Agile Integration of Complex Systems

Wayne O'Brien, Raytheon Company IIS
Raimund Merkert, Raytheon Company NCS

Introduction

Service Oriented Architecture (SOA) is fundamental to realizing DoD's Net-Centric Vision. SOA provides a powerful infrastructure for integrating disparate systems and technologies through services. However, current practice relies heavily on human intervention for such integration that provides little flexibility to the edge user during mission execution. We have developed and applied a tool-assisted method that allows the edge user to quickly identify non-organic systems and technologies[1] of interest, augmented with special services, and to integrate them dynamically to support a mission. This method for Agile Integration of Complex Systems (Agile Integration) was implemented and used in multiple demonstrations. As the mission evolves, the edge user can easily adapt the integrated solution.

Our emphasis is on integrating systems and technologies, not the customary service orchestration. Agile Integration is based on three graphically enabled SOA services (Graphically Enabled Discovery Service, Graphically Enabled Messaging Service, and Graphically Enabled Mediation Service) that use a mission-limited Community of Action (CoA) registry (see Graphically Enabled Messaging Service and Figure 4).

The remainder of this paper is organized as follows: SOA in DoD discusses the DoD mandate for SOA; Baseline SOA establishes a foundation for comparison with Agile Integration; the next six sections provide a detailed comparison of the three graphically enabled services with the corresponding baseline services; the final section summarizes the paper and suggests some areas for further investigation and development.

SOA in DoD

DoD has mandated that all systems support the Network-Centric Environment and SOA is fundamental to realizing DoD's Net-Centric Vision (DoD Architecture Framework (DoDAF) 1.5, volume 2, p. xiv and DoDAF 2.0, volume 1, p. 2). SOA is mandated by multiple DoD policies, reference architectures and models [see Appendix A: DoD SOA Mandates], and the acquisition process (Joint Capabilities Integration and Development System and 5000-series regulations).

Baseline SOA

In order to see what changes with Agile Integration, some reference point is needed. There are many definitions of SOA and many vendor[2] implementations. To establish a reference point, this paper describes a SOA baseline using the DISA NCES CDD[3]. The NCES CDD aligns with all of the DoD mandates for SOA and may provide the most comprehensive description of a pure SOA available. The description of the Baseline SOA in this section is not intended to provide a SOA tutorial, but only to provide enough information for comparison purposes, with an emphasis on those parts of the baseline where graphical enablement improves.

Beyond the baseline, the NCES CDD provides a good taxonomy, both of which are independent of product implementations, making them technology and vendor neutral. Vendor products use overloaded terms, e.g., Enterprise Service Bus (ESB), Governance, etc. The NCES CDD taxonomy decomposes what are sometimes viewed monolithically because of how they are bundled by vendors. This decomposition helps in multiple ways:

It clarifies what the monolithic functions are doing at a finer-grained level for comparing products (e.g., ESB is more than Messaging)

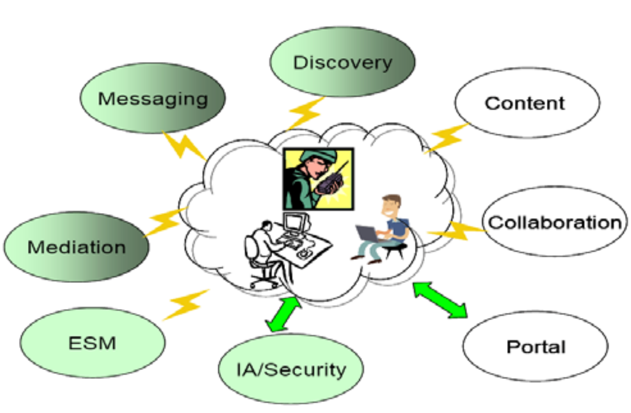
It helps to see how to do things differently (e.g., variation points)

It provides a partitioning of services to identify where issues arise (e.g., a Discovery, Mediation, or Messaging issue)

The NCES CDD describes what it refers to as Core Enterprise Services (CES) with a subset called the SOA Foundation (SOAF) Services[4]:

Figure 1 shows the CES, with the SOAF services shaded. Three SOAF services (darker shading) were specialized for Agile Integration and are discussed in more detail in the following sections (Figure 2, Figure 3, and Figure 6 provide high-level views of the three SOAF services affected by Agile Integration).

Figure 1: Core Enterprise Services provide net-centric infrastructure



Many products are available for these three SOAF services, often combining all or parts of them into an ESB. To be considered as providing a SOA compliant with the NCES CDD, the products must comply with industry standards such as eXtensible Markup Language (XML), Web Service Definition Language (WSDL), and Simple Object Access Protocol (SOAP). SOA products compliant with such industry standards are available both as open source and proprietary implementations[5].

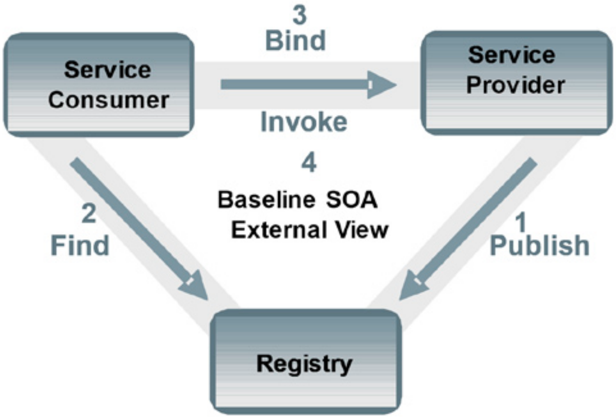
SOAF Service Discovery Service

The SOAF Service Discovery Service (Figure 2) is needed to link decoupled providers and consumers. Service Providers publish specified information (e.g., WSDL) to service registries (e.g., Universal Description Discovery and Integration—UDDI) for which they have been certified, providing standard format and interface information (e.g., WSDL). In the NCES CDD view, consumers do not register to be notified of the availability of services, but only discover existing services. Consumers (humans or services) use the SOAF Service Discovery Service to search known, widely accessible registries identified from various sources, including metadata registries; they then select the services they find that are of interest and the Discovery Service returns the endpoint and other interface details. Consumers must then assign (bind) the endpoints for the service in code (directly or indirectly through a proxy that is identified in the code) that can be interpreted at runtime or compiled for runtime (see Figure 2). In general, consumers of a service are not known in advance, and the time of use is not known in advance

CES SOA emerged out of the commercial sector where time-consuming human browsing or rules-based dynamic discovery of services is acceptable. Dynamic discovery not only introduces potentially large latencies, but a high risk of not finding exactly the desired service. As with the latencies, the cost of not finding the best service is more likely to be acceptable in commercial applications, especially those involving personal information gathering or purchasing activities, than in the life-and-death circumstances of combat or civilian disasters. Without dynamic discovery during mission execution, SOA loses one of its significant intended advantages—flexibility at runtime to select resources through services based on a rapidly changing mission situation.

In order to use dynamic discovery, (e.g., by fully utilizing UDDI) the consuming service must make a provision to search general purpose registries and determine that an identified service is appropriate. Determining appropriateness involves a sophisticated inference engine and complex decision rules. Processing the decision rules for each discovered service involves substantial processing. The combination of processing and search latencies would be potentially unacceptable for real-time applications, while still leaving some risk that the service would not meet the performance requirements. The latency and risk involved in finding the correct service, especially in real-time safety or mission critical applications, have limited the use of dynamic discovery largely to identifying transformations and adaptors (see Graphically Enabled Mediation Service). Consequently, in practice, service discovery often involves manual browsing of very large, general purpose registries for metadata or services, prior to execution of the service.

Figure 2: High-Level View of SOAF Discovery



The practical concerns about latency and risk leave little or no flexibility for the edge user at runtime, during mission execution. As discussed in the next section, Agile Integration performs the SOAF Service Discovery Service functions through the CoA registry to increase the edge user's flexibility during mission execution. The Agile Integration approach uses the mission-limited CoA registry (see Graphically Enabled Messaging Service) along with graphical enablement (see Figure 5 and Figure 7) to provide acceptable latency and known services to the edge user in life-and-death circumstances.

Graphically Enabled Discovery Service

Discovery for the edge user, who is the consumer in this context, is accomplished by noticing a provider-service icon (or consumer icon—see Graphically Enabled Messaging Service for discussion of not distinguishing icons for consumers from those for provider-services) on the edge user's display; finding the service is accomplished by selecting the icon. Whatever is placed on the edge user's display by the display service (see Graphically Enabled Messaging Service), as in Figure 5, would be available for the edge user's mission. Invoking a service and executing a workflow is accomplished by dragging and dropping the desired icons on the Orchestrate icon, as in Figure 7.

Some over-arching doctrine would be required for common understanding. For example:

Icons would stay on the display for a specified minimum time

Prior to dragging and dropping to the Orchestrate icon, icons might drop off the display because of needs updated by situational awareness—the edge user with the asset displayed might have a decreased need or another edge user might have a higher need

After dragging and dropping, the resource would be committed until mission completion (not available for contingencies for other missions—see contingency pools in Graphically Enabled Messaging Service and Graphically Enabled Mediation Service)

Mission planners would determine which displays to target for a given CoA registry, such as:

Traditional workstations in Command Operations Centers, Combat Operations Centers, and Tactical Operations Centers

Handhelds with multi-touch screens for dismounted warfighters or first responders

Laptops for mounted warfighters or first responders

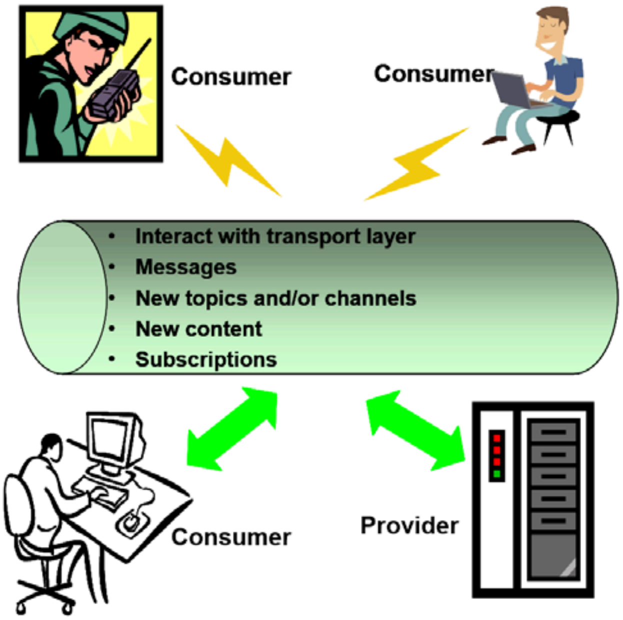
Multitouch would be useful for all displays, but especially for the edge user handhelds, for quick use in a hostile environment.

The need to search general purpose registries is eliminated by entering additional details in the CoA registry and limiting it to mission scope (see Graphically Enabled Messaging Service). The net effect is to provide information in the CoA registry that would be provided during design time or dynamic discovery in a SOAF application, including information needed for creating a workflow script, e.g., with Business Process Execution Language (BPEL) in a SOAF application (see Graphically Enabled Messaging Service and Graphically Enabled Mediation Service sections). Unlike the SOAF Service Discovery Service, Graphically Enabled Discovery Service does not register services; this is done by the Graphically Enabled Messaging Service. The reason that the Graphically Enabled Discovery Service does not register services is twofold. First, it no longer needs to search registries; and second, the Graphically Enabled Messaging Service manages subscriptions (as does the SOAF Messaging Service), so by also registering services, it is able to perform all of the functions to manage the CoA. Also, in contrast to the SOAF Service Discovery Service, there is no posting to general services registries; registries are not publicly known or widely accessible; consumers of a service are known in advance; and the time of use is known in advance.

SOAF Messaging Service

The SOAF Messaging Service provides basic distribution support for the other SOAF services, as shown in Figure 3. This support includes interacting with the transport layer through middleware such as Java Messaging Service (JMS), routing, queuing, and delivering messages, creating new topics and/or channels, transmitting content, and managing subscriptions.

Figure 3: High-Level View of SOAF Messaging Service



The DISA SOAF view of subscriptions only covers topics, channels, and content (data), not services, as noted in the SOAF Service Discovery Service section.

Graphically Enabled Messaging Service

Figure 4 presents a notional view of the relationships among Communities of Action (CoAs), Communities of Interest (CoIs), and Communities of Practice (CoPs), from most temporary (CoA) to most permanent (CoP). While roughly consistent with common definitions, the specific communities shown are not necessarily actual ones. The common definitions suggest more overlap than the distinctions used here, which are intended to show the idea that CoA registries are mission-limited and differ temporally and spatially from the general registries for large, long-lived CoPs or CoIs and are available through the Global Information Grid (GIG) or Internet.

Figure 4: Notional Relationship of CoA, CoI, and CoP[6]

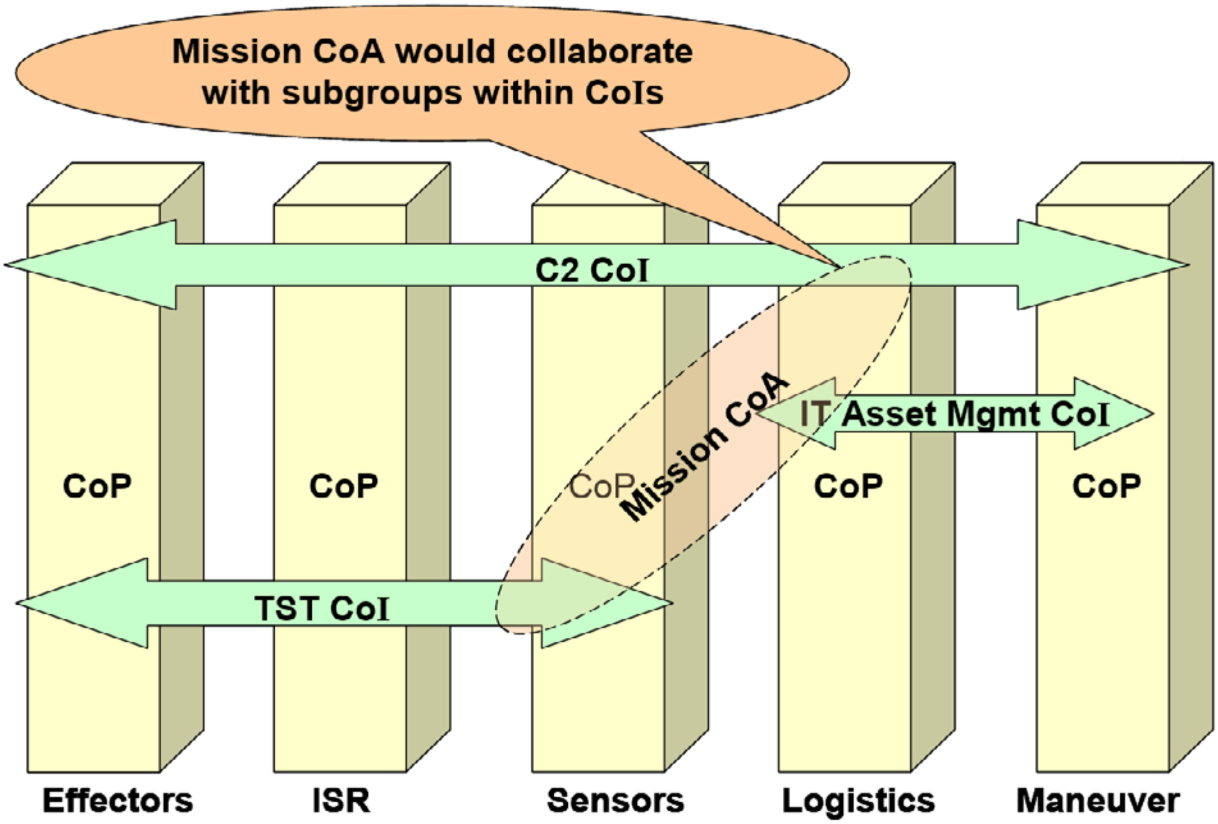
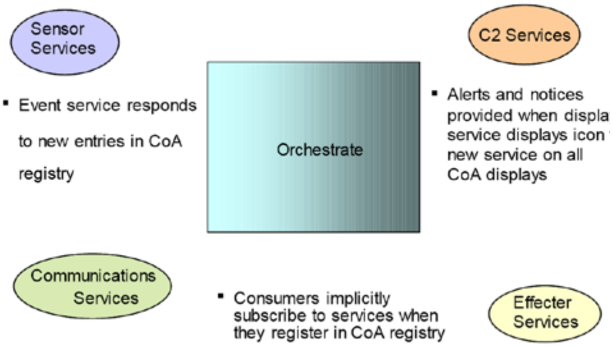


Figure 5: Graphically Enabled Messaging Service


Sensor Services
C2 Services
Event service responds to new entries in CoA registry
Orchestrate
Alerts and notices provided when displa service displays icon new service on all CoA displays
Communications Services
Consumers implicitly subscribe to services when they register in CoA registry
Effecter Services

Graphically Enabled Messaging registers consumers and their interests along with providers and services in the CoA registry. The consumer entry effectively subscribes the consumer to all of the services included with the registered consumer interests, and provides details to be used for graphically enabled mediation (see Graphically Enabled Mediation Service). In SOAF, these actions are split between two services: SOAF Service Discovery Service to register services (Graphically Enabled Discovery Service does not register services—see Graphically Enabled Discovery Service); and SOAF Messaging to manage subscriptions.

The CoA registry is graphically represented (see Figure 5), allowing finding, binding (assigning), and invoking to be done by selecting, dragging, and dropping (for orchestration). This brings to runtime SOA the kind of graphical support BPEL brings to design-time SOA.

A display service within the Graphically Enabled Messaging Service displays an icon for each provider-service and consumer in the CoA (Figure 5). This would be done through one or both of two actions in operational systems: the mission planner would select the CoA to be displayed at the time the edge user deployed, causing the icons to be displayed on all logged-in edge user displays and/or the edge user would log in and select the CoA for the mission.

The distinction between consumers and providers is not needed for the CoA registry. In practice, participants in the registry could consume and/or provide services, so related icons (e.g., on Figure 5) will be referred to simply as service icons. In designing a workflow for a SOAF application, designers and planners would select services, such as a netted weapon, that would consume another service, such as a sensor, and also be consumed by a targeting service. In the CoA, all three services would be entered with the details necessary to accomplish the respective interactions of consuming and providing services.

The combination of additional information entered in the CoA registry and the display of icons, as in Figure 5, accomplishes what would be done in preparation for creating a workflow script for a SOAF application (see SOAF Mediation Service) using manual or automated dynamic discovery to obtain information for the script. Such SOAF scripts either offer no runtime options to the edge user, require tedious selection steps, or rely on pre-determined rules (e.g., data driven), all offering no or little practical flexibility to the edge user. In Agile Integration, rather than completing the script for execution, the display allows the edge user to orchestrate the mission in the field (see Graphically Enabled Mediation Service). Once the mission is underway, execution may include any of the resources identified during design or planning, including a contingency pool that would be dynamically updated based on situational awareness (see below in this section).

Entering a service in a CoA registry using the Graphically Enabled Messaging Service also creates a new topic when needed for the middleware in use (e.g., JMS), as well as registering a service. SOAF Messaging also manages topics, so there is no change for Agile Integration regarding which service performs this function, but SOAF Messaging requires separate actions to create a new topic. Also, as with SOAF Messaging, Graphically Enabled Messaging routes, queues, and delivers messages.

Translators and adaptors are preprovisioned based on details in the CoA registry to reduce latency and the risk of not discovering the best ones at runtime. Planners and designers are supported by registration (utility) services to check adaptor and format needs of consuming services against the service to be consumed to ensure that translators and adaptors or their locations (e.g., URLs) are in the CoA registry.

The CoA registry will contain information on all resources for a mission, including a contingency pool. The contingency pool would include resources potentially needed by a mission, but not yet committed (e.g., displayed on the edge user's screen but not dragged onto the Orchestrate icon—Figure 5). The contingency pool could be dynamically managed as discussed later in this section.

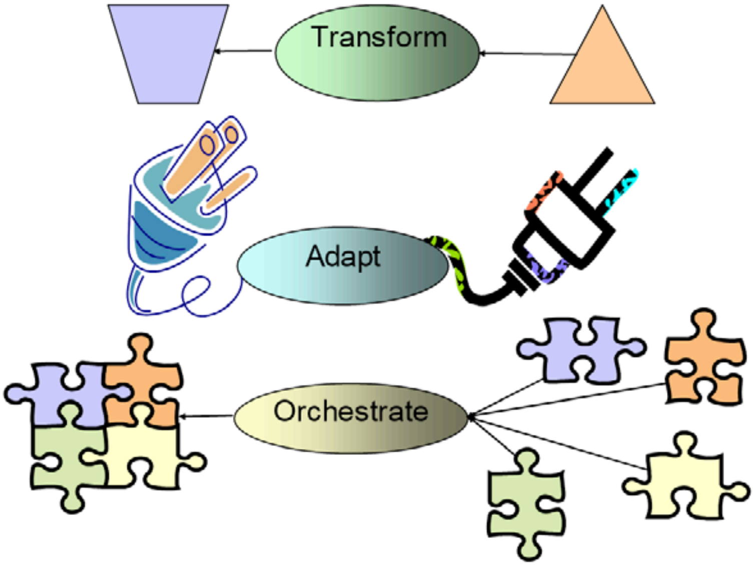
The following is an overview of how a CoA-based registry would be used for Agile Integration:

- **>> Designers and planners enter services in the CoA registry for a mission**
- **>> Consumer description (might also be a provider)**
- **>> Services, topics, or content to be consumed for a mission—implies a subscription**
- **>> Provider description (might also be a consumer)**
- **>> Services, topics, or content to be provided for a mission**
- **>> Adaptors and translators to be preprovisioned**

- **>> Display service graphically displays CoA registry (see Figure 5)**
- **>> Only icons of interest to the mission, because CoA is restricted to the mission**
- **>> Only entries available now (based on CoA entries modified by resource management—see dynamic reallocation below)**
- **>> An icon (other than the Orchestrate icon) represents an alert, notice, or discovered service when observed by the edge user**
- **>> Edge users select (find) icons of interest (see Figure 5), displayed from the CoA registry**
- **>> Drag to Orchestrate icon (bind [assign] endpoint for service)**
- **>> Drop on Orchestrate icon (invoke service)**
- **>> In the case of alerts or notices, dropping the icon onto the Orchestrate icon makes the topic or content available (e.g., current METOC data)**

Not only would the initial CoA for a mission provide agility to edge users, but the CoA could be updated dynamically based on situational awareness. Designers, planners, and edge users could release resources from the CoA, making them available to other missions, until the resources were committed by the edge user (as discussed above and in Graphically Enabled Discovery Service). In turn, planners could expand the mission's contingency pool by adding uncommitted resources to a CoA, even during a mission. A key to managing the contingency pool would be real-time resource management, with extensive dynamic scheduling and allocation of resources, including reachback to higher echelons and clear understanding, by all stakeholders, of the resource management rules. Fully automating the real-time resource management would require interfaces between the CoA registries and the resource management system. This is closely related to the basic understanding and over-arching doctrine outlined in Graphically Enabled Discovery Service above.

Figure 6: High-Level View of SOAF Mediation Service



The CoA registry could be populated by mission designers and planners or by consumers and providers who have been certified and given the registry location with entry requirements.

The CoA registry could also be populated with services by a software agent. The agent would search as many registries as needed, based on rules constructed from the consumer details in the CoA registry. This would not be an option for populating the consumers unless general registries contained consumer details for services, not just subscriptions to topics or content.

SOAF Mediation Service

The SOAF Mediation Service (Figure 6) enables integration of services from disparate systems. SOAF Mediation allows services to interoperate by transforming data formats (e.g., using XML schemas) and supplying protocol adaptors. The SOAF Mediation Service also supports workflow scripts for orchestrating multiple services.

Transformation and adaptation are two of three use cases for the SOAF Mediation Service (the other is Orchestration). The consumer (a human or consumer service) must explicitly request a format transformation (e.g., using an XML schema) for specified content or adaptation for a specified protocol. Providers and consumers[7] develop standards-based (e.g., UDDI for registration) workflow scripts (e.g., with BPEL) and register them with the Mediation Service for orchestration. The consumer (same or different than the one who registered the workflow) explicitly executes the workflow script through the Mediation Service.

The steps involved in any of the three SOAF mediation use cases contribute significant latency, along with the risk that no appropriate format translator, adaptor, or workflow script (e.g., out of date endpoints) exists.

Graphically Enabled Mediation Service

The diagram in Figure 7 is similar to the display, managed by the display service, used in our demonstrations. The CoA registry provides sufficient information to develop workflow scripts encompassing all three SOAF Mediation Service use cases—transformation, adaptation, and orchestration.

In a SOAF application, the information for transformation and adaptation is obtained at runtime by discovery services from general purpose metadata registries, with the latencies and risks noted in the SOAF Mediation Service section. For SOAF orchestration, designers and planners create workflow scripts, including service endpoints, to be registered and executed for the mission. The executing service (consuming or providing) would have the responsibility for determining whether a transformation or adaptation was required and for discovering the necessary schemas and adaptors.

A key difference between the Graphically Enabled Mediation Service and the SOAF Mediation Service is in localizing the information required in the CoA for simple, quick access at design time, plan time, or runtime, greatly reducing runtime latency. This localization allows the orchestration step to be deferred until runtime in Graphically Enabled Mediation, giving the edge user the agility to select the most effective mission capability based on the situation in the field; the workflow is not executed, and resources are not committed, until the corresponding icons are dragged and dropped onto the Orchestrate icon.

In a mission using SOAF workflows prepared in advance, the edge user's actual situation might not need all of the planned resources and might need other resources not originally planned. With real-time resource management in Agile Integration, the edge user would return resources to be available to other missions by deselecting them. New contingency resources, in turn, would be displayed on the edge user's device as they are entered in the CoA (see Graphically Enabled Messaging Service), allowing the edge user to commit additional resources to the mission. Discipline on the part of edge users in returning and adding resources would clearly increase the overall effectiveness of the Graphically Enabled Mediation Service.

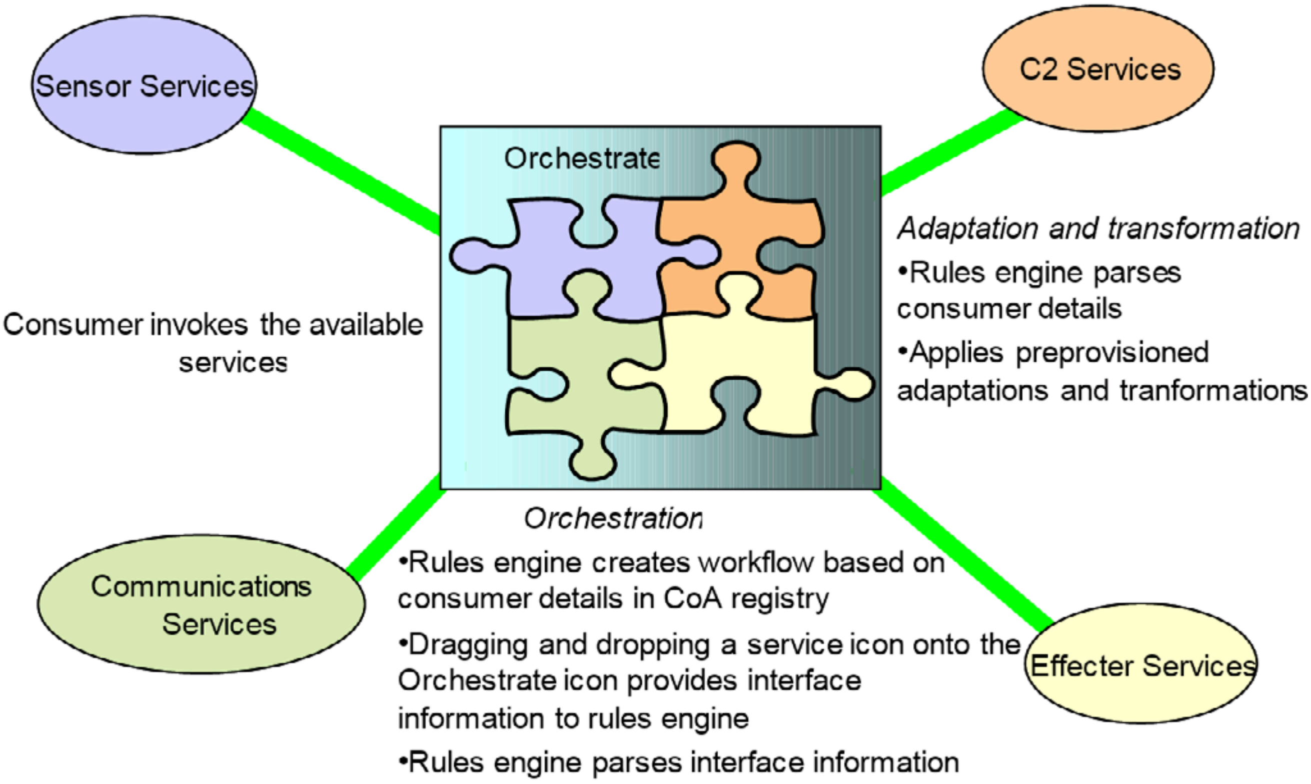
Dragging and dropping an icon that provides service onto the Orchestrate icon causes a simple rules engine (not a full inference engine) to extract the CoA registry information associated with the service. Likewise, dragging and dropping an icon that consumes a service onto the Orchestrate icon causes the rules engine to extract the corresponding CoA registry information describing the consumer and the services it needs to consume. The rules engine parses the extracted information to obtain details such as the following:

>> **Endpoints**
>> **Operation signatures**
>> **Services of interest to a consumer**
>> **Protocols**
>> **Formats**

The rules engine then connects the services to the interested consumers by inserting the service endpoints in the consumers' interface code (e.g., WSDL) and provides the required transformations and adaptations with preprovisioned translators and adaptors. The preprovisioning reduces both latency and risk by assuring that the right translators and adaptors are immediately available at runtime. This would continue for additional services and consumers until the desired orchestration was complete, based on details from the CoA and/or explicit input from the edge user; execution would begin upon such completion.

Multiple tree structures could be used as an alternative to the type of display shown in Figure 7. Two separate trees would be used, with one corresponding to the service icons in Figure 7 (i.e., corresponding to the CoA registry), and the other (in a separate pane) corresponding to the mission (equivalent to the Orchestrate icon). Mission planners would drag items from the service tree to the mission tree, creating separate mission trees for planning and comparing multiple missions. While tree structures could also be used by edge users, the type of display in Figure 7 seems easier to use under intense pressure. As a further extension, CoP and CoI registries, in addition to the CoA registry, could be shown in separate trees. Multiple CoA registries could then be built from the CoI and CoP registries during mission design and planning.

Figure 7: Graphically Enabled Mediation Service—The diagram first appears on the display as in Figure 5. When the user drags the service icon to the Orchestrate icon and drops it, it bounces back to its original position while creating the green line, showing that it is connected.


Sensor Services
C2 Services
Orchestrate
Adaptation and transformation
•Rules engine parses consumer details
•Applies preprovisioned adaptations and tranformations
Consumer invokes the available services
Orchestration
•Rules engine creates workflow based on consumer details in CoA registry
•Dragging and dropping a service icon onto the Orchestrate icon provides interface information to rules engine
•Rules engine parses interface information
Communications Services
Effecter Services

Summary

Agile Integration provides the edge user increased flexibility during mission execution, both directly by selecting the resources needed for the mission based on the situation during mission execution, and indirectly by releasing resources not needed to a contingency pool, increasing the likelihood of having additional resources available across multiple missions. The more missions involved in the contingency pooling aspect of Agile Integration, the greater would be the agility for the edge user and overall efficiency and effectiveness of resource use.

The basic graphically enabled services and CoA registry described in this paper were demonstrated, but there are also extensions that would be worth investigating and potentially developing. One such extension would be to expand resource contingency pools through real-time resource management, including interfaces between resource management and the CoA registry. Such interfaces would allow increased access to available resources for the contingency pool, using the automated resource allocation and scheduling algorithms of the resource manager, based on battlespace awareness across multiple missions. The dynamic allocation and scheduling of resources could include reach-back to higher echelons.

Another extension would be assessing the related implications for doctrine and training to maximize the agility and benefits of the contingency pool. Edge users would need to fully understand the importance of releasing assets they did not need as soon as possible to obtain maximum effectiveness from the contingency pool, considering both the benefit to other missions and to themselves as edge users from other missions did the same. While the doctrine could be readily established, indoctrinating the required discipline to give up assets through training might be a significant challenge.

The CoA registry feature could also be extended by populating it with services by a software agent. The agent would search as many general purpose registries as needed, based on rules constructed from the consumer details in the CoA registry, providing an automated way to use general purpose CoI and CoP registries in conjunction with the CoA.

The current design of the CoA registry includes all details required for discovery and mediation; rules use these details to dynamically create workflows during mission execution, eliminating the need for the separate workflow registry used by the SOAF Mediation Service. However, performance issues might require extending the CoA registry by separating details related only to dynamic workflow creation into a separate sub-registry for large missions.

Another extension would be the use of multiple tree structures representing the service icons. The tree structures could be useful to designers and planners by displaying large numbers of alternative resources concisely in separate trees for multiple missions. Also, CoP and CoI registries could be shown in separate trees to assist planners and designers in identifying resources.

ABOUT THE AUTHOR

Wayne O'Brien is an Engineering Fellow in the Intelligence and Information Systems unit of Raytheon Company. He has worked for Raytheon for eighteen years, focusing on software systems architecture with model-driven development for the past twelve years. Beginning in 2006, he was the architecture SME or lead on several Enterprise Campaigns and is currently assigned as Technical Director on an international program.

Wayne received his Ph.D. in Information Technology and Engineering in 2006 from George Mason University and became a Raytheon Certified Architect in 2009. He is the inventor on one granted patent and the inventor on four Raytheon patent applications; one additional patent disclosure is under review. Wayne's doctoral dissertation was published in book form (Breakdowns in Controls in Automated Systems) in the Untied States and Europe in December of 2008. A related article was published in November of 2008 in the Journal of Systems and Software.

Phone: 540-905-3203
E-mail: wayne_p_obrien@raytheon.com

NOTES

1. E.g, netted sensors and effectors, command and control
2. Vendor in this paper refers to any provider of a SOA product, including open source.
3. DISA. Capability Development Document (CDD) for Net-Centric Enterprise Services (NCES). Increment 1.0, Version 1.0, May 2006.
4. These are referred to as services, but each actually consists of multiple operations that are invoked through standards-based (e.g., WSDL) external interfaces and internally through method calls.
5. E.g., Mule Enterprise Service Bus; JBoss Software; Apache Synapse ESB; WSO2 Registry; IBM Websphere SOA products; Oracle SOA Suite; SOA Software Policy and Repository Managers; Microsoft Web Service Software Factory; and Universal Description, Discovery, and Integration (UDDI) Services.
6. Graphic based on input from DISA and Al-Masyabi, Walid. Advancing SOA Interoperability in the Net-Centric Enterprise. Raytheon 2008 SOA Workshop.
7. The NCES CDD only suggests that consumers develop and register workflows. However, providers might also offer workflows as services.

APPENDIX A: DOD SOA MANDATES

1. Defense Information Enterprise Architecture (DIEA)–Presents the vision for net-centric operations and establishes near-term priorities to address critical barriers that must be overcome to achieve that vision (intended to replace the NCOW RM (which has been withdrawn from public access to avoid confusion with the DIEA guidance), the Net-Centric Checklist, and related enterprise architecture guidance).

2. Net-Centric Checklist (NCC)–Assists program managers in understanding the net-centric attributes that their programs need to implement to move into the net-centric environment as part of a service-oriented architecture in the Global Information Grid (GIG).

3. Net-Centric Data Strategy (NCDS)–Defines a modified paradigm for data management within DoD that expands the focus to visibility and accessibility of data rather than just standardization.

4. Net-Centric Services Strategy (NCSS)–Expands upon the DoD Net-Centric Data Strategy (May 2003) by connecting services to the Data Strategy goals.

5. Net-Ready Key Performance Parameter (NR-KPP)–Measurable and testable characteristics and/or performance metrics required for timely, accurate, and complete exchange and use of information to satisfy information needs for a given capability.

6. Net-Centric Enterprise Services (NCES)–DISA's approach to enable the secure, agile, robust, dependable, interoperable data-sharing environment for DoD where warfighter, business, and intelligence users share knowledge on a global network.

7. Net-Enabled Command Capabilities (NECC)–Led by DISA, Department of Defense's (DoD's) principal command and control (C2) capability focused on providing the warfighter with the data and information needed to make timely, effective and informed decisions.

8. Communities of Interest (CoI)–over 50 multi-agency bodies chartered to develop common data and services schemas in particular domains.

9. Multi-Agency SOA Consortium–DISA-sponsored cooperative body involving NCES, NECC, Navy CANES, Army FCS, Air Force AOC, and potentially other programs.

10. DoDAF 2.0 – serves as the overarching, comprehensive framework and conceptual model enabling DoD managers at all levels to make key decisions more effectively through organized information sharing across Department, Joint Capability Areas (JCAs), Mission, Components and Program boundaries