessary to communicate with that service.

Flow of Events

1. Locate mediation service that can bridge the protocol.
2. Ask mediation service to call the desired service on the client's behalf.
3. Proxy service returns results or exceptions to the client.

Post-Conditions

- Client has access to the service.

4.2 Orchestration Use Cases

4.2.1 Use Case: Process Orchestration (non-persistent session)

Description

Client wants to make several calls to the same mediation service as part of a longer process. On each invocation, the service coordinates several calls behind the scenes, possibly determining which to call based on state saved between calls. The mediator is prepared to save state in between invocations to continue the orchestration of the end-to-end goal. Note that the ongoing process could be synchronous or asynchronous with relation to the client.

Example:

A Special Forces planning system is being used to schedule a Special Forces unit's deployment. There are several steps that must be taken by the planning system to develop the schedule. These include such items as checking the unit's readiness, checking unit's equipment status, ordering supplies, and finding and booking transportation. The planning system will make multiple calls to the Mediation Service, which will properly format each request and call the appropriate services.

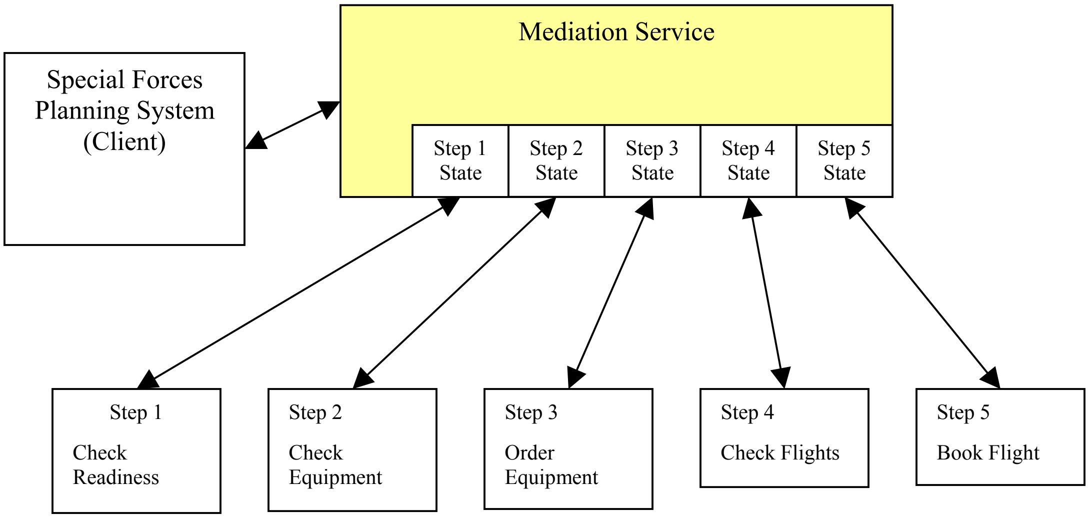
Step 1: Properly format the request and call the unit readiness system to check readiness status. Save the readiness status.

Step 2: If the unit's readiness is at a certain level, call a service that provides the unit's equipment status report. Return the readiness data and equipment status to the planning system.

Step 3: The planning system makes a call to the Mediation Service to call a service that will order additional equipment for the unit. The Mediation Service properly formats the request in a form the ordering service understands, makes the request, and returns the results.

Step 4: The planning system makes another call to the Mediation Service, asking it to call a service that will return available flights for the deployment date. The call is made and the results are returned.

Step 5: The planning system makes another call to the Mediation Service, asking it to call a service that will book the correct flight.



Actors

Client – initiator of call to a mediation service

Service – mediation service that performs some processing on behalf of the client

Pre-Conditions

Client has located desired service.

Flow of Events

1. Client calls service that performs orchestration.
2. Service calls other services behind the scenes, “orchestrating” part of a (potentially) long-term process on behalf of the client.
3. Service stores data about the state of the conversation with the client.
4. Service returns results or exceptions to the client.

This process may be repeated over several invocations of the service.

Post-Conditions

- Client has received results or exception indication from that service.
- The current state of an overall process remembered the next time they call the same service.

4.2.2 Use Case: Process Orchestration (persistent session)

Description

Client wants to call a service that orchestrates a potentially complex process behind the scenes. Each call to the service is independent of any other calls; no state is saved between calls. The orchestrated process calls one or more other services, coordinating their execution, managing intermediate results and handling exceptions. Note that the ongoing process could be synchronous or asynchronous with relation to the client. In this case, if the process is asynchronous, the user will receive acknowledgement of the process being executed, but the service may not have completed that process.

Example:

A Special Forces unit is planning a reconnaissance mission and wants to know if any air strikes are planned along the unit’s route. However, there are several steps that have to take place for the planning system to retrieve the data and the planning system does not know how to complete the steps. The planning system simply wants to make a request and get the information back that it needs. The planning system calls the Mediation Service with the request for any air strikes that are near the unit’s route. The Mediation Service then completes a sequence of steps to get the data.

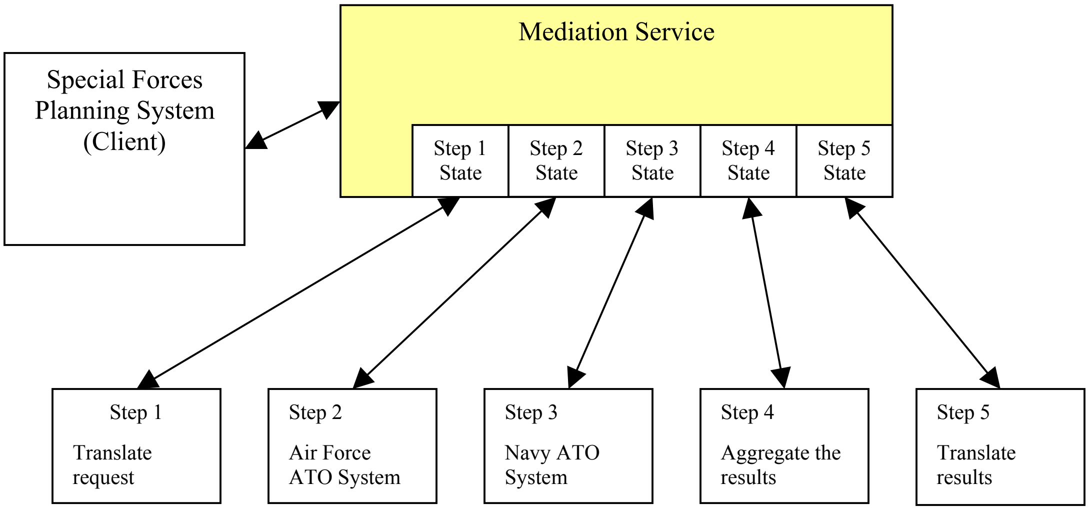
Step 1: Call a translation server to properly format the request.

Step 2: Take the properly formatted request and send it to the Air Forces ATO system and save the result.

Step 3: Take the properly formatted request and send it to the Navy’s ATO system and save the result.

Step 4: Take the results and aggregate them together and save the results

Step 5: Translate the results into a format that the Special Forces Planning System understands and return the formatted results



Actors

Client – initiator of call to a mediation service

Service – mediation service that performs some processing on behalf of the client

Pre-Conditions

Client has located desired service.

Flow of Events

1. Client calls service that performs orchestration.
2. Service calls other services behind the scenes, "orchestrating" them as a single unit of work on behalf of the client.
3. Service returns results or exceptions to the client.

Post-Conditions

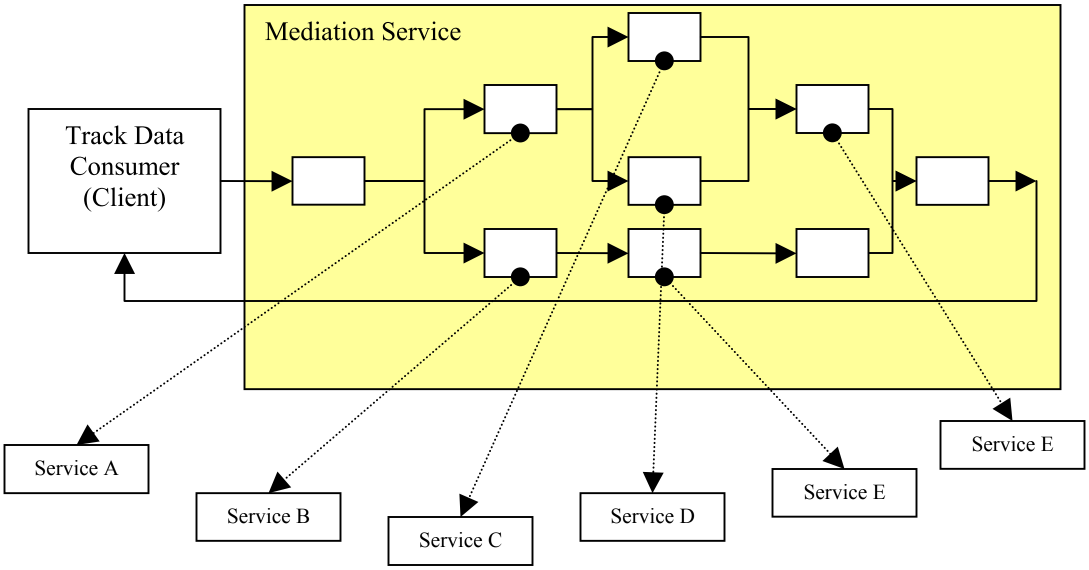
- Client has received results or exception indication from that service.

4.2.3 Use Case: Dynamic Coordination

Description

Client wants to call a service that orchestrates a potentially complex process behind the scenes. Each call to the service is independent of any other calls; no state is saved between calls. The orchestrated process calls one or more other services, coordinating their execution, managing intermediate results and handling exceptions. In some cases, the orchestration may be gathering data from several sources to return as a single result.

Determination of which other services to call is based on data retrieved by called services or retrieved data.



Actors

Client – initiator of call to a mediation service

Service – mediation service that performs some processing on behalf of the client

Pre-Conditions

Client has located the desired service.

Flow of Events

1. Client calls service.
2. Service calls other services behind the scenes. Determination of some services is dependent upon data retrieved.
3. Desired service collects results and returns result or exception to the client.

Post-Conditions

Client has received results or an exception from the service.

4.3 Transformation Use Cases

4.3.1 Use Case: Formatting/Algorithmic/Lookup/Augmentation/Repurposing

Description

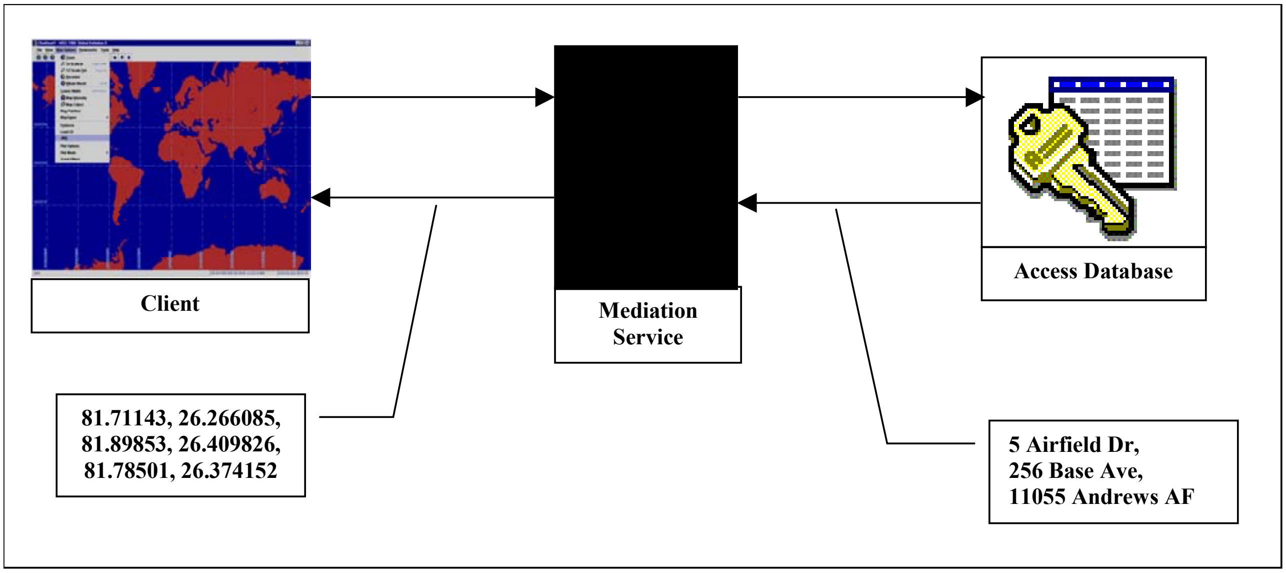
Client wants to call a service that transforms data using one of the following schemes:

- **Formatting** - converting values based on rules such as locality (date/number format)
- **Algorithmic** - logic applied to change values, such as unit conversions (feet-to-meters, etc)

- **Lookup** - convert parts of data based on data retrieved from another source
- **Data Augmentation** - additional information added to output content, such as markers to indicate whether data is authoritative or not, which data source it came from, etc.
- **Repurposing** - Changing the structure of the output to a different target structure

Example

From CES Discovery, a client finds that there is a service in the enterprise that will return address information from a Microsoft Access database for airfields in a remote region. The client needs those addresses converted to geographic coordinates in order to plot on the situational awareness display. The client calls the Mediation Service who determines this transformation can be done within the Mediation Service or from a service on the LAN. The Mediation Service invokes the target service and transforms the addresses into geographic coordinates. If the Mediation Service cannot do the transformation locally, it will use CES mediation services to attempt to perform the transformation.



Actors

Client – initiator of call to Mediation Service

Service – target service

Transformation Service – mediation service that calls the target service and transforms the resulting data

Pre-Conditions

Client has located the desired service.

Flow of Events

1. Client calls transformation service.
2. Transformation service calls the desired service.
3. Desired service returns a result or exception to the transformation service.
4. Transformation service modifies result according to one of the above schemes.
5. Transformation service returns results or exception to the client.

Post-Conditions

Client has received results or an exception from the transformation service.

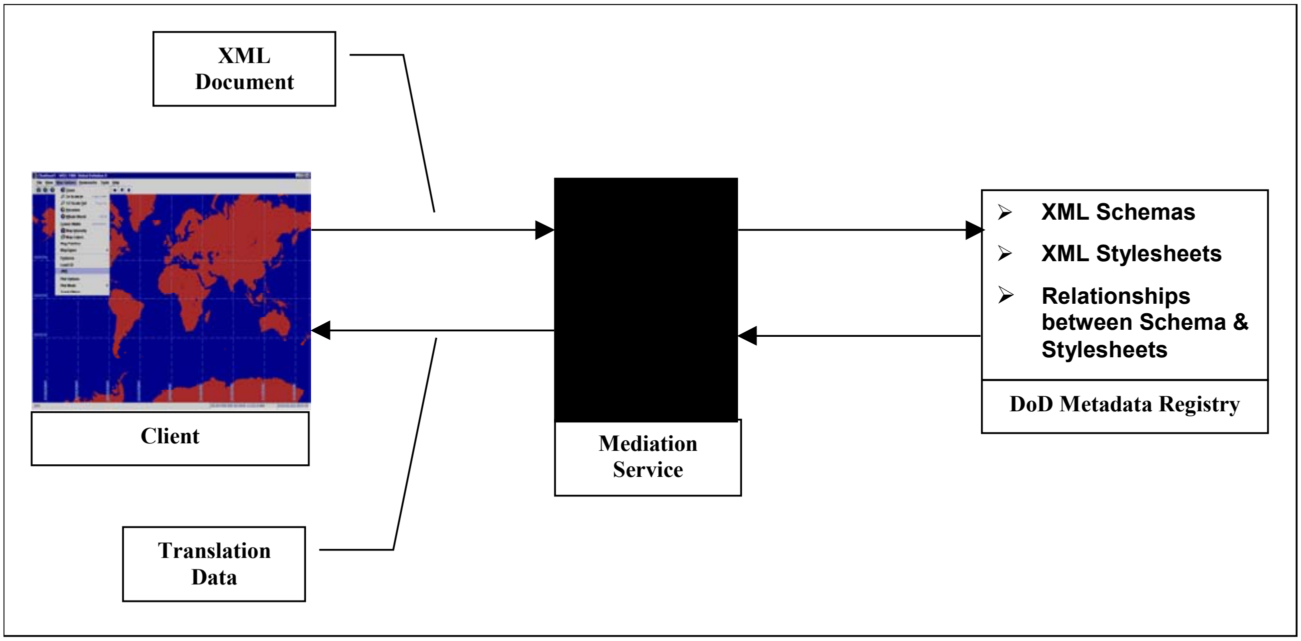
4.3.2 Use Case: Specification-based Transformation

Description

Client wants to locate and use an appropriate specification to transform data into another format or structure *within the client*.

Example

A client would like to find alternate views for an XML document. The client calls the Mediation Service with information like the schema name related to the XML document. The Mediation Service retrieves data from DoD Metadata Registry about the translation provided. The client would then translate the document*. The Mediation Service may hold these translations within it's own registry or within some cache in which case a call out to the DoD Metadata Registry isn't needed.*



Actors

Client – the actor who wants to translate data using a specification

Service – used to locate a specification

Specification – a specification (such as an XSL stylesheet) that can be used to perform a translation in the client

Pre-Conditions

Client has data that it wants to modify and believes there may be an existing specification for performing that modification.

Flow of Events

1. Client calls the service to locate the appropriate specification (such as an XSL style sheet).
2. Mediation returns the specification or an exception to the client.
3. Client performs the transformation using the specification. (For example, the client could run an XSLT engine, such as Xalan, to transform a piece of XML using an XSL stylesheet). *The client could call the Mediation Service to perform the transformation.*

Post-Conditions

Client has transformed data or an exception.

4.4 Aggregation Use Cases

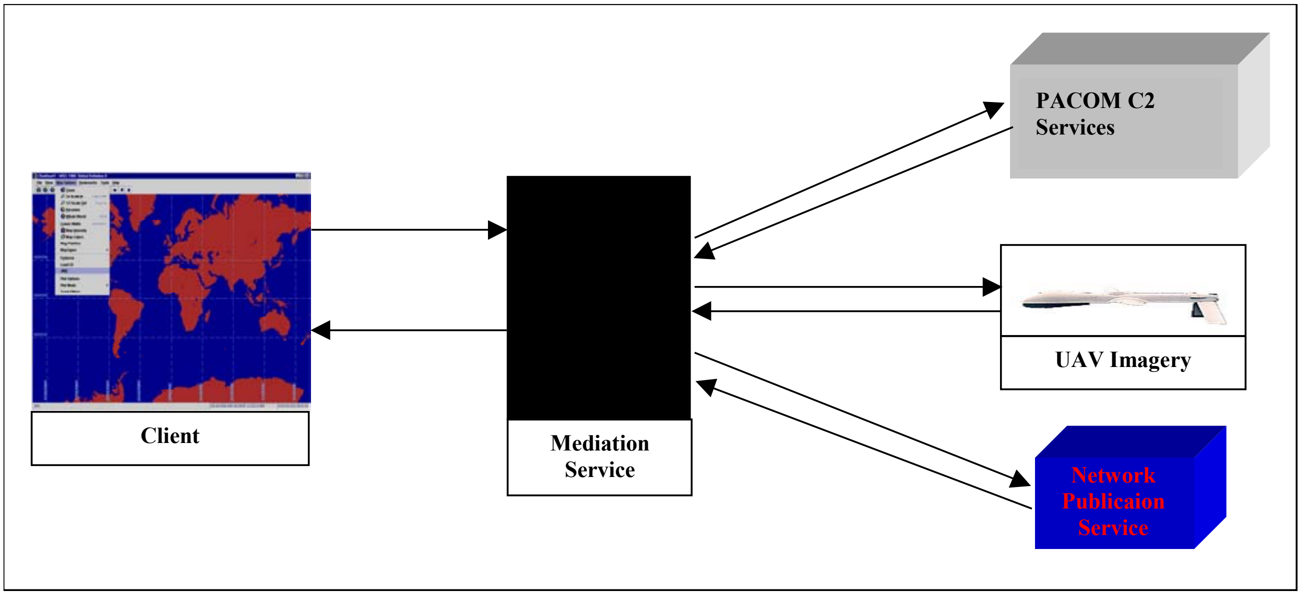
4.4.1 Use Case: Information Combination

Description

The client desires to combine data from multiple sources. A mediation service retrieves the data from one or more sources and combines that data into a single result.

Example

A client would like an integrated data picture for a specific red track. His view includes attaining positional and attribute data from a Track Management Component, retrieving images from a predator feed, and extracting documents from a Network Publication Service database that associates external documents to Tracks. The Mediation Service will call the appropriate services, combine the data and return in format useful for the client. *The Mediation Service may use a set of C2 specific business rules to determine how to collect this information based on various factors like performance, local availability, or authoritative source.*



Actors

Client – initiator of call to the Mediation Service

Service –service that combines data for use by the client

Pre-Conditions

Client has located the desired services.

Flow of Events

1. Client calls the service.
2. The service gathers data from one or more data sources/services.
3. The service combines the data into a single result, making the data appear to have come from a single source.

4. Service returns the result or an exception to the client.

Post-Conditions

Client has resulting data or an exception.

4.4.2 Use Case: Information Coalescence

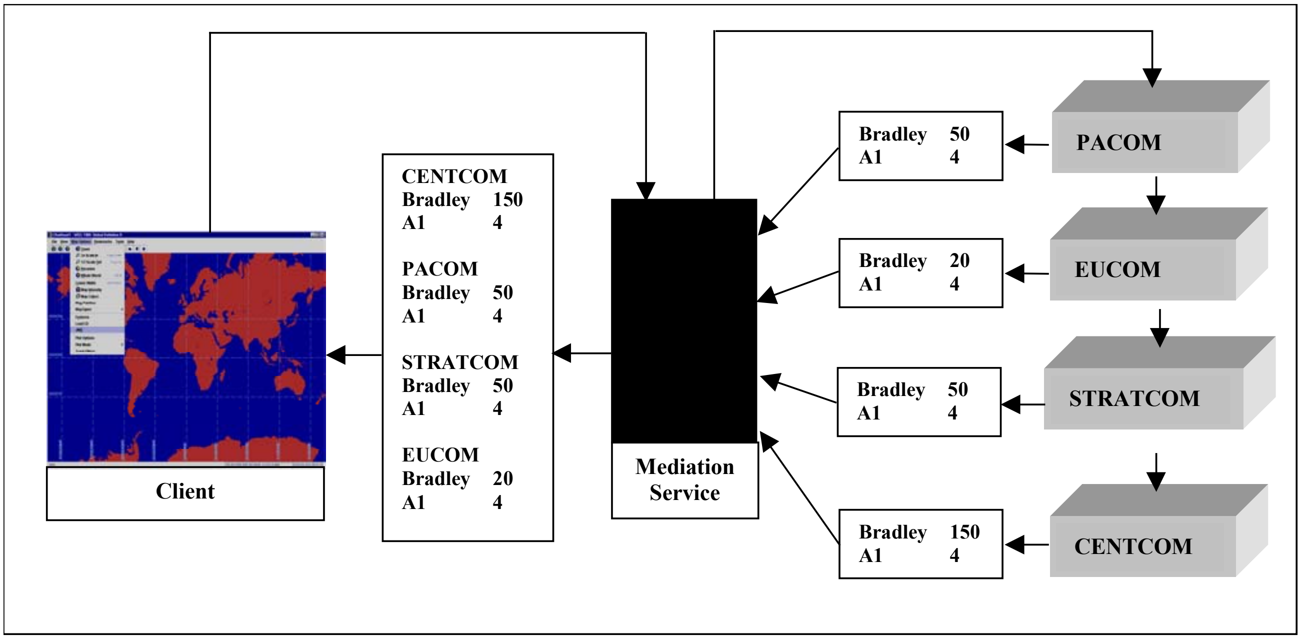
Description

The client desires to retrieve data from multiple sources. A mediation service retrieves the data from one or more sources and groups that data into a single result.

In this example, there is either no reasonable means to join the data to appear as a single data source, or there are differences in policy, such as differences in sources being authoritative or not.

Examples

Such a grouping will not necessarily be a common picture. It will be generated on demand by the user rather than by command of a single authority. In such a data driven environment, a client might like to see a set of logistical data that can be produced by multiple communities. The Mediation Service would call the appropriate target services and group the logistics sets based on an authoritative order into a single result. *The Mediation Service may use a set of C2 specific business rules to determine how to collect this information based on various factors like performance, local availability, or authoritative source.*



Actors

Client – initiator of call to the Mediation Service

Service – mediation service that groups related data for use by the client

Pre-Conditions

Client has located the desired service.

Flow of Events

1. Client calls the service.

2. The service gathers data from one or more data sources/services.
3. The service groups the data into a single result. All data from all sources is returned.
4. Service returns the result or an exception to the client.

Post-Conditions

Client has resulting data or an exception.

5 Recommendations

5.1 Immediate – Today

Mediation is one of the most essential capabilities required to enable the GES. However, our understanding of what we need to mediate is incomplete. Mediation requirements will evolve as more information and operational communities become actively involved in the GES. The task of integrating these new communities will include the identification of disconnects between environments and identifying the mediation capabilities needed to fix them. The following recommendations identify work that can be done in the short term to reduce the long term mediation efforts.

5.1.1 Transformation:

In preparation for the deployment of transformation services, an understanding of the current information communities would be in order. An excellent model of the as-is state can be found in the DoD XML Registry. The registry Namespaces are based on the existing DoD organizations that have authority over an information community. The content of each namespace captures a significant percentage of the information constructs used within that community. NCES should leverage this opportunity to plan for the transformation services that will be needed in the future. In particular, the NCES Program should:

- Work with the Namespace Managers Forum to identify information constructs that are common across most DoD information communities. These constructs should be promoted as common constructs.
- Work with the Namespace Managers to identify information constructs that cross information communities in the course of operations. This is the subset of information that is a candidate for transformation.
- Work with the Namespace Managers to identify which members of that subset cannot be used by the receiving organization in its' native format. Identify the transformations needed to make the conversion.
- Establish the previous activities into the normal operations of the Namespace Managers Forum. Use the forum to manage the definition of what transformations need to take place and the nature of those transformations.

5.1.2 Adaptation:

For the short term, the NCES Program should focus their adaptation efforts on enabling the Core Enterprise Services and integrating them with the existing infrastructure. In particular:

- Establish adapters allowing J2EE based clients to invoke the CES Discovery Services using J2EE discovery interfaces.
- Establish adapters allowing J2EE based clients to invoke other CES services using J2EE interfaces on an as needed basis.
- Establish adapters integrating legacy information security infrastructures (such as Active Directory) with the CES security services.
- Establish a taxonomy of the services types needed to support DoD operations and identify or define standard interfaces for those service types.

- Develop a process for managing the portfolio of services and interfaces. Define the services and interfaces using UML models so that semantically equivalent implementations can be generated as new distributed computing technology becomes available.
- Develop adapters for key legacy systems to integrate them into the greater GES community.

5.1.3 Aggregation:

Aggregation does not appear to be a near term need. Some aggregation capabilities exist in the C4I community already. Additional aggregation capabilities can be expected to appear as an integral part of emerging services. The NCES program office should focus their aggregation efforts on identifying the requirements as they develop and shepherding development efforts toward common implementations.

5.1.4 Orchestration

Orchestration is of little use without a rich portfolio of deployed services to orchestrate. Near term efforts should focus on orchestration pilots and prototypes to discover the capabilities of this technology and to identify possible opportunities to use it.

5.1.5 General

There is a lot of work in the research and standards communities in the area of mediation and semantic interoperability. It is doubtful that any of this is ready for robust operational use. Just the same, this work lays out the framework under which future technologies will be built. The NCES program must monitor and, where possible, participate in these activities. Even more important, the NCES program must be careful that decisions made today will have a viable migration path to the future.

5.2 Short Term – Next 5 to 10 years

Within five years expect to see the semantic web work mature to the point that it is suitable for early implementation. NCES should participate in these efforts both by providing requirements and by sponsoring pilots to exercise and evaluate the technologies. Only after being proven through the pilots should a given mediation technology be integrated into the operational environment.

5.3 Vision – 10+ years

Eventually mediation will be an invisible service. As part of the find-bind process, a negotiation between the two parties will take place to determine what adaptations have to be made to establish an acceptable interconnection. These adaptations will include the application of the appropriate mediation capabilities, controlled through standard ontology and rules documents available from semantic registries.

Appendix A - Mediation Terms

Ad-Hoc COI – an operational COI that forms in response to immediate events. Ad-hoc COIs come into existence to address an issues and disband once that issue have been resolved.

Application Schema – An application schema provides the formal description of the data structure and content required by one or more information communities. --- set of conceptual schema for data required by one or more applications.

COI – Community of Interest.

DCP – Distributed Computing Platform

F**eature** – abstraction of a real world phenomenon or attribute of a system

Federation – an IT configuration where organizations and systems collaborate without a single management framework.

GML – Geographic Markup Language

Information Community - a collection of people (a government agency or group of agencies, a profession, a group of researchers in the same discipline, corporate partners cooperating on a project, etc.) who, at least part of the time, share a common digital geographic information language and common spatial feature definitions.

Interface – named set of operations that characterize the behavior of an entity

Jurisdiction - an administrative entity with a single management authority that can establish standard policies, procedures, and technologies. All systems within a jurisdiction are subject to this management framework.

Metadata – data about data.

OGC – Open GIS Consortium

Ontology – the working model of entities and interactions in some particular domain of knowledge or practices, such as electronic commerce or "the activity of planning." A set of concepts - such as things, events, and relations - that are specified in some way (such as specific natural language) in order to create an agreed-upon vocabulary for exchanging information In artificial intelligence (**AI**), an ontology is, according to Tom Gruber, an AI specialist at Stanford University, "the specification of conceptualizations, used to help programs and humans share knowledge." . One or more taxonomies can be developed for the ontology and taxonomies can be used as part of the ontology model.

Operation – specification of a transformation or query that an object may be called to execute. Also, a virtual enterprise established to achieve some real world goal (e.g., Operation Iraqi Freedom) – see Ad Hoc COI

Operational COI - a collection of individuals, organizations, and systems with similar business and information needs. Operational COIs operate across Jurisdictions and Federations and in fact are the primary reason for their existence. Operational COIs develop their own operating conventions addressing such issues as information models, policies, and practices.

Service – distinct part of functionality that is provided by an entity through interfaces accessible over the GIG network.

Taxonomy – the science of classification according to a pre-determined system, with the resulting catalog used to provide a conceptual framework for discussion, analysis, or information retrieval. In theory, the development of a good taxonomy takes into account the importance of separating elements of a group (taxon) into subgroups (taxa) that are mutually exclusive, unambiguous, and taken together, include all possibilities. In practice, a good taxonomy should be simple, easy to remember, and easy to use. However most real world entities and concepts can be viewed as belonging to multiple taxonomies, based on the operational context in which they are referenced. For example, a main battle tank is both a vehicle and a weapon system. It can also be a shelter, cargo, asset, target, etc. in other operational contexts and thus taxonomies.

Viewpoint – form of abstraction achieved using a selected set of architectural concepts and operational contexts with associated structuring/representation rules, in order to focus on particular concerns within a system development, acquisition, or virtual enterprise context.