2004 Command and Control Research and Technology Symposium
The Power of Information Age Concepts and Technologies

Air Force Information Workflow Automation through Synchronized Air Power Management (SAPM)

Carl Benkley
cbenkley@mitre.org
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730-1420

Irene Chang
inchang@mitre.org
The MITRE Corporation

John Crowley
jdcrowley@mitre.org
The MITRE Corporation

Lt. Thomas Oristian, USAF
Thomas.Oristian@hanscom.af.mil
Electronic Systems Center (ESC/ACU)
50 Griffiss Street
Hanscom AFB, MA 01731-1625

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **JUN 2004**	2. REPORT TYPE	3. DATES COVERED **00-00-2004 to 00-00-2004**
4. TITLE AND SUBTITLE **Air Force Information Workflow Automation through Synchronized Air Power Management (SAPM)**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **The MITRE Corporation,202 Burlington Road,Bedford,MA,01730-1420**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**20**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Abstract

Utilizing emerging information technologies, the Synchronized Air Power Management (SAPM) initiative presents an automated business process to integrate the Air Force (AF) Command and Control (C2) systems at the Wing level. The SAPM Phase I proof-of-concept effort has demonstrated a significant reduction in the time to plan unit taskings, evaluate missions, and execute decisions. SAPM is a joint venture among ESC/AC, AFC2ISRC/DO, MITRE, Microsoft, Lockheed Martin, Northrop Grumman, and DSRC. Implementing Extensible Markup Language (XML) messages, web services, and workflow automation, SAPM expands existing web-based capabilities, enables machine-to-machine interfaces, and streamlines the war fighter kill chain process. SAPM Phase I was successfully demonstrated to senior AF officers and representatives of DoD. Phase II is being developed at the MITRE facility in Bedford, Massachusetts.

Background

The AF war fighter kill chain is an operational process that cuts across most of the existing Battle Management Command, Control, and Communications (BMC3) systems. The traditional focus on building and acquiring individual systems has made it difficult to fully capture and implement this process as an end-to-end information workflow. The result has been stove-piped systems where data has to be manually entered or reentered, where status of the ongoing process is not easily visible, and where reports such as Wing commanders' briefings and mission reports have to be constructed off-line. To address those issues, the AF ESC has been advocating C2 enterprise integration. The goal is to integrate C2 systems, components, and services into the broader view of the operational C2 entities using appropriate architectural frameworks and business processes to ensure affordability, military utility, efficiency, and timeliness. With the same objective in mind and by implementing information workflow automation, SAPM initiative offers a modern technology process toward the C2 enterprise integration.

SAPM Concept

In January 2003 the SAPM initiative began as a collaborative prototype effort of ESC/AC, AFC2ISRC/DO, MITRE, Microsoft, Lockheed Martin, Northrop Grumman, and DSRC. Exploiting XML, web service, and workflow automation technologies as well as the Universal Description, Discovery and Integration (UDDI) registry, the SAPM Phase I proof-of-concept was completed in May 2003.[1] It has demonstrated a drastic reduction in the time it takes to plan, evaluate, and execute decisions, as well as a decrease in associated manpower needs in a lab environment.

---

[1] Other SAPM Phase I development team members included D. Hebert , D. Konstantopoulos, T. McDevitt, J. Sexton (MITRE); E. Rosenkranz, D. Stampfli (Microsoft); S. Allen (AFC2ISRC); B. Reed (Nothrop Grumman), B. Donohue (LMMS), R. Guerrero (DSRC); S. Taylor, R. Raymond (DRC); D. Mirra (Quantec)

The concept was to build web services to automate the information flows and improve interoperability between TBMCS Force Level (FL), TBMCS Unit Level (UL), Joint Mission Planning System (JMPS), and Joint Weather Impacts System (JWIS). SAPM Phase I implemented workflow management capabilities within and among the above mentioned C2 systems to provide commanders status and control from the generation of the air battle plan all the way through the creation of detailed mission planning routes. These systems are either J2EE or .NET enabled running within UNIX and Windows environments. The SAPM architecture is illustrated in Figure 1.

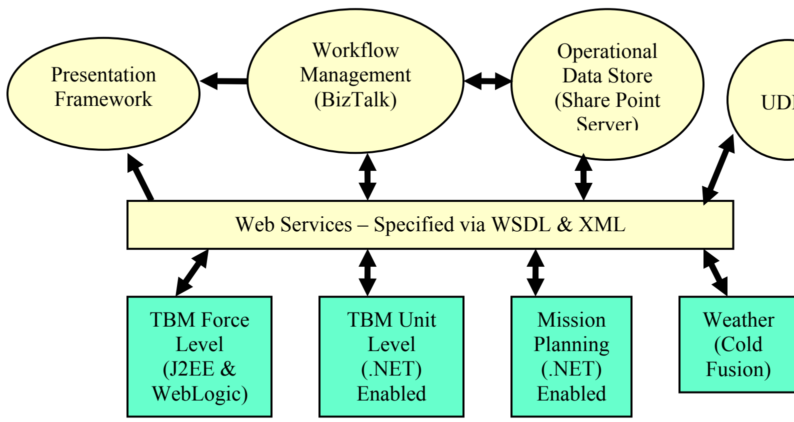

Figure 1. SAPM Architecture

All SAPM web services are Simple Object Access Protocol (SOAP) based, and the information will flow as XML files. SAPM web services are defined by the World Wide Web Consortium (W3C) Web Services Definition Language (WSDL). For example, a UL scheduling web service provides scheduling information such as tail number, pilot name, and Standard Configuration Load (SCL). A mission planning route web service provides detailed route information in Common Route Definition (CRD) format. The vision is that all of these web services will be consumed by the AF C2 systems to automate existing interfaces that are either "hand-jammed" or partially automated.

Another capability demonstrated by SAPM is the ability to manage the workflow across multiple systems. An example of workflow control would be to provide automated control of the generation and delivery of the Air Operations Database (AODB) Oracle Database Exchange (ORDBEX) file from TBMCS Force Level to TBMCS Unit Level. This would allow a commander to obtain status as to the availability of the updated AODB on both the FL and UL sides as well as providing more automation of this complex transfer of information. Detailed implementation of SAPM web services and workflow automation is described in the sections below.

Air Force C2 Programs Overview

TBMCS FL is used by the Air Operations Center (AOC) to plan and execute theater-level air campaigns in support of joint operations. TBMCS supports all phases of the C2 cycle, provides the strategic planning through target analysis, defensive planning, airspace planning, and task analysis. The output of this planning phase provides detailed air battle plans constructed to provide tasking for the air battle forces. Once the tasking has been accomplished, the unit level systems begin the scheduling and mission preparation activities. After flight scheduling and mission planning are completed, detailed flight missions will be executed. Finally the flight missions will be assessed and the report will be generated and fed back to the AOC.

The TBMCS UL scheduler application assigns crew members, tail numbers, and SCLs to mission sorties in support of tasking and publishes daily flying schedules. During peacetime, Wing commanders use the Unit Level system as a resource management tool. During contingencies, TBMCS Unit Level is used to support Wing level operations, planning, logistics, and intelligence activities.

JMPS provides the mission planner with unit level mission planning support for every phase of a mission, ranging from the preflight planning, departure, attack/cargo delivery, deconfliction, recovery, and post-mission debrief. JMPS information management tools allow the user to access data and develop a flight plan for the combat mission. JMPS supports collaborative planning of all mission elements including attack, bomber, cruise missiles, fighter, assault, airborne early warning, command, control and communications, etc. JMPS can transmit, accept, and process large amounts of near real-time data on a frequent basis. It has the capacity to rapidly process these data to create a situation updated threat (or weather, terrain, routes etc.) picture as well as to display new information to the planner when requested.

Workflow Automation Overview

The goals of workflow automation are to provide flexibility, visibility, auditability and extensibility to process management. Workflow automation has often been solely viewed as an information technology solution. To be a success, it needs to involve all segments of an organization and requires a thorough understanding of their roles. In addition to the benefits identified above, a successful workflow automation project helps organizations achieve a clearer understanding of the relationships that exists between different systems and frequently discover more robust processes and solutions.

Developers and implementers of workflow automation systems need to perform a critical analysis of how the solution will add organizational value and determine the potential return on investment prior to investing the time and resources into the process analysis and system integration. Often the core areas that provide the greatest value to the organization have heuristic traits that cannot be fully captured by an automated process, for example, mensurating a target. These processes can still greatly benefit from orchestrating and tracking the affiliated tasks.

Workflow automation is not designed to completely alter the way a business operates, rather it is designed for management and process participants to assert control over their existing processes, and to provide the flexibility to modify these processes when necessary to reflect the dynamic environment where they are deployed.

Solutions need to emphasize the fundamentals of defining and managing process information. As integration standards, tool sets, and protocols continue to evolve, successful solutions will be defined by focusing on the process and management layers

Leveraging Current Infrastructure and Investments

One of the mantras within the workflow automation space is the introduction of automated workflow solutions that leverages an organization's existing infrastructure. The theory is that workflow automation permits organizations to realize request of interest from their existing applications by abstracting processes to a higher level, allowing them to be connected and orchestrated by a workflow automation solution.

The argument assumes that there is a driving need to integrate the systems in question and that no alternative steps have been taken to optimize the processes and movement of information between the systems. Due to operational and competitive pressures, most organizations have performed and continue to optimize their systems and processes either manually of through traditional system integration tasks.

It is important that workflow automation processes do not simply repeat these optimizations but rather look to increase the efficiency, flexibility and reliability of the processes.

Improving Process Management and Control

A promise of workflow automation is the ability to achieve greater process control and end-to-end process visibility.

A workflow automation solution is not simply a collection of integrated systems, rather a choreographed network of systems and capabilities that can quickly adapt to a changing environment and incorporate new opportunities by managing information in a process-centric/Service Oriented Architectures (SOAs) manner. SOAs are an enterprise architectural style designed to achieve loose coupling among interacting systems. Each system is designed to fulfill specific tasks for a requester. By loose coupling the interface between systems an internal change in a service will have no effect on the requesters of that service.

Workflow automation solutions demonstrate real value when an organization needs to adapt or change in order to minimize the effects of the change or maximize the opportunity that it may uncover. Process optimization requires gathering metrics on the processes under management. These metrics may include the time to complete each task, users associated with tasks, network bandwidth, and process resolution. As such, the workflow management system needs to be integrated into the overall information architecture of the organization.

The gathering of process intelligence and the process performance metrics discussed above should be built into the workflow management solution. These metrics can be analyzed and mined for valuable information pertaining to the managed processes and to discover previously unrecognized operational and data patterns.

With the successful adoption of workflow automation tools, organizations will be able to respond more rapidly and with greater confidence to change and to optimize their existing processes, with the performance and process metrics providing management with a clearer understanding of the cause-and-effect relationships that exist between process elements.

Technical Issues of Workflow Automation

Workflow Automation is being driven by the formation and rapid acceptance of key industry standards maintained by the W3C, IEEE, and OASIS. It relies on the capability of the individual components, systems and organizational units to communicate in an economical way with suitable performance. Typical integration methods include SOAP, and XML based protocol requiring little if any new organizational infrastructure or network configuration.

Currently evolving standards such as the Business Process Modeling Language (BPML) and the Business Process Execution language (BPEL) will insure that workflow automation solutions are more focused on process management and organizational needs and less on the difficulties of system integration.

Workflow Design Strategy

Comparing similar organizations using a collection of common systems, it is the organizational processes and the application of organizational intelligence that permits the discrimination between groups.

Though workflow automation solutions help organizations manage their workload and processes, it does not, per se, introduce excellence or a competitive advantage. If correctly designed and implemented, workflow automation can, however, provide an infrastructure that can be leveraged for a competitive advantage.

A bottom-up approach to workflow automation begins with analyzing an organization current systems and data models. Designing automation systems from the bottom-up approach tends to result in a typical systems integration process driven by the current processes and systems. Often the approach results in an inflexible solution that is expensive to modify. Preferably a top-down approach to designing workflow management systems focuses on analyzing the organizational processes and the dynamics governing these processes.

Many common business processes exist between the different AF operational units. These units will broadly tend to use similar applications and systems to support their operational requirements, though some systems may be highly specialized for the unit's particular mission. The monitoring and reporting requirements are also quite similar. The focus of the effort involved in creating workflow automation solutions within this environment should be in providing the correct set of workflow tools and techniques that will encapsulate the common elements while permitting the unique aspects to flourish.

The further the operational organization can move away from managing interfaces and technical micro-detail, the better. SAPM utilizes Microsoft's BizTalk server as an orchestration engine. The BizTalk solution provides a graphical process modeler which allows an analyst to develop a business process by using intuitive graphical elements to represent different services and consumers of information within the process. A snapshot of the SAPM BizTalk process flowchart is show in Figure 2.

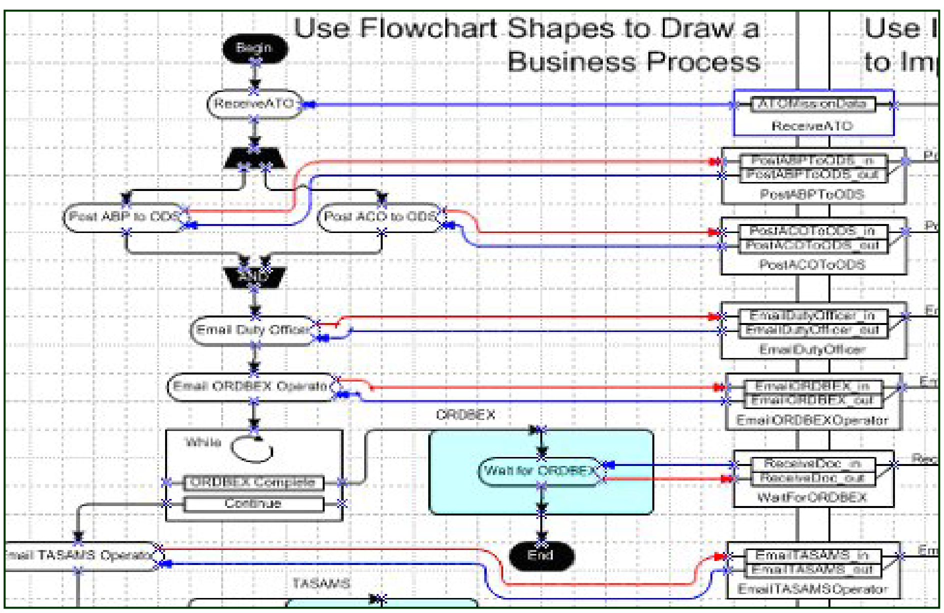

Figure 2. SAPM Biztalk Process Flowchart

If we are to successfully manage processes from a model level, we have to have the necessary technical plumbing in place to fill the void between the process model and low level technologies.

The Standard Workflow Reference Model

The standard workflow reference model (Figure 3) has been developed from analyzing generic workflow application structures and by identifying the interfaces required within this structure that enable different products to interoperate. All workflow automation systems contain a number of generic components that interact in a defined set of ways; different commercial products typically exhibit different levels of capability within each of the identified components. To achieve interoperability between workflow products a standardized set of interfaces, such as web services or CORBA, and data interchange formats between such components is necessary.

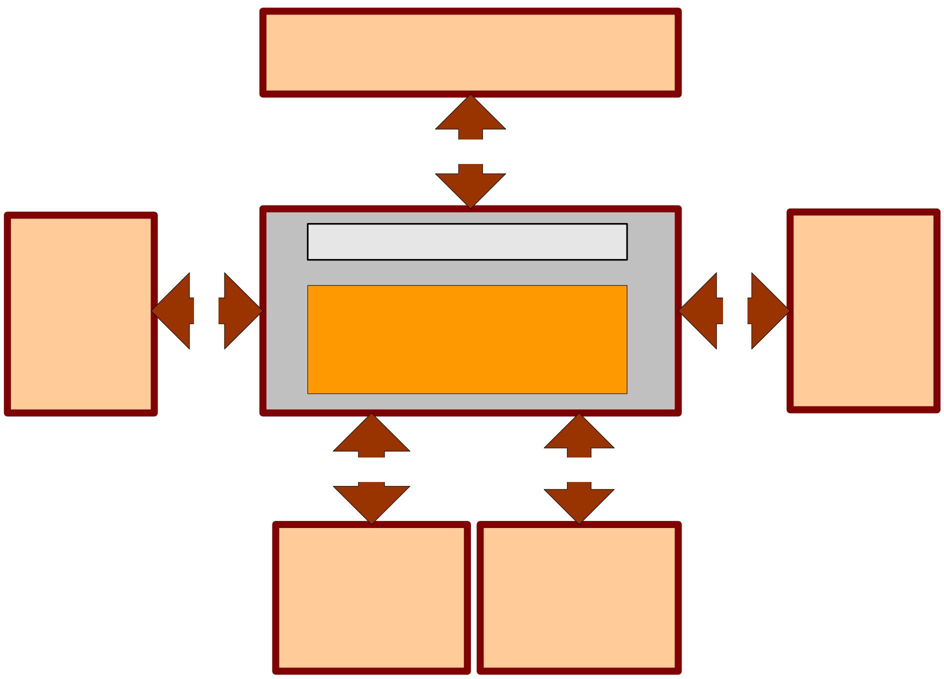

Figure 3. Standard Workflow Reference Model

Interface 1: Process Definition Tools Interface

The process definition tools interface defines a standard interface between process definition modeling tools and the workflow engine(s). The process definition and modeling tools should be capable of publishing the workflow process to the workflow engine in a standard format such as BPEL. The customary users of the process modeling tools are the process engineers. Many commercially available workflow engines, such as Microsoft's BizTalk, bundle the process modeling tool directly into their product offering.

Interface 2: Workflow Client Application Interface

The workflow client application interface defines an Application Program Interface (API) for client applications to request services from the workflow engine to control the progression of processes, activities and work-items. Often the client application will communicate automatically with the workflow engine, for example a client application may automatically notify the workflow engine that a user has performed a specific task such as updating a database. In other scenarios a user may need to directly notify the workflow engine that a task has been completed.

Interface 3: Invoked Application Interface

The invoked application interface defines a collection of APIs that allow the workflow engine to invoke a variety of applications, through common agent software. Common standard interfaces models include web-services and CORBA.

Interface 4: Administration & Monitoring Tools Interface

The administration and monitoring interface defines how different applications can integrate with the workflow engine to provide monitoring and control functionality.

Interface 5: Workflow Interoperability Interface

The workflow interoperability interface defines an interoperability model to support the interconnection of multiple workflow systems. The Workflow Management Coalition, an organization composed of industry experts and vendors, is promoting a common standard, wf-XML, as a solution for providing this interoperability.

Workflow Design Patterns

Workflow processes can be decomposed into an orchestrated collection of common design patterns. The workflow design patterns represent those common process elements such as conditional execution branches and cycles that can be found in all processes. They do not define the underlying data model required to support an instantiation of the pattern.

The process engineer uses the selected process definition tool to interweave different design patterns to create a complete process. Abstractly, the process definition tool does not need to be aware of the data model associated with the workflow process to design the process; however in practice the data model must be known to actually orchestrate the process.

(1) Basic Control Patterns

- Sequence: execute activities in sequence.
- Parallel Split: execute activities in parallel.
- Synchronization: synchronize two parallel threads of execution.
- Exclusive Choice: choose one execution path from many alternatives.
- Simple Merge: merge two alternative execution paths.

(2) Advanced Branching and Synchronization Patterns

- Multiple Choice: choose several execution paths from many alternatives.
- Synchronization Merge: merge many execution paths. Synchronize if many paths are taken. Simple merge is used if only one execution path is taken.
- Multiple Merge: merge many execution paths without synchronizing
- Discriminator: merge many execution paths without synchronizing. Execute the subsequent activity only once.
- N-out-of-M Join: merge many execution paths. Perform partial synchronization and execute subsequent activity only once.

(3) Structural Patterns

- Arbitrary Cycles: execute workflow graph without any structural restriction on loops.
- Implicit Termination: terminate if there is nothing to be done.

(4) Patterns Involving Multiple Instances (MI)

- MI without synchronization: generate many instances of one activity without synchronizing them afterwards.
- MI with a priori known design time knowledge: generate many instances of one activity when the number of instances is known at the design time (with synchronization).
- MI with a priori known runtime knowledge: generate many instances of one activity when a number of instances can be determined at some point during the runtime (as in FOR loop but in parallel)
- MI with a priori no known runtime knowledge: generate many instances of one activity when a number of instances cannot be determined (as in WHILE loop but in parallel).

(5) State-Based Patterns

- Deferred Choice: execute one of the two alternatives threads. The choice of which thread is to be executed should be implicit.
- Interleaved Parallel Routing: execute two activities in random order, but not in parallel.
- Milestone: enable an activity until a milestone is reached.

(6) Cancellation Patterns

- Cancel Activity: cancel (disable) an enabled activity.
- Cancel Case: cancel (disable) the process.

SAPM Workflow Implementation

The SAPM solution was implemented according to the standard workflow reference model described in previous paragraphs. The SAPM workflow implementation model is illustrated in Figure 4. Microsoft's BizTalk server tool was selected as the workflow engine to orchestrate the process. A simplified view of the SAPM workflow process model is depicted in Figure 5.

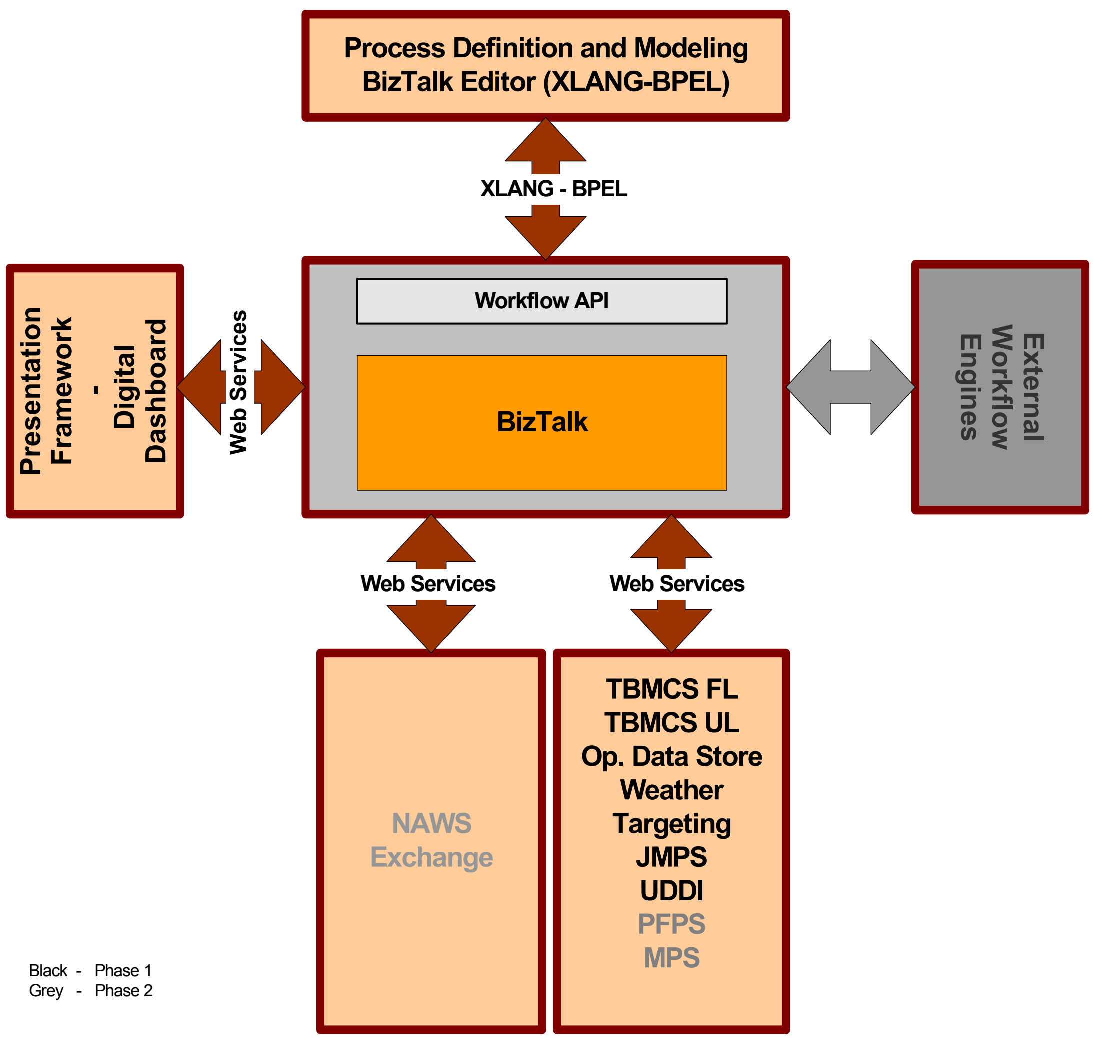

Figure 4. SAPM Workflow Implementation Model

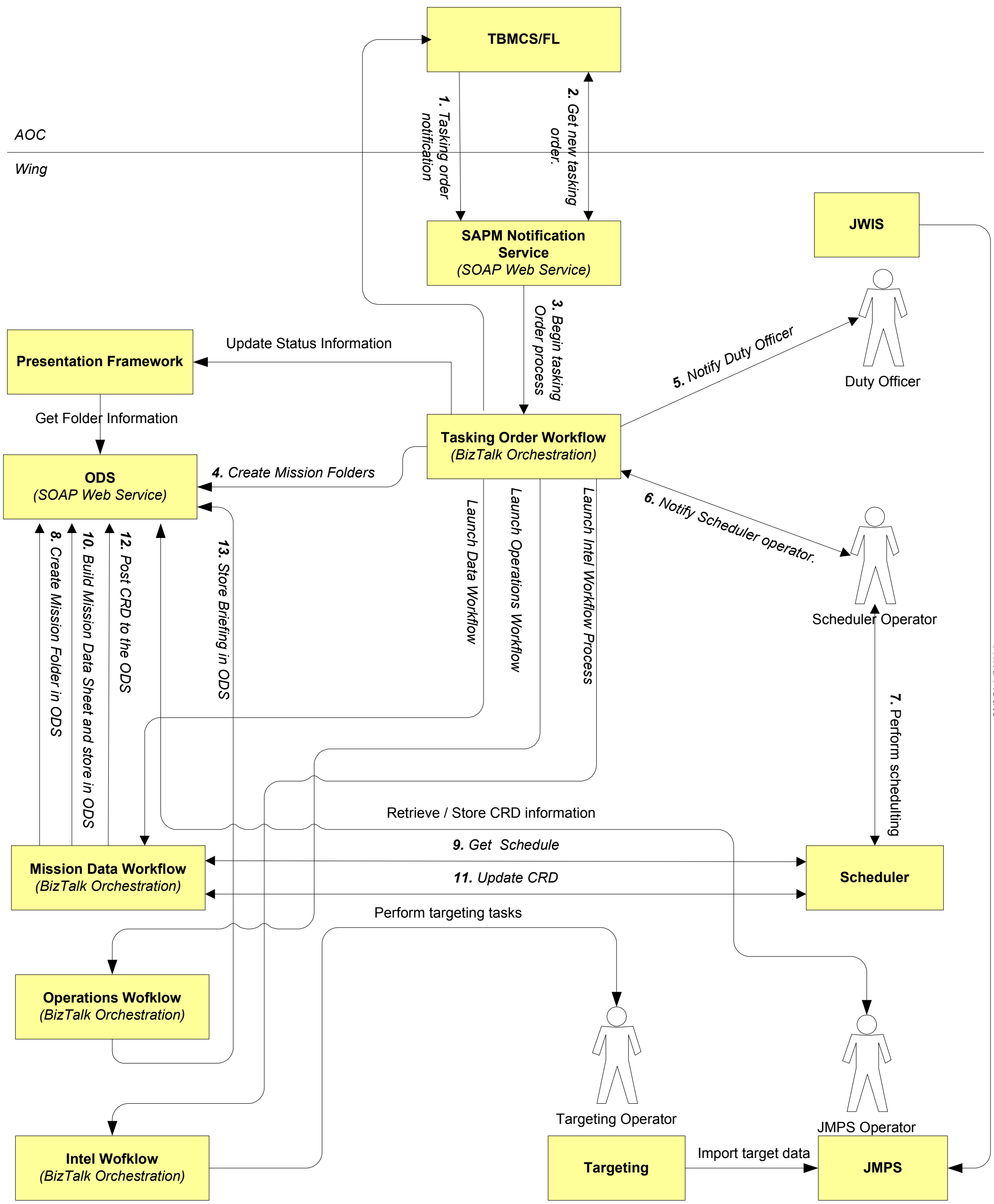

Figure 5. Primary SAPM Process Model - Simplified View

A principal element of a successful workflow project is the capability to adapt dynamic changes. Changes causing process modification can be additive. They could be caused by the inclusion of new systems or precipitated by a network or system failure. The SAPM workflow application monitors the status of the network and individual systems to insure that it is capable of completing its tasks. If a service is unavailable through difficulties with the actual service or the network, then alternate paths of execution will be executed.

The SAPM presentation framework (Figure 6) gives commanders and operational personnel a global view into their mission planning environment, clearly representing the status of the different missions being planned and the status of the systems involved.

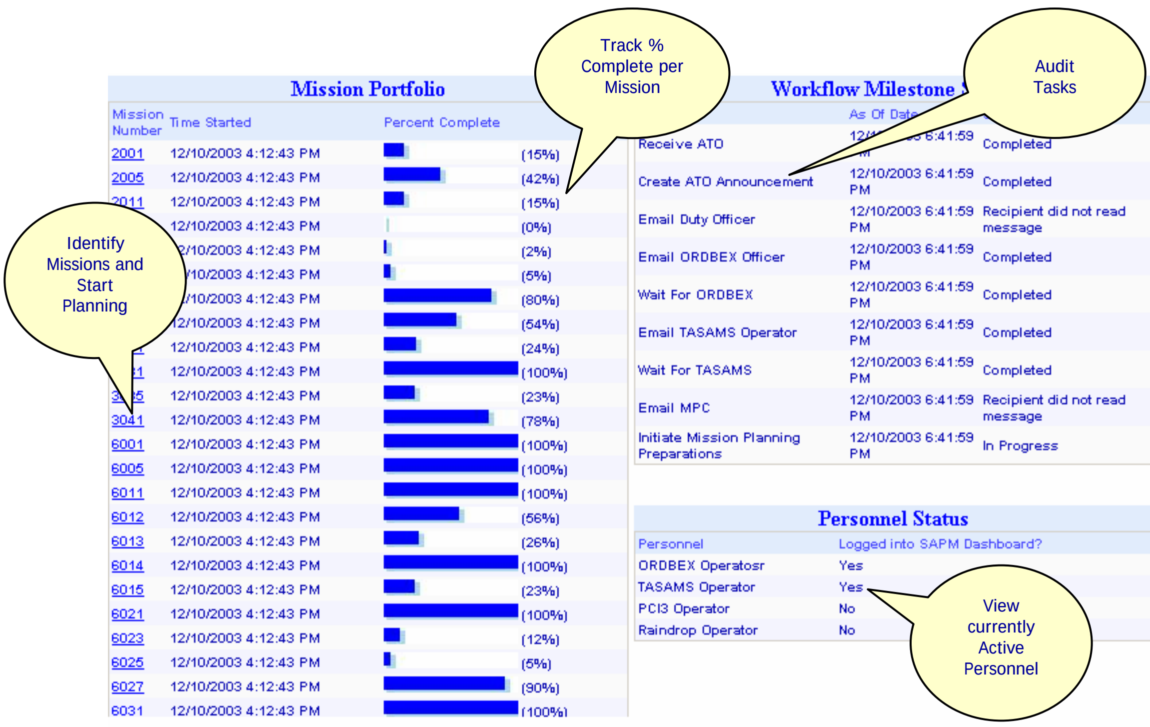

Figure 6. SAPM Presentation Framework

The SAPM workflow application required the introduction of a service, the Operational Data Store (ODS), to persist information common to many of the systems associated with the mission planning process. The ODS, designed to operate on a Microsoft Share Point server, stores the route information in an XML format. Other systems of record are capable of retrieving the CRD via a web service exposed by the CRD.

The unbounded system coupling offered by web services, and the flexible design tools and framework provided in the Biztalk environment resulted in the implementation of the alternate process execution models being primarily a process modeling task and less of an engineering system integrations task.

Web Services Overview

An important feature of web services is that the invoking application needs not to know anything about the language the remote application is constructed in nor the platform it is deployed on. Another desirable feature is that web services run over standard TCP/IP networks requiring no infrastructure changes. To support the existing network infrastructure and the net-centric capability of the existing systems, BizTalk uses web services to invoke remote services. The web service programming model allows one to construct highly scalable, distributed applications using XML based messaging to exchange data between different systems in a possibly heterogeneous environment.

Web services do have drawbacks. The verbose nature of XML and SOAP messaging imposes a network overhead that other methods such as CORBA or Remote Method Invocation (RMI) do not. As such, for network constrained environments a careful analysis should be performed on the effects that web services may have on the network. Security is an additional consideration.

The web service infrastructure consists of several components that enable applications to discover and consume services as illustrated in Figure 7. These components are described in the following paragraphs.

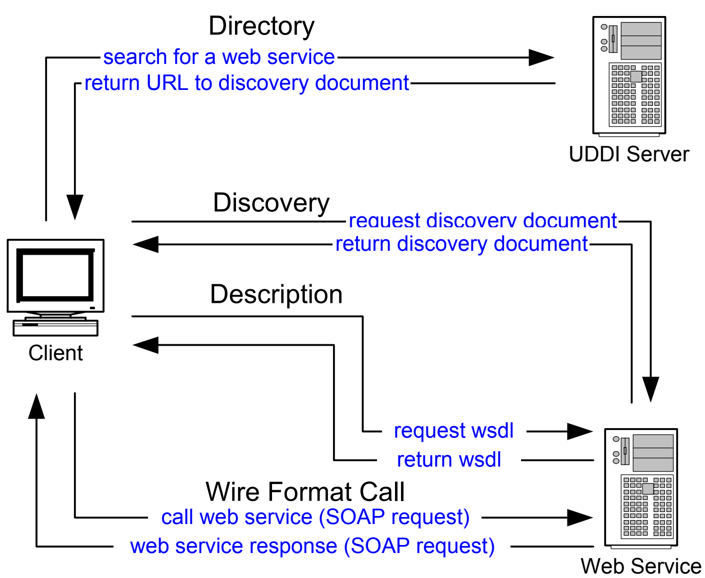

Figure 7. Web Service Invocation Infrastructure

Webs Service Directories

Web service directories provide a central location to publish information about web services. The UDDI specification defines the web service publishing guidelines. UDDI defines three types of information associated with a web service: business data, descriptive service information, and detailed service specifications.

Web Service Discovery

The web service discovery process involves locating the documents that detail the necessary specifications required to call a web service. The standard format for the specifications is the Web Service Description Language (WSDL).

Web Service Description

An individual web service may expose multiple operations. The web service description component provides the descriptions required to allow a user to determine what operation to call, the required parameters, and how to resolve the service. The web service descriptions are part of the WSDL file. Typically the WSDL associated with a web service is initially resolved during an application's design phase to ensure that the application's data model has the prerequisite elements to successfully call the web service. The actual service point, generally a URL, that is used to call the web service is typically resolved at runtime by querying a network naming service, the UDDI server. Once the service has been resolved, it is typically cached by the invoking machine.

Web services may also be discovered, bound and invoked at runtime. But similar to other protocols there is a performance penalty associated with such "late-binding" techniques.
The SAPM initiative utilized design time binding and run-time discovery to prove the performance associated with early-binding and the flexibility to relocating services offered by late time discovery and caching.

SOAP Package

Web services use protocols that can be understood by any system that is capable of supporting common Internet formats such as HTTP and HTTPS. The HTTP(S) GET and POST operations are the common methods for invoking web service operations. The operation calls are embedded in a SOAP package. The SOAP protocol allows the structured transfer of typed information between clients and servers over the inter/intranet. If a web service is being invoked by an HTTP GET or POST, operation the SOAP package is the body of the HTTP operation.

A SOAP package consists of four parts:

- SOAP Envelope: The mandatory SOAP package wrapper.
- SOAP Header: A section that defines encoding, capabilities and forwarding rules.
- SOAP Body: The SOAP body contains the necessary information and parameter to call a web service.
- SOAP Fault: The SOAP fault details error handling mechanisms.

The UDDI Server

The UDDI server is an integral part of the Windows Server 2003. The registry allows an organization to publish information about itself and the services it provides. The services are organized into a defined topology, or hierarchy of service and binding information. The UDDI services within the SAPM UDDI server have been organized according to the following UDDI.org specified taxonomy as shown in the Table below .

Table SAPM UDDI Taxonomy

UDDI.org Specification Term	SAPM representation
businessEntity	Name of the local wing (i.e. 99 WG COMPOSITE)
BusinessService	Name of the SAPM Systems (i.e. NAWS)
bindingTemplate	Service endpoint (i.e http://naws/naws.asmx)
tModel	A reference to the WSDL document for the service

Individual services involved in SAPM expose interfaces (web services) in unique format that may require translation before being consumed. BizTalk provides the ability to graphically transform data formats and map data elements between different models. This capability supports the concept of escalating the task of developing workflow automation solutions from an information technology task to an operational task (see Figure 8).

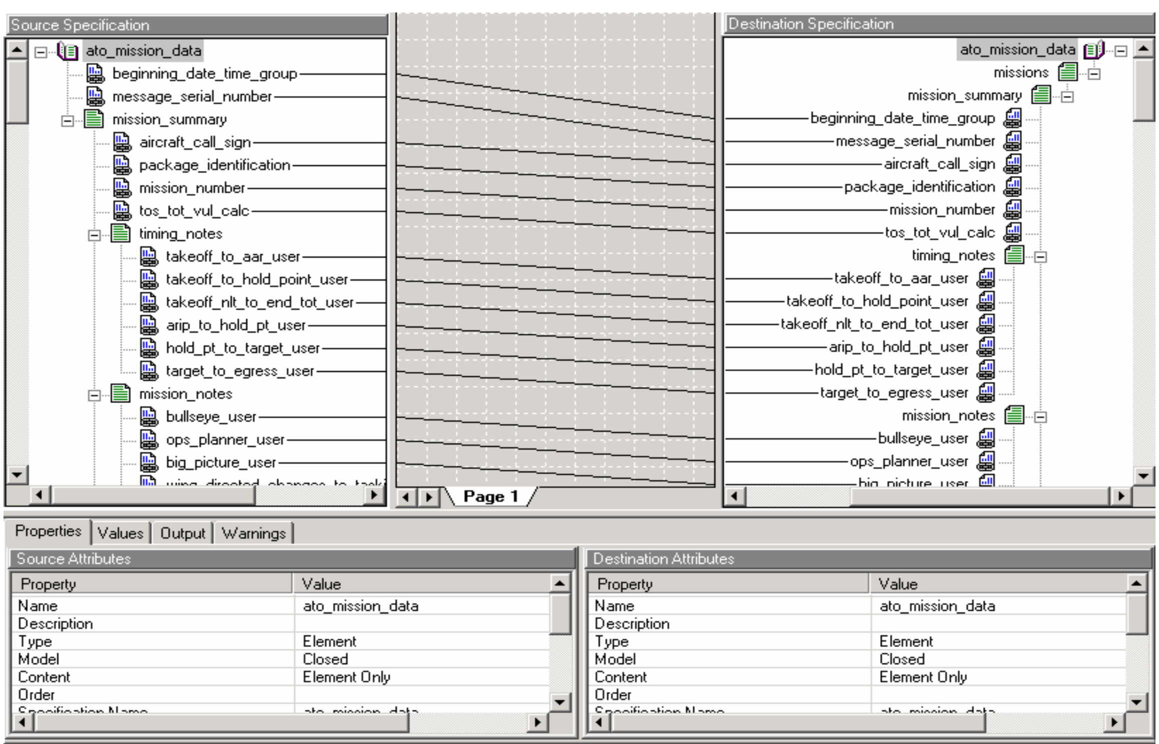

Figure 8. BizTalk Graphical Data Transformation and Mapping Utility

Webs Service Security

Early workflow projects with web service interfaces often overlooked security due to a lack of web service security standards. This risk is being mitigated by the adoption of the web services security model (WS-Security) by the Organization for the Advancement of Structured Information Standards (OASIS), a global consortium driving the adoption of standards, and the inclusion of WS-Security into most commercial workflow engines.

The WS-Security specification describes enhancements to the SOAP based web service applications. WS-Security provides a general purpose mechanism for associating generic security tokens with a SOAP message. The specification covers three primary areas: token propagation, message integrity, and message confidentiality. The specification does not provide a comprehensive security solution, but is intended to be used with application specific protocols and encryption techniques.

Similarly, the Security Assertion Markup Language (SAML) developed by the OASIS defines a framework for exchanging security information between entities. SAML defines a common XML framework for exchanging security assertions. SAML is different from other security systems due to its approach of expressing assertions between an asserting party and a relying party about a subject that other applications within a network can trust.

Neither SAML nor WS-Security provides a solution for managing information across security domains. Ongoing research and product development into XML guard technology that can address this issue is currently being conducted by leading vendors and government labs.

Result and Lessons Learned

SAPM Phase I was completed in 90 days of intensive effort. Along with the SAPM workflow operational process models, over 30 web services were developed. SAPM was successfully demonstrated to ESC Commander, Lieutenant General William Looney, USAF in May 2003 and to many other flag officers. SAPM was showcased at the 2003 AF C4ISR Summit in Danvers, Massachusetts, the 2003 Microsoft DoD Air Force Symposium in Redmond, Washington, and the 2004 AF Mission Planning Users Conference (MPUC) in Las Vegas, Nevada. Figure 9 depicts the amount of operational streamlining that SAPM has demonstrated.

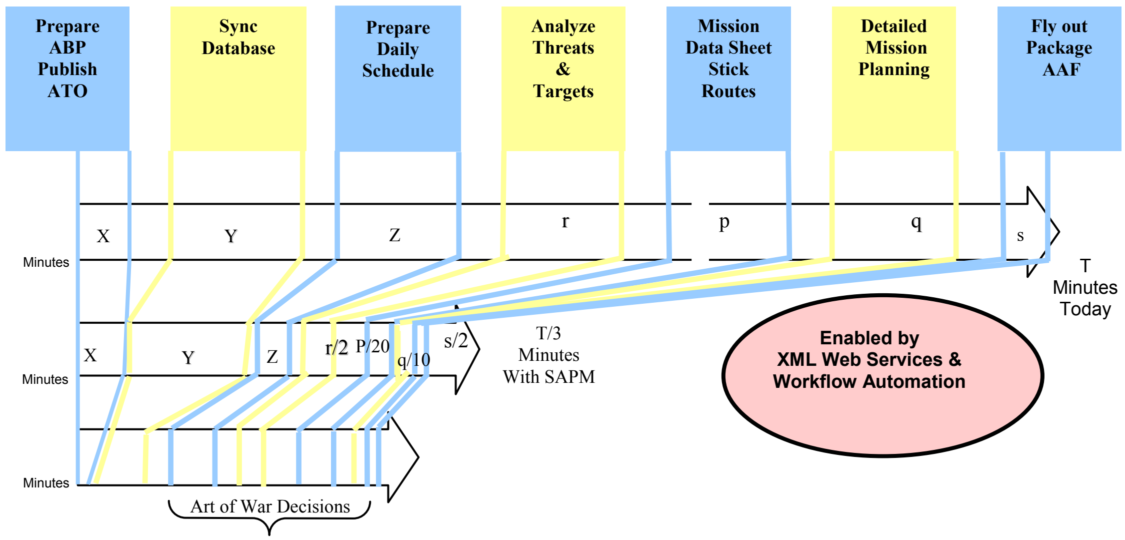

Figure 9. SAPM Process Streamlining

Some of the lessons learned are described as follows.

- SAPM has proven that enabling machine-to-machine communications using web services and workflow automation helps break down the barriers to rapid information exchange among AF C2 systems. Those technologies could drive us closer toward General John Jumper's, USAF Chief of Staff, vision of enterprise interoperability and the implementation of the DoD network-centric warfare.

- Phase I was a successful collaborative effort of the government and industry. It was accomplished through strong support from program offices and significant contributions from vendors and contractors. The scope for SAPM Phase II and beyond would be much bigger and will require continuous funding and commitment from participating programs.

- SAPM has demonstrated that loosely coupling via web services facilitates distributed development. Therefore, enterprise integration could be accomplished in heterogeneous environments (e.g., UNIX and .NET) as long as those web services are WSDL compliant and developed according to the tenets of the AF ESC Command and Control Enterprise Reference Architecture (C2ERA).

- Due to resource and time constraints, the Phase I development team was unable to fully define the SAPM architecture and requirements before development started. In fact, most of the requirements were evolving during development. It was manageable during the Phase I proof-of-concept, but it was not considered as a sufficiently rigorous engineering process. The prototype architecture and requirements should be defined prior to prototyping development in order to better address individual program's needs and expectations.

- The implementation of web services and workflow automation enabled SAPM to transmit mission tasking data and battle plan information in real time. Therefore, with SAPM, Wing commanders' briefings, mission checklists, and mission reports could be automatically generated within minutes of receiving the tasking order. This not only could minimize the presence of slow and error-prone data entry, but also could provide Wing commanders early visibility to the status of missions, logistics, and weapons allocations.

SAPM Phase II Enhancements

Some enhancements and new capabilities will be implemented in SAPM Phase II. For example, Phase II will allow simplified administration of UDDI. All web services will be maintained in the SAPM UDDI. Thus, web service consumers will retrieve web service information directly from this UDDI. Additionally UDDI settings will accommodate secondary providers in case of a primary failover. The SAPM ODS will be enhanced to a scalable solution in order to store metadata in a database. It will leverage the document storage features of Share Point server. Also ODS replication features will be designed into the Phase II ODS but not implemented in the BizTalk workflow model.

Bibliography

ESC/AC, "*Concept of Operations for SAPM"*, 13 August 2003, Hanscom AFB, MA

ESC/AC, "*SAPM Warfighter Rapid Acquisition Process (WRAP) FY04 Proposal"*, 13 August 2003, Hanscom AFB, MA,

ESC/AC, "SAPM Warfighter Rapid Acquisition Process (WRAP) FY04 Proposal", 13 August 2003, Hanscom AFB, MA,

William Nelson, Colonel USAF & Daniel Hebert, ESC/AC, *"SAPM Proof of Concept Briefing to Lieutenant General Looney",* 16 May 2003, Hanscom AFB,

Kathy Harding & Jean Trenary , *Developing XML Web Services and Server Components*, Microsoft Press 2003

Paul Kulchenko, James Snell, & Doug Tidwell, *Programming Web Services with SOAP,* O'Reilly 2002

Scott Seely, *Web Service Development Using XML*, 2002, Prentice Hall

"Workflow Design Patterns ", 20 February 2004 http://tmitwww.tm.tue.nl/research/patterns/patterns.htm

UDDI.org, March 12, 2004, http://www.uddi.org

The Butler Group, "*Business Process Management*", 8 March 2004
http://www.butlergroup.com/operations/BusinessProcessManagement,

OASIS, February 20, 2004
http://www.oasis-open.org/home/index.php