Interoperability Strategy

Concepts, Challenges, and Recommendations

Concept Level WHITE PAPER

Developed for the Federal Enterprise Architecture
Program Management Office (FEA-PMO)

Industry Advisory Council (IAC)
Enterprise Architecture SIG

March 2003

Disclaimer

Credits

This paper is the result of work coordinated under the Enterprise Architecture (EA) Shared Interest Group (SIG) of the Industry Advisory Council (IAC).

Primary Authors

John Dodd, CSC/IAC, jdodd@csc.com
Bruce T. Peat, eProcess Solutions/IAC, BPeat@eProcessSolutions.com
David R. Mayo, Everware/IAC, david@everware.com
Eliot Christian, U.S. Geological Survey, echristi@usgs.gov
David RR Webber, eProcess Solutions/IAC, DWebber@eProcessSolutions.com

Other Contributors
Mike Lubash, Defense Finance and Accounting Service - DFAS
Murali A. Iyengar, SAIC/IAC
Brien Lorenze, BearingPoint/IAC

Table of Contents

1 Introduction

The Federal government is the largest user of information technology (IT) in the world. It is estimated that over $50 billion of the Federal budget is devoted to IT annually. A substantial portion of this supports the maintenance of thousands of legacy systems, the vast majority of which were not designed to work together. Recent events have underscored the need we have, as a nation, to be able to consolidate information from disparate systems to support homeland security, criminal justice, and other missions. These missions often cross Federal agency boundaries and extend into State and local government areas. Whereas our systems and databases have been constructed as "silos," the requirements are along the lines of cross-jurisdictional processes and extended "value chains." One of the most urgent and important challenges currently facing governments is to get these systems to interoperate and share information.

1.1 Purpose

The purpose of this paper is to provide some background on the issues underlying the interoperability challenges, to shed some light on potential approaches to dealing with the problem, and to offer some specific recommendations, based on industry experience, that government at all levels can implement to rapidly address this challenge. The Industry Advisory Council (IAC) brings an industry perspective to the issues facing government and offers solutions that have succeeded in commercial settings that may be useful in addressing the issues facing government. These recommendations are "No Regret" proactive actions that our government should take to move forward. This paper represents a starting point, a basis for initiating a dialog on how to address the issues of interoperability and information sharing. Concepts and Context

At its most fundamental level, the concept of interoperability is simply about making things work together. This can be accomplished in a number ways. At one end of the interoperability spectrum is integration, which involves bringing things together into a whole. However, in most cases, interoperability is based on communication between two or more entities. As a result, most of the underpinnings for interoperability come from the field of communication. Successful communication relies on three principles:

Common Syntax (the structure of the message)
Common mechanism, Common Semantics (the "meaning" of something).

XML is becoming a common standard syntax, alternative to ASN.1 and other syntaxes. The common mechanism for communicating between systems has become TCP, supporting higher level

Doctrine for Interoperability

Business First
Shifting power to the users; customer and business experts, e.g. self-service
Provide traceability from business vision to implementation (and status)
Managing information assets to ensure: visibility, accessibility, interoperability, and understandability through metadata
Semantic-driven; technology neutral context supported by classifications, ontology and patterns for semantic alignment
Moving the semantics from applications to the infrastructure layer
Objective: not standard language - but instead standard reusable mechanisms to better negotiate differences
Capture rationale for pragmatic interoperability; Templates and models to define 'what' not 'how';
Its not just technology; people are key asset

Multi-Faceted Architecture
Function-centric; not system or entity
Choice: Web (human), data, process, services
Modular and layered to address complexity; leverage open initiatives such as XML
Service-oriented; loosely coupled interfaces
Wrap legacy systems with services
Provide structure for business patterns
Defer physicalization as long as possible

communication mechanisms such as HTTP over TCP/IP. But, having a common syntax and common mechanism is not enough. XML is not enough. Interoperability requires that the systems have a common definition of what is to be shared or communicated. We need an infrastructure to support semantic alignment. In *Appendix E* we show a Homeland Security example which leverages the work at OASIS; Organization for the Advancement of Structured Information Standards, on Content Assembly Mechanism (CAM) which uses XML templating to allow users to construct information exchanges

To demonstrate the further need for semantic alignment support, given the example, the term, "salary" has to mean the same thing to both systems, such as: gross annual compensation including bonuses and other financially quantifiable benefits, expressed in 2003 US dollars. Even if two systems use the same syntax or structure for the message (such as US dollars to the cents level), if "salary" does not mean the same thing to both systems, they are not interoperable.

Interoperability of systems only requires a common basis for those elements that are, in fact, shared. Typically, not all of the information managed by two systems is shared. Therefore, interoperability requires identifying the shared elements. Not even all elements that have common (or close) definitions need to be shared. Interoperability involves common semantics and syntax only for those elements that must be combined, compared or aggregated. In the example, if the salaries from the two systems are not to be compared, added, etc., then there is no need to reconcile the definitions. This point is important, because reconciliation is difficult and time-consuming, especially after the fact (post-coordination). It is well known in the IT world that correcting defects in systems is much easier at the analysis stage than at the construction or deployment stage. The same concept applies for reconciling data element definitions. Thus, interoperability should be architected-in or reconciled during the architecture stage (pre-coordination). For Federal agencies this means at the stage of creating the Enterprise Architecture (EA), and shifting our view in managing our information assets by including a complementary model – a model for agility as shown in *Appendix B*.

While this description focuses on data sharing, interoperability covers process sharing as well. Interest in process interoperability has increased in recent years as the standards for "web services" have become more mature. Web services allow different applications to access common processes, or services, for provide specific functionality. Legacy applications can become reusable components through encapsulation, such as Web services or proxy servers. It is relatively easy, inexpensive, and low risk to encapsulate rather than the alternatives. Web services can apply to legacy batch processing and message-oriented online applications. Therefore, if the legacy applications are still fulfilling their business purpose, encapsulation may be the best strategy, particularly if you can also resolve any other structural issues during the implementation. In general, the concepts that apply to data and information sharing also apply to services interoperability.

1.2 Agreements are Key for Interoperability

In many e-Government initiatives, Government seeks to be as inclusive as possible, spanning information systems managed at all levels of Government (Local, State, Tribal and Federal), as well as academic, commercial, and other non-Government organizations. This goal demands a high levelof interoperability, i.e., differences among systems must not pose a barrier to tasks that span those systems. Therefore, e-Government initiative participants must agree on ways to accommodate system-to-system differences and so support a shared information architecture.

Interoperability in the shared information architecture should be broad and sustainable at a strategic level. Fewer agreements accommodating most systems are preferred over many agreements accommodating few participants each. Interoperability agreements also should entail minimum impact on affected systems other than their interfaces with the shared architecture. Active participants that specialize in particular interfaces may have a lead role in building consensus on interoperability agreements (e.g., a certain government agency may have a leadership mandate to build consensus on a particular set of interfaces).

As a practical matter, interoperability agreements must be driven by specific needs as they are identified at the interfaces among active participants. Wherever possible, interoperability agreements must be based on non-proprietary standards and profiles should be specified when standards are not sufficiently specific. All interface implementations should be specified in a platform independent manner and verified through interoperability testing and public demonstrations.

2 Alignment Challenges

Without a firm grasp on the issues, how can an effective set of solutions be put in place? A miss on understanding the real challenges, addressing the secondary instead of the primary challenges can put in jeopardy the success of any e-Government project. The solutions, technical and non-technical, themselves will be difficult enough to be implemented. In addition, everyone involved will appreciate the discipline required to address the root of the problem, and not once again superficially promote stale prepackaged answer sets. Government agencies will need to address a wide variety of problems for a number of reasons, including incompatib le existing information systems, legal and privacy restrictions on the sharing of information, and organizational barriers between agencies, to name a few. Applying resources consistent with these principles should eliminate much of today's stovepipe architecture; improve both the time of response and the quality of decision support required for Homeland security. Horizontal information sharing equates to greater security.

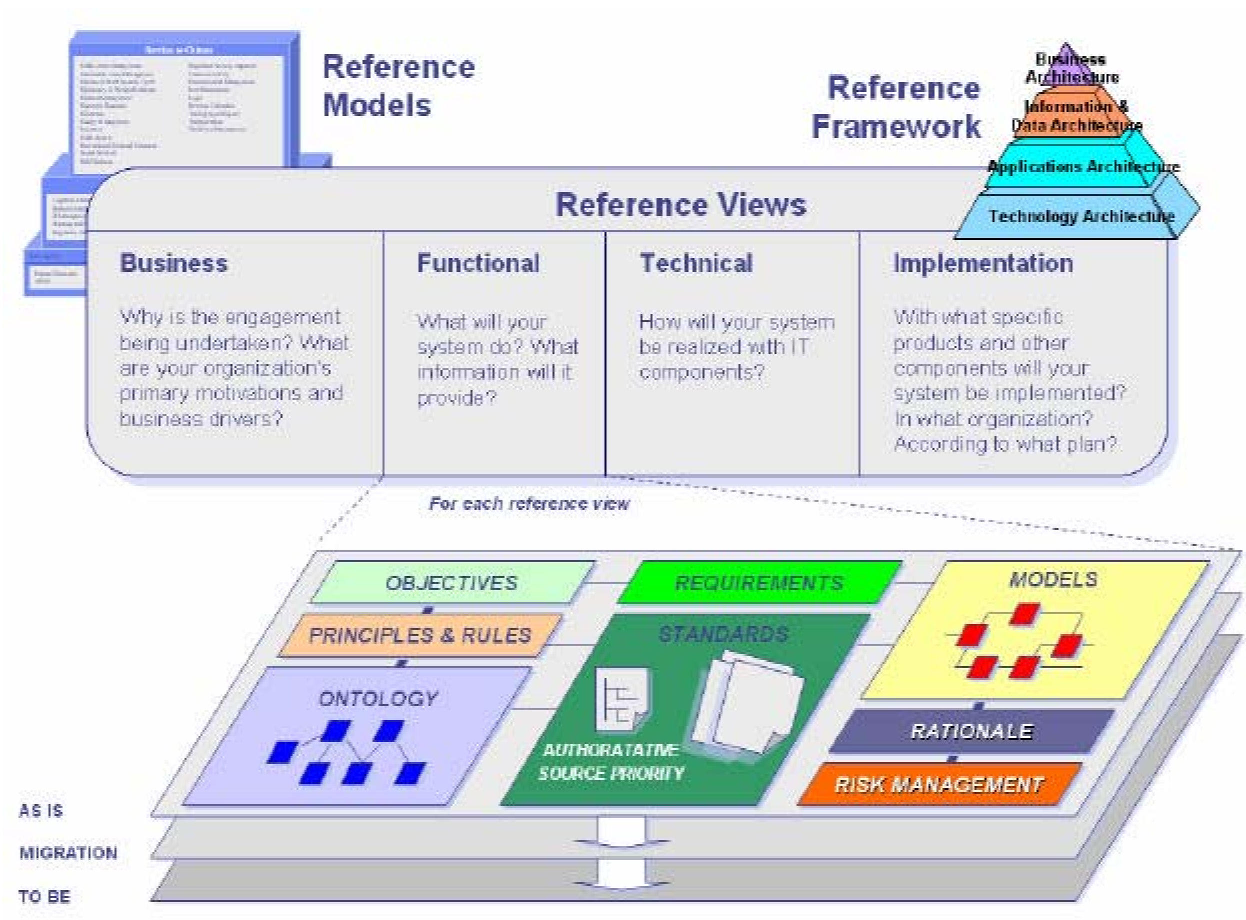

Figure 1: Enterprise Architecture for Interoperability

2.1 Specific Challenges of Business Line Alignment

Crossing government organizational boundaries to produce Enterprise Business Line Architectures is very difficult. This is especially true without a "Boundary Crossing Roadmap" and examples of success to follow. It is especially difficult when everyone is talking in a different language and using a different Enterprise Architecture process. Enterprise Architectures throughout the government will need to be aligned with the government-wide Business Reference Model but also to align and create a complete horizontal information chain exchanging key global information that must be shared and maintained in a consistent manner. But only architectures with the proper focus provide interoperability. This cross-government information sharing and consistency problem is critical to delivery of timely information for Homeland Security or the improved delivery of service to the citizens. This will require a consistent approach to modeling and representing information as presented in a companion white paper on "Data and Information Viewpoints :Federated Information Model". Along with the process, security and other aspects discussed in related papers while this paper discusses the technical aspects of integration and interoperability. Our focus is on the interoperability along business lines. We recognize that this will force extensive cooperation among members of agencies that are not within the same Department.

Depicted in Figure A is the overall management approach for migrating from a systems-centric architecture to one aligned to e-Government Business Lines based on functionality. The best practices and technologies are the focus of the rest of this paper.

2.2 General Challenges

This section identifies some of the challenges facing any effort to achieve interoperability and information sharing. In addition, *Appendix A* details the scope of the challenge and the 'Top 10 List of Integration Inhibitors" that our government needs address for a holistic enterprise solution. Any effort that is to succeed must address and overcome these issues.

Organizational – Achieving consensus on meaning is the most difficult challenge. Agreement on semantics and syntax is difficult to achieve due a variety of factors including:

Perceived loss of control (resistance to change); reluctant to give up one's view of the world (process and data)
Lack of incentives to cooperate: what's in it for me? (WIFM)
Costs – lack of program budget for activities outside the program

Architectural – The Enterprise Architectures of the agencies are not aligned and a process for alignment has not been defined and disseminated to the agencies. It is not our contention that a "super-EA" needs to be created; rather, the interoperability requirements need to be layered over the EAs based on the extended value chain concept described above. In addition, the lack of a forum and governance structure for developing consensus is a hurdle that must be overcome for progress to be made.

Technical – The infrastructure to support interoperability at the service/component and data level is not in place. The standards for the technical platform for interoperability have not been adopted. The number of legacy systems in place with disparate definitions for activities and data elements is a significant challenge. A mechanism for technical reconciliation must be developed because it is simply not possible to replace all systems that must be involved in information sharing.

3 Recommendations

The following recommendations are "No Regret" proactive actions that will provide our government great strides in addressing the interoperability challenges. These recommendations are targeted to OMB; Office of Electronic Government, to be used as suggested guidelines and directives to Agencies. Best practice studies have shown that cultural change is most effectively achieved when there is sustained leadership commitment and the institutionalization of new processes. Establishment of the Office of Electronic Government can promote successful practices across Agencies and communities of interests (CoI).

3.1 Business-Centric Methodology

Adoption of Business-Centric Methodology for agility & interoperability that …

Addresses the root cause rather than just symptoms of our integration problems by providing *semantic* and *pragmatic interoperability*

Business-centric; shifting power to the business experts; managing Enterprise artifacts and governance through Communities of Interests (CoI)

Provides visibility, accessibility, understandability, using open *declarative mechanisms* that allow for *mass customization* of diverse vocabularies and models within *heterogeneous environments*

Insulates business from the high rate of change of technology by dividing the problem into multiple levels and applying constraints properly to *reduce complexity* and *promote reuse*

Provides for Enterprise agility and prepares the Enterprise for new opportunities in doing business by providing the required *transformation* underpinnings in adopting the Federal Enterprise Architecture.

An overview of the Business-Centric Methodology is presented in Appendix C.

3.2 Move to Standard Mechanisms

The Government should continue to promote the following principles:

- Avoid non-standard data syntaxes - There are myriad ways to represent data and each has its peculiar strengths and weaknesses. Especially troublesome from an interoperability perspective is the fact that conversion between different data representations can degrade the data. It is therefore very important that participants avoid non-standard data representations and agree on a small number of robust data representation syntaxes for data that traverses shared interfaces. In networked applications today, there are two robust data syntax mechanisms that should be agreed upon in e-Government initiatives: the international standard known as Abstract Syntax Notation (ASN.1) and the emergent industry standard Extensible Markup Language (XML).

- Register the semantics of shared data elements - Interfaces among participant systems in e-Government initiatives are typically comprised of a large number of data elements. In this situation, it is a non-trivial task for participants to gain a common understanding of the meaning of data elements defined at the interface. (The syntax information available in an XML schema or an ASN.1 definition does not fully address the requirements of semantic interoperability.) The agreed international standard for representing such understandings in a commonly accessible registry is ISO 11179 (formally designated ISO/IEC 11179, Information Technology--Metadata

Registries). This standard supports registration of data using virtually any syntax, and may also provide a basis for interoperability among industry-led registry initiatives.

- Document service interfaces in a standard way - Interoperability within e-Government initiatives is fundamentally dependent on specifying common interfaces among disparate information systems. In addition to specifying the syntax and semantics of the data elements defined at the interface, it is necessary to fully describe how the systems interact at the interface. One information system mechanism for describing such interaction is an "interface definition language" (IDL). An elaborate example is the IDL for CORBA (Common Object Request Broker Architecture). Industry-led movements have recently shifted toward WSDL (Web Services Definition Language) or ebXML specifications to describe service interfaces. Each of these service interface description mechanisms has an associated mechanism to automate discovery and access of service interfaces. Unified Modeling Language (UML) is also commonly used to document the interactions occurring at a service interface. At present, participants in e-Government initiatives should agree to use any one of these standard mechanisms to describe service interfaces, and to convert to a single standard when appropriate.

For example this would include implementation of standard interfaces for geospatial data. In e-Government initiatives especially, data and information resources are often referenced to a place. Such "geospatial data" may be viewed in the form of a map but the underlying digital data is usefully applied in many other forms as well. Interfaces to discover and use these data and services have been standardized, ranging from "yellow pages" and "product catalogs" down to "technical manuals". International standards supporting discovery of and access to geospatial data and services are agreed upon through the various Spatial Data Infrastructure initiatives. These include the discovery interface standard referenced above (ISO 23950), as well as a range of international standards covering documentation and representation (ISO 19115) and place codes (e.g., ISO 3166). In addition, participants may support important emergent standards such as the range of geospatial interfaces being developed through the OpenGIS Consortium.

> Information Discovery Standard
>
> The information discovery service provides a common interface to the network and allows very diverse information management to be maintained behind the interface
> The information discovery standard ISO 23950 is being used to achieve interoperability among online information services, directories, library catalogs, data catalogs, atlases, and many other kinds of information resources
> A profile of the common standard service for information discovery, ISO 23950, is specified in FIPS 192-1, as required by OMB policy and public law, 44 USC 3511, "Establishment of Government Information Locator Service"

- Implement the standard interface for information discovery - A central thrust of e-Government initiatives is to enable the discovery of and access to a wide range of information resources and services. Support of a common service interface for information discovery is therefore an important information architecture principle that should be agreed upon. (The term "information discovery" refers to the process of finding relevant data and information resources without prior knowledge of where those resources may reside, how they are organized, or how they are usually accessed.) The agreed service interface implements the international standard (ISO 23950 Protocol for Information Search and Retrieval) that is well supported and broadly adapted to most information search and retrieval system interfaces. This common interface applies to discovery of resources in the form of traditional library, museum, and archives holdings as well as digital resources distributed across global networks. This service supports XML and ASN.1 data syntaxes; data element semantics are registered in the ISO Basic Semantics Register; and, service interfaces have been provisionally defined using CORBA IDL and WSDL and have been published via UDDI (Universal Description, Discovery, and Integration).

- In addition, NIST needs to get back into the standards business publishing federal interoperability standard mechanisms (FIPS) to support government via conformance test suites and their validations facilitate meetings.

3.3 Provide Infrastructure for Visibility

Registry services, such as OASIS ebXML efforts, contain information that describes the structure, format, and definitions of data. Typically, a registry is a software application that uses a database to store and search data and document formats, definitions of data, and relationships among data. Concepts with definitions, authoritative source identified by each community. Users and applications can discover the existence of data assets through registries, and other search services. All data are provided or "made visible" by providing metadata, which describes the asset. A critical aspect of the service set is the collaboration for aligning vocabularies around concepts and providing the navigation for users to understand comparisons and contrasts.

3.4 COI – Communities of Interest

Communities of Interest (COI) are collaborative groups of users with shared goals, interests, missions or business processes, operating under agreed upon terms. They are of two general types: [1] institutional, and [2] expedient. Institutional COIs, whether functional or cross-functional, tend to be continuing entities with responsibilities for on-going operations. They also lend support to contingency and crisis operations. Expedient COIs are more transitory and ad hoc, focusing on contingency and crisis operations. The infrastructure for collaboration, including the registry(s) assist in discovering personnel, subject matter experts, etc. The COIs also provide for notice of intents of initiatives getting word out early, and for the collection of feedback.

Federal Geographic
Data Committee (FGDC)

The FGDC community of interest provides a focal point for geospatial data and mapping, across all levels of government and including commercial and other non-government parties FGDC and partners such as the OpenGIS Consortium help to create and test international standards for interoperability, including common syntax, semantics, and services Geospatial systems have now converged on a high level of interoperability, from a global clearinghouse of data products and services down to common mechanisms for rendering maps and interchanging sets of geospatial features

3.5 Develop centers of excellence for interoperability mechanisms

Best practices show that new operating practices are assimilated more quickly when repeatedly, consistently promoted. Agencies demonstrating distinction can be nominated to lead and provide hands-on training for other government entities as 'Centers of Excellence'. Industry leaders likewise can be brought in to the centers to share their experiences as best practices are developed in partnership.

3.6 Other Areas of Opportunity

The following are other areas of opportunity that may offer large paybacks to the Government:

Develop and scope FirstGov as a model for interoperability

Identify incentives for collaboration and providing information
Specifications such as FIPS must dictate interoperability

4 Conclusion

In short, much of the e-Government movement is the evolution from static, undocumented, rigid stovepipe systems to dynamic metadata-driven and navigated agile business lines comprised of reusable components residing in a Service-Oriented Architecture (SOA). The SOA allows the redeploying of legacy applications as XML-encapsulated, trusted components and solutions with native XML logic providing the encapsulation and componentization.

The move to e-Government has improved on all government levels; Federal, State and Local. Citizens will increase their usage of online interaction with the government inline with IT investment. This investment, particularly in the areas of interoperability will result in significant taxpayer savings despite the challenge of changing work practices and political wrangling.

This document has outlined a few high return-on-investment recommendations that will yield our government positive results in solving the problems with interoperability today. Though few in number, if properly applied gains in information exchange between agencies should provide immediate demonstrable outcomes. If the U.S. government is to see e-Government become a reality, it must work with industry to ensure that this roadmap can be followed in a cost effective and expedited manner. Integration of legacy systems is a significant e-Government challenge.

Appendix A: The Interoperability Challenge

The following are the 'Top 10 List of Integration Inhibitors" that our government needs address for a holistic enterprise solution:

1. Semantics, Semantics, and Semantics
4. Frameworks are complex (and conceptual)
5. Failure for business managers to 'take back' the steering wheel
6. One size really doesn't fit all
7. Information is power
8. Brain Drain Paralysis
9. Funding for integration infrastructure
10. Culture

The inhibitors are detailed in the following:

Semantics, Semantics, and Semantics are the top three challenges for interoperability. Interoperability or integration efforts are about making information from one system syntactically and semantically accessible to another system. Syntax problems involve format and structure. An example is converting the representation of data from numeric to a character string. These conversions are well known and the problems documented. Many standard data sources, such as databases and applications can export XML for data transformation using code-free mapping tools. The accessibility of the information, or transport problem has been reduced to routine engineering tasks due to widespread investment in messaging infrastructures. Semantics relate to the understanding and integrity of the information. To put even greater emphasis on the challenge, the Gartner Group stated, *"Only 5% of the interface is a function of the middleware choice. The remaining 95% is a function of application semantics."*

On a positive note, the semantics of the government is fairly stable, and provides a foundation which to build – and assist in interpreting and aligning exchanges through ontology-based mechanisms.

Frameworks are complex and many times provides conceptual differences to working approaches; e.g. understanding and relying on classes in an object-oriented system. In addition, to the adoption hurdle problem, at times frameworks are duplicative and contradicting with multiple levels.

Failure for business managers to 'take back' the steering wheel– and are not eager to accept responsibility for even the 'what' objectives much less than the 'how' details. Due to tool immaturity integration development has required technical know how which excluded the business practitioners. Today top-down techniques have exhibited impedance mismatch with current programmer's tools (bottom-up) – with no automated solution that addresses development from business goals to the physical implementation well.

One size doesn't fit all - Understanding the critical difference between (1) decontextualization of data 'Standards' and (2) 'Conceptual-adaptive' alignment. 'Standardized data' provides for inflexibility which leads to a plethora of standards – creating the "Tower of Babel". Where as adopting a minimalist methodology built upon shared business concepts is simpler, doable, without expensive overhead which "Tower of Babel" syndrome brings to the enterprise. Experience tells us that (1) one-size architectures don't work, (2) one-size process models don't work, (3) one-size data model doesn't work, and (4) one-size transaction 'standards' don't work.

Information is power - thus interoperability requirements become skewed and outputting information becomes the driver, not a template driven exchange from the receiver's input. In typical situations, the organization receiving the information is just plain glad to obtain it, and takes is in any form possible, dealing with the integration issues. The better model certainly would be where the receiver drives the exchange and the exchange is based on aligned concepts.

Brain Drain Paralysis - Without knowledge retention, it is very difficult to determine impact of any effort to modernize – in some cases, there does not exist a baseline. For successful e-Government, the ability to perform impact analysis is one of the prime challenges. Adding new information or making changes to database structures can have multiple effects. One change can ripple across an entire enterprise. If data values are calculated from one another, based on one another, tied to one another — evaluating the effects of change can get very complicated very fast. Efforts on Y2K have given visibility into systems, and keen insight on the scope of the problem and provided government with a lesson learned, but probably will too be forgotten.

Funding for integration infrastructure - Funding and goals are to business lines and IT with very few independent 'integration' tools/team initiatives – interoperability though a prime challenge for the enterprise isn't funded as such. Acquiring integration infrastructure capability is seldom funded properly as their success outcomes are intangible and difficult to measure. Ironically, these integration projects typically are funded through application projects via business lines or IT departments, of which integration between these two groups which typically their lack of communication is the source much of today's problems. Should our federal government appoint an 'Interoperability Facilitator' as well as an 'e-Gov Director'?

Culture – human survival instincts for positioning in lieu of collaboration leads to anarchy and balkanization. In fact, outcomes typically are not measured on the whole; success metrics need to be viewed across traditional boundaries, with business goals and responsibilities aligned and traceable from the 'out' to 'in'. The human element must be kept in mind with any proposed system. Report cards need to bring back the category of "work well with others" and rewarded accordingly. Sometimes just getting the right people in the room does wonders for interoperability, trust and sharing. Interoperability will not be achieved if real problems are not confronted, we have learned interoperability starts with people first. Keeping this in mind, e-Government systems need to do whatever is technically possible to [1] reduce the politics of knowledge and its influence of power, [2] provide incentives to share, [3] provide collaboration tools with trust mechanisms, and [4] functions to share semantics of the business artifacts. Boundaries are not always physical stovepipe systems, but apply to division of groups – departments, agencies, and branches of government. Without a roadmap, the business users (goals) become disenfranchised from the technical management of the business itself, an intolerable effect that reduces business agility. Integration must include collaborative mechanisms for business users to and technical personnel to share and thus make a difference in the planning and management of e-Government projects. Any e-Government initiative must take into account the human element if it is to be successful.

Scope of the Challenge

The government has literally tens of thousands of systems each with:

- Different User Interfaces
- Different Models (business, activity, data)
- Different Business Activities, Events & Time bases
- Different Resolutions of Concern
- Different Business Rules (policy, location, laws), etc.

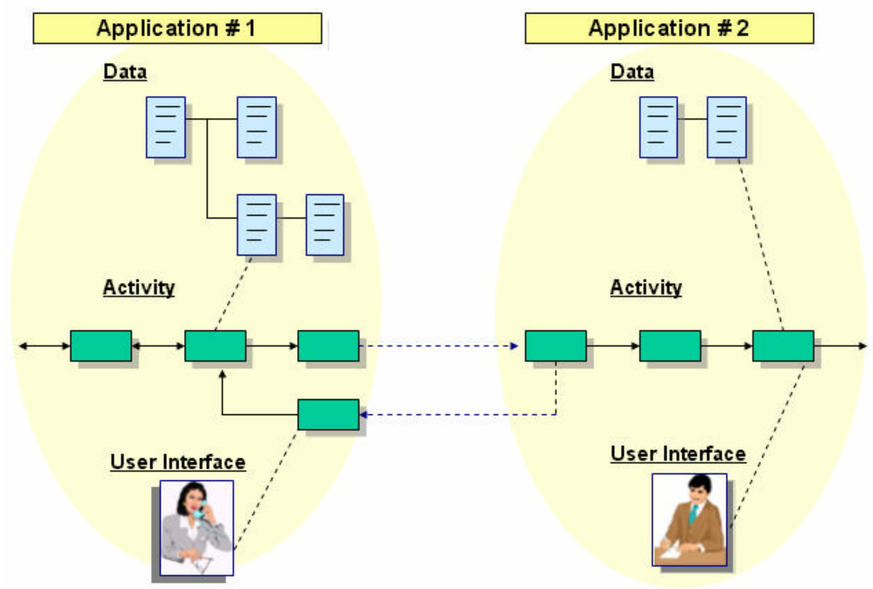

Figure A-1: Application Interoperability through Data, Activity, and User Interfaces

In many cases the systems are difficult, if not impossible, to change. For a scope of Federal and State implementations billions of decisions were made over time in narrow context and without regard to an Enterprise Architecture. Key to taking control is to understand the magnitude of the situation, correct over time and provide compressive integration strategy that includes 'target' reference points, milestones and metrics. While it is important to note the massive challenge, it is also important to note that the solution will come about through divisibility, in a phased approach, exposing differences and highlighting common attributes and semantics. Later Figure B will be uses to illustrate divisibility to leverage portioning and loose-coupling as one principle of the e-Government roadmap.

Appendix B: Paradigm Shift for Agility

Interoperability needs to be addressed on multiple layers, and at times requires us to view the problem differently. Architects design by adding constraints to the blueprint as requirements are gathered, these limits applied correctly define a process, application or building to meet the customer's needs. 'Modularity' has proven to be a key factor in providing reuse and encapsulating complexity.

In particular the Open System Interconnection (OSI) model has proven to be extremely successful to depict layering of communications among computers from different vendors. The OSI, developed International Organization for Standardization (ISO), [http://www.iso.org] addressed very difficult problems of different data formats and data exchange protocols. Granted the OSI model has taken us far down the road, particularly for the transmission stream over the physical link, transfer of data, switching technologies used to connect systems, transparent transfer of data between end points, and sessions; but leaves open the lexical alignment required for semantic exchange in the application layer. Today, this encapsulation strategy has evolved and incorporated into advanced architectures such as Object Management Group's (OMG) Model Driven Architecture (MDA). [http://www.omg.org] The fact is that it is rare to find an architecture that diverts much from the 1994 OSI general model.

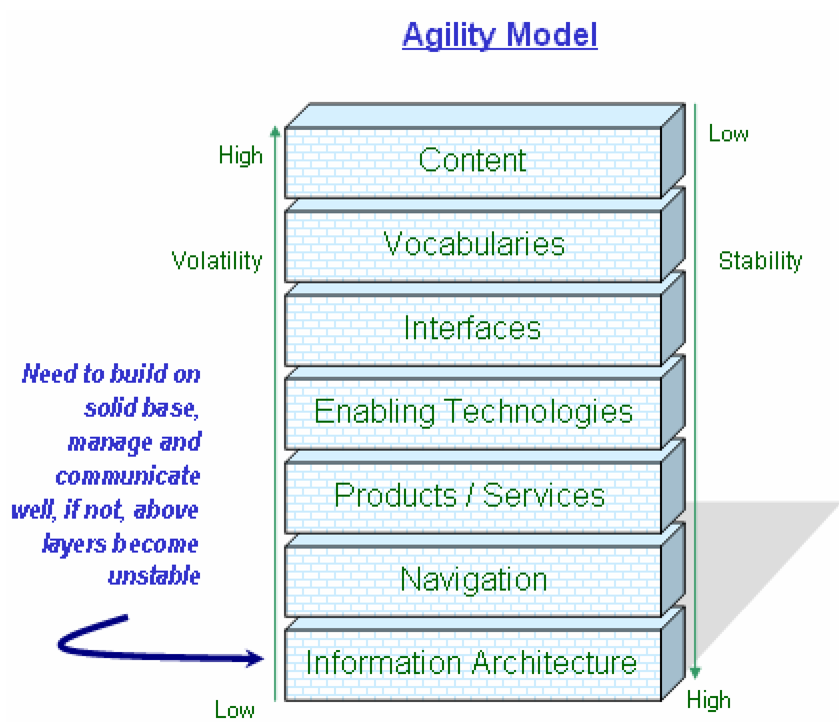

Figure B-1: Agility Model

Its 2003, and we are well positioned to address the challenge of semantic exchange. But to do this we need to adopt a different view, a complementary view. The new view needs to address agility in the Enterprise, understanding what components are stable and what are volatile. From a strong base, our Enterprise can be agile to provide business with "choices". Interoperability is all about choice and meshing or aligning choices at various layers. So what does this new model look like?

At first glance, it appears that the world has been turned upside down. But closer inspection reveals more than a connectivity diagram. This complementary model provides for a semantic base in the form of an information architecture, but declares vocabularies to be precarious, even more fluid than our interfaces themselves!

This 'Agility' model and the idea of 'choice' are the underpinning of the Business-Centric Methodology (BCM) described in *Appendix C*.

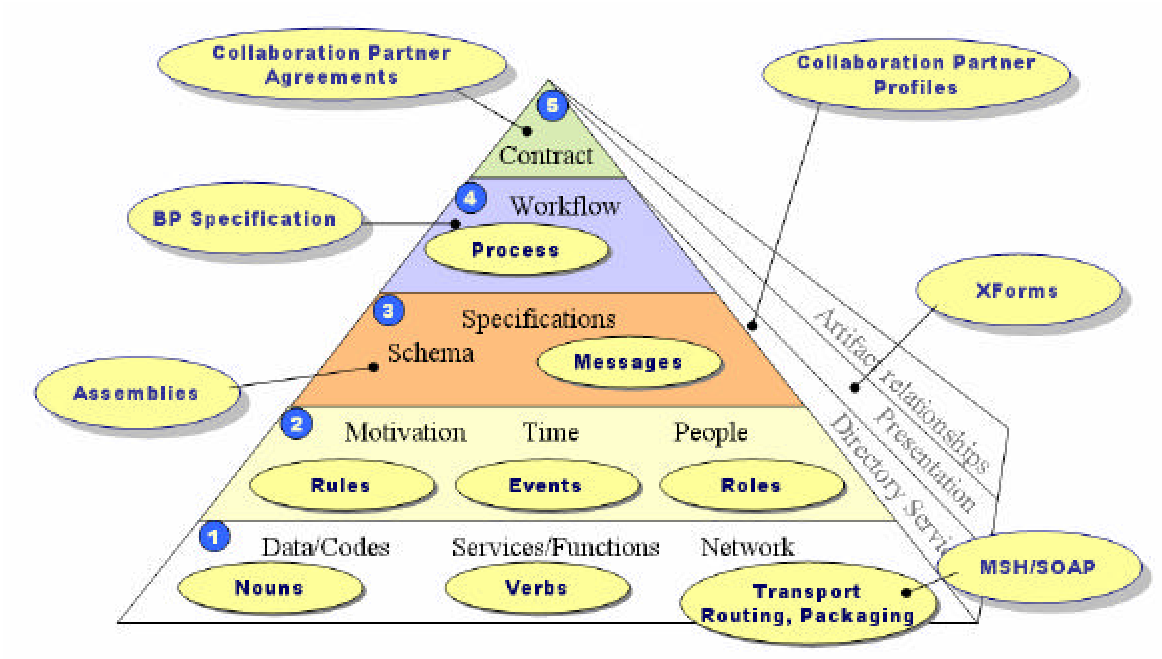

Figure B-2: Lubash Pyramid: An Information Architecture for Agility and Interoperability

Appendix C: Business-Centric Methodology

To achieve the results defined in the 'Doctrine for Interoperability' and address the issues discussed in *Appendix B;* Paradigm Shift for Agility, an information architecture and methodology is provided in the Business-Centric Methodology (BCM). The BCM prescribes a protocol for agility and interoperability for aligning disparate systems and Enterprises. For more information on the BCM reference http://www.DFAS.info, where the methodology is a candidate for implementing the architecture for DoD's Financial Management Modernization.

The Lubash Pyramid, Figure C-1 depicts the required artifacts that an organization considers should be registered and managed for interoperability and agility. The pyramid is an information architecture developed at the US Defense Finance and Accounting Service (DoD-DFAS) and highlights those critical items required for business integration either within a community of interest or Enterprise. Any information valued as a business asset should be controlled, made visible and shared with partners for integration as logically shown in Figure C-2.

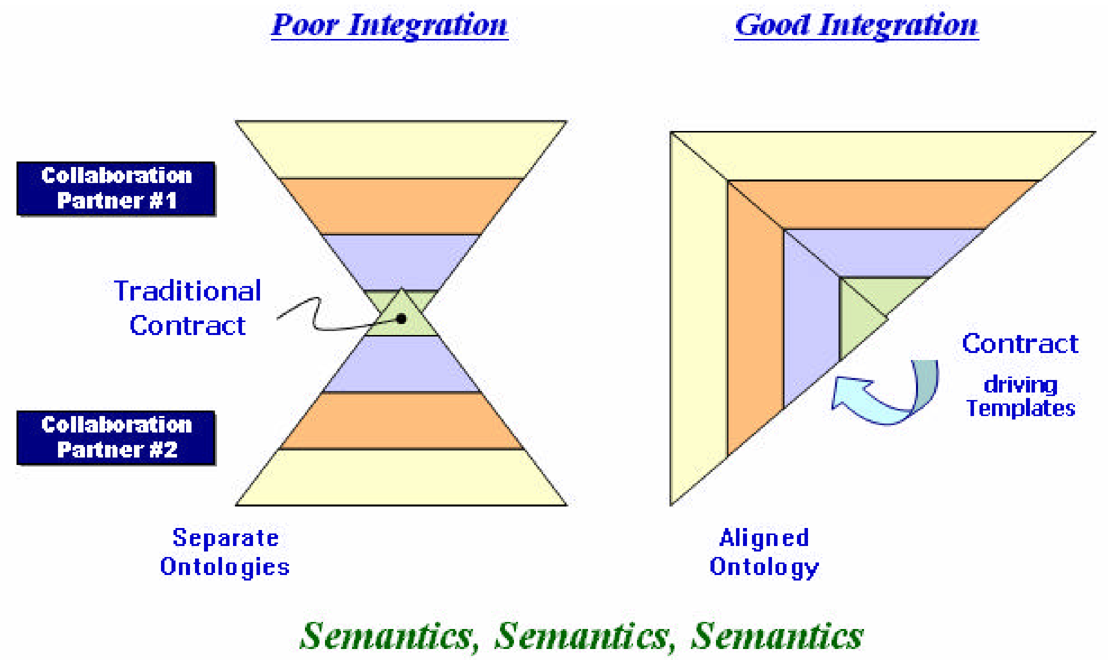

Figure C-1: Discovery and Alignment for Interoperability

The Pyramid provides a metadata management view of the Zachman framework (http://www.zifa.com). Those familiar with the Zachman framework should recognize the bottom layers (Inputs, Outputs) and Controls covers the six verticals of the framework. These layers abstract to WHAT, HOW, WHERE, and WHO, WHEN, WHY. The Lubash Pyramid builds on the framework and specifically identifies Specifications, Workflow, Contract, Presentation, Relationships, and Directory Services. The highlighting of these components/layers makes a distinction between requirements of interoperability of information and integration. Likewise, this model highlights the requirements placed on the Registry/Repository's faceted classification mechanisms to handle the permutations and relationships required for taking full advantage of the power of the registry and what the registry can bring to the value chain as discussed in *Appendix D;* Advantages Gained By Moving to a Registry-based Service-Oriented Architecture (SOA).

Business-Centric Methodology focuses on increasing best value within an eBusiness environment to reduce development time, integration resource requirements and maintenance costs through reuse and coordination of efforts. The BCM's advantage arises from its simplicity. By adopting and following an intuitive approach for [1] unconstrained conceptual alignment, [2] authoritative source collaboration, [3] layering of business constraints and constructs, and [4] the capture of rationale through templates one *gains pragmatic interoperability* as well as *semantic interoperability*. Stakeholders and can incorporate [5] UIDs either during development or align later to exchange precise communication to meet their business objectives.

The journey begins with establishing and outlining your organization's strategy and tactics for how to achieve exact communications among your stakeholders. The task is expanded to identify and manage your information assets, their associated metadata, context, ontology and design rationale with common template-based mechanisms. These technology-neutral artifacts become the building blocks for assembling reusable components that increase productivity and enable the enterprise to become more agile. The methodology is a solution focused on aligning the semantics of the business through open mechanisms, such as eXtensible Markup Language (XML) resulting in "fluid" data that removes the shackles that proprietary vendor solutions place on your enterprise. By facilitating the capture of business targets, best practice patterns and decision rationale with common mechanisms your enterprise will evolve and be competitive.

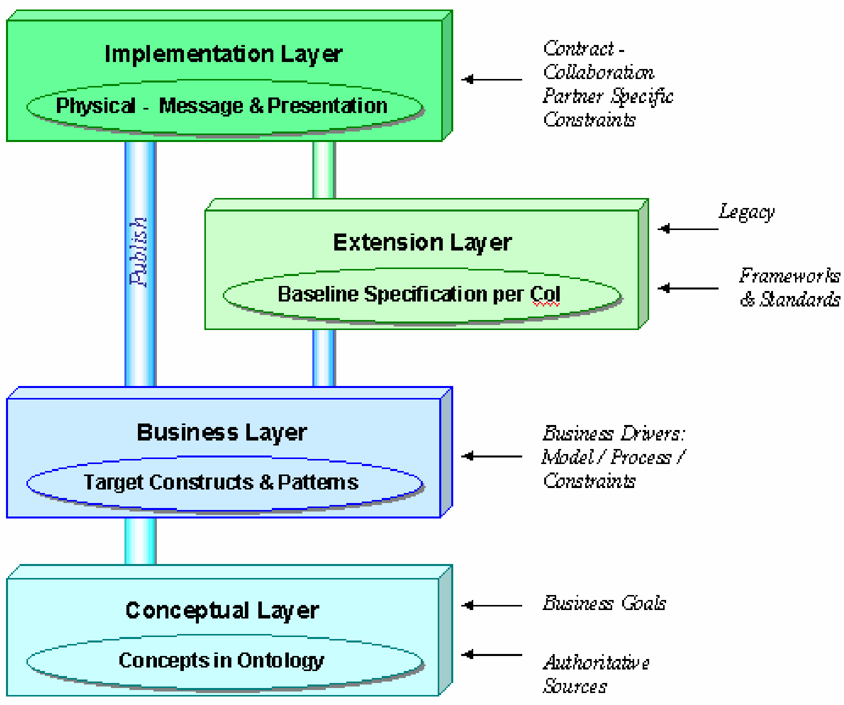

Figure C-3: Business-Centric Methodology Layers Overview

One objective of the business-centric model is to graphically represent the variety of shared artifacts for reuse, each having different constraints exercised. Reference figure C-3. By applying the right constraints at the right level, and not physicalizing them too soon, the process enables business, not technology to drive the exchange. The result is a far more agile enterprise. Information exchange and proper interoperability are possible if, and only if alignment occurs from the (1) Implementation, (2) Extension, (3) Business, and (4) Conceptual layers.

- **Implementation** – involves performing in-depth technical requirements analysis of the message and the selected framework driven by the Collaboration Partner Agreement (CPA). It is here where business objects become physical with agreed upon XML tagnames, lengths, header information, etc. In addition to the output of the message, maps are published for possible reuse and aligned concept aliases are registered for later reference.
- **Extension** – provides outreach for mapping the enterprise Target constructs to the desired industry consortiums, standard bodies, and internal legacy systems format. The product of this mapping includes a *Baseline Specification* for each desired community perspective.
- **Business** - understanding of the core business goals that the "preferred" business objects must accomplish, constrained by the defined business processes and patterns. Business rules allow for the capture of enterprise logic by analyzing the impact of changes, identifying areas of reuse and defining functional requirements from an

enterprise perspective – *Target Constructs* in your business context.

- **Conceptual** – aligning terminology by understanding the semantics of the business and uncovering the real meaning of the business vocabulary use can be extracted and interpolated to higher business aggregates. One of the primary resulting products is the *Concept Definition Template*.

Key to the above is providing the facilitation infrastructure for artifact discovery and navigation and the classification and ontology for the clustering of like terms and to differentiate business terms usage through decomposition.

The prime shift components are -- [1] Taxonomy, [2] Registry, [3] Workflow, and [4] Content management system. Various facetted taxonomy views of the business with the capability of defining thesaurus (e.g. synonyms, alias) relationships that reside on a registry. Advances later may include more complex structures to round out a more complete ontology. The registry provides reference assistance and stores information about the supporting classifications and metadata artifacts. This occurs independent of them being link references to external artifacts or links to stored artifacts in the content management system(s) and processed workflow. The workflow allows for the status of the enterprise's value-chain 'pipelines' to be analyzed and corrections made quickly. The links and relationships assist the discovery, search, and notification services by providing a mechanism for cooperative actions. Metadata in many cases provides the critical controls and metrics of the enterprise and only together with the ideas above does the enterprise have a holistic solution for integration. Other critical supporting services/components are – [5] Indexing/serach, [6] Visualization Tools, [7] Template Processor, [8] Rules/Mapping Engine, and [9] other collaboration tools such as group authoring for information enabling, group leveraging, function sharing, and information sharing.

With Business-Centric Methodology, your enterprise can not only take advantage of technology innovations that complement and enhance the architecture, but also provide the environment to foster vendor development of technology that exploits instead of attempting to make obsolete the deployed systems. In short, BCM provides the base for mass customization required - supporting the enterprise's stakeholders and customers.

Appendix D: Advantages Gained By Moving to a Registry-based SOA

The following diagram depicts the trend toward loosely-coupled, metadata-centered Service-Oriented Architecture (SOA):

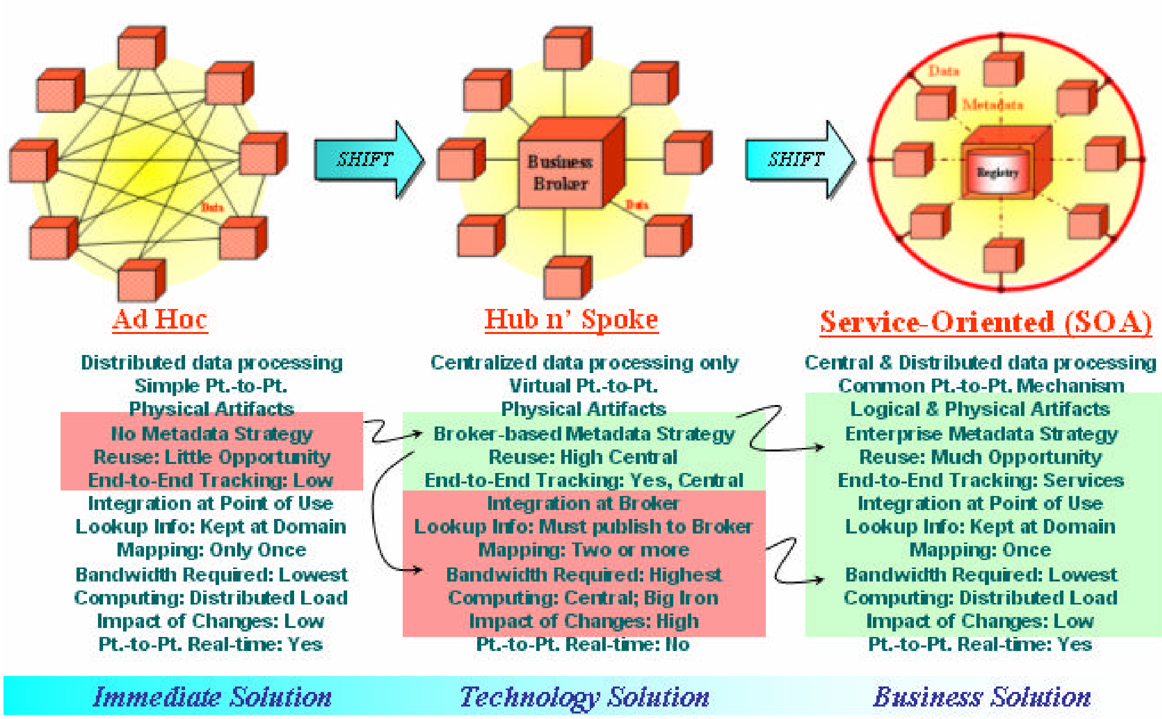

Figure D-1: Momentum towards Service-Oriented Architectures (SOA)

The left most view shows the typical point-to-point interfaces (internal and external) of an enterprise. The business communication from sender and receiver is direct, but at the low levels very cumbersome. Each interface requires detailed knowledge of each other's application for mapping values, there is little in the way of sharing metadata between entities, as we have neither a standard format nor automation. Suffice to say because of this the mapping task requires great technical skills. The task can't be based on lessons learned in the past because each interface is almost starting from scratch, with little or no reuse. For audit purposes there isn't a central control point, nor is there an end-to-end tracking mechanism. The perceived problem is the n^2 (N-squared) challenge; that the number of mappings between the endpoints can become astronomical.

One architecture that can be applied to address the issues described above is a "hub n' spoke" pattern, creating an integration broker for the enterprise – as shown in the middle diagram.

From a conceptual viewpoint the system looks much simpler than a point-to-point approach and it is therefore easy for business managers to buy into this approach, even from a 30,000' level. Looking closer at the benefits of this hub n' spoke scenario we find some very positive characteristics; primarily that, through centralization, we can add consistency, share expertise, and gain other controls required for robust message passing. But it is important to recognize a few of the side effects of this solution:

- We have added another processing step, slowing the flow of information.
- We have added another possible area for mistakes to be made, leading to a potential loss of information; we now have two maps (point-to-hub and hub-to-point) whereas before we had just one (point-to-point).
- We have added a constraint to our architecture by forcing a centralized design.
- We have moved into a queuing environment, eliminating any solutions where synchronous communications are nearly impossible with heavy loads. This single point integration could lead to bottlenecks if not architected with load balancing techniques.
- Changes in the hub affect all interfaces, thereby impeding the fulfillment of business requirements.
- It is a possibly incomplete solution; if we don't constantly feed the hub with new domain reference tables, we will need to perform lookup processes at the endpoint (thus requiring a mapping or joining at the destination). For instance, if current manufacturing information is to be added to the purchase order, it can be added to the transaction at the hub, if the hub is updated, or needs to be joined in at the receiving application. In some cases, at the application is not an option, with this information being added into note fields for users.
- There is a need for larger machines, as the computing power at the hub is taxing.
- The number of messages being passed around is doubled, thus requiring greater communication bandwidth.

The integration broker is likely to be a vendor solution that doesn't completely meet the business needs. The 30,000' view looks good until we implement and discover the new problems that have cropped up – the hub n' spoke approach tends to not decrease the organization's efforts, rather the effort either remains the same or even increases, but with a different set of problems.

The ideal solution is one that combines the best of both worlds, in other words a hybrid solution, (depicted on right of the diagram) which gives us the optimum result and eliminates many of the problems associated with the point-to-point and hub n' spoke approaches.

So what should be decentralized? The information, the transactions, the Web queries, the Web Services, (in other words, data). What do we keep centralized? Metadata (context) and a few support functions such as end-to-end status mechanisms (logging, etc.). What about our 'N-squared' problem? While mathematically correct, in business typically communication patterns is a very small subset of the theoretical possibilities, does every application in your business 'talk' to each other application? Of course not, but for sure the best, most accurate communication between each required exchange is direct understanding of each application's requirements. Note that the hybrid solution has only been economically feasible for medium to large enterprises in the last few years with XML-based tools coming to market. Being that the hybrid solution is a distributed model, it allows the enterprise to coexist with centralized infrastructure components as well – it supports a mix, as an enterprise balances its functions and those processes which work best in a synchronous environment are weaned to a distributed model. The bottom line is that the hybrid (registry-centric) approach isn't driven by technology but is instead properly guided by business requirements.

Principles of Partitioning and Decoupling

One of the objectives of the Federal Enterprise Architecture is to promote proper partitioning and loose-coupling by dividing the enterprise in divisible components that may be swapped out with a minimal impact on the rest of the applications. Referencing Figure B, integration can be at the (1) Data level, (2) Activity or process level, and (3) User Interface level. For example Application #1's Data to Application's #2 Data by copying or replicating databases. Or in a tight-coupling arrangement Application #1 can share a database with Application #2, or read/write directly through a database access via ODBC or JDBC – these approaches need to be viewed as 'hard-coded' and may present problems over time. At the activity level, a program can call another's activity (as depicted in the diagram). And as experienced on the Web, a user interface can call or bring up another user interface to allow users to create/edit other information. Certainly, there is a mix of these three levels as well, one user interface calling out to a remote function (activity), or screen scrapping of presentation objects or report objects into a application. Integration can be manual or automatic. In addition to the technical swap out aspects, these divide and conquer approaches tend to reduce overall costs due to the reduction in complexity of the functionality, even though the number of interfaces may increase.

Implementation of a Service-oriented Architecture (SOA) as outlined in Figure D-2 provides an example application of the management of business services with several defining benefits. The figure depicts an Enterprise Information Services Layer (EISL) that incorporate *Web services* to build upon the notion of collaborative eBusiness commerce, enterprise application integration (EAI), workflow, and business process integration. Adopting and developing the XML-based EISL will allow e-Government to deliver services and content externally to a wide variety of audiences and physical environments. It will also allow e-Government to internally standardize on a service-oriented architecture enabling interoperability between internal federal and state agencies.

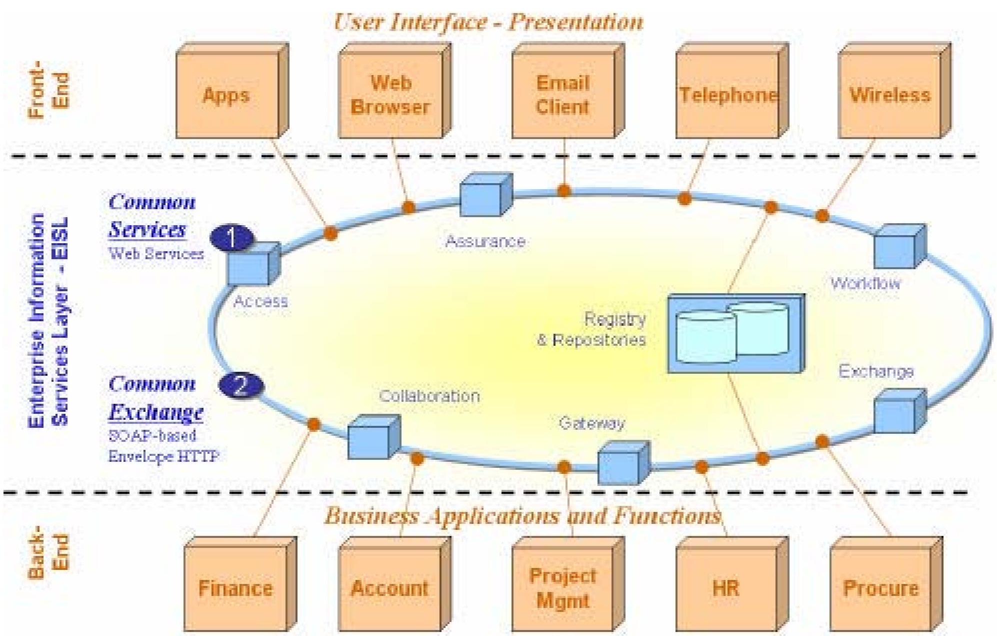

Figure D-2: Integrated Access based on Conceptual View of E-Government Infrastructure

The EISL will contain a set of web services (i.e., query/response services) that will be used to interact with the Registry as described below). The architecture provides the benefits in an effective manner:

- The ability to rapidly specify controlled changes to business operations in business terms and objectives, and to implement many of those changes directly in near-time (business agility and integrity)
- The ability to measure the effectiveness of those changes in business terms (business metrics and analysis)
- The ability to provide implementation services for business lines – such as Finance, Accounting and HR – aligned with business line objectives, thereby optimizing operational costs and efficiently (operational services alignment)
- Eliminates the need for developers to repetitively design and build the same software components that solve the same set of recurring business problems, over and over again.
- Separation of the 'what' (goals) from the 'how' (resource management), except insofar as knowledge and use of specific processes is dictated by the goals (process and data independence). This allows end users to concentrate on the goals of the business, and not have to understand the facilities for use.

SOA services are offered as components to anyone, anywhere via a computer network. This means that any distributed service application can interact with any other service-based application regardless of either's network location. Thus, a centerpiece component of a SOA solution is the *Registry* for discovery and to aid in reflection of the services. The Registry provides the means to coordinate namespaces, nomenclature, and data standards across departments, outside industry and with other government initiatives. The concept of the Registry discussed here is a *logical registry*, one that encompasses many physical registries to support the infrastructure of the Federal and State governments. One could substitute the term Registry with that of a "*Registry System*". The System incorporates distributed supporting bases/nodes required for operation together in combination housing our government's ontology. Each component in the Registry System is aware and communicates with the appropriate other components to create a network of components, and managed through a 'Registry of Registries.' Existing repositories within the Government can be thought of an extension of the Registry, where the actual business transactional or records information described by the registered items may be obtained.

Distributed, scalable registries for business processes, transaction characteristics, and *context everywhere* are essential to managing the continued increase capability. Initially the communication may be through manual execution of batch executions, but in time these links will become more automated, and evolve into Web services of the Information Services Layer.

The Registry will greatly benefit e-Government by making data more coherent throughout the Federal and State governments by standardizing business artifacts such as nomenclature and providing through automation active participation. e-Government will achieve harmonization by creating and maintaining only one set of prime components and map other components to the prime components as required. The first step toward harmonization is to extract the common terminology, properties, organization, and processes used by many of the business lines and then create a generic framework for developing new, or updating existing, components. Because similar procedures can be applied to related components, the implementation and the development of new crosswalks are simplified. This active view of registries moves beyond the thinking of static simple lookups approach similar to that of a dictionary. The benefits of the e-Government Registry are:

- Allows for discovery of processes – for function and service which to build applications
- Promotes reuse – system developers can locate a business object in the Registry will same time and effort, and reduces the number of required crosswalks
- Enables efficient version control – the Registry enables tracking multiple versions of a business object efficiently
- Promotes unified understanding of registered objects - metadata for registered objects are accessible from a single location, a unified understanding of the purpose and rationale can be maintained
- Allows for collaboration – finding partners (internal or external) connected to the metadata to share ideas and receiving notifications as to configuration changes
- Enables navigation of business – with metrics assigned via processes or users, management can see at an enterprise level operations at a glance
- Assists with impact studies – provides input as to changes and how it impacts the organization, also benefits gap analysis as well
- Collect independent metadata – which is separate from COTS tools to supplement capture of required business information that can not be housed in the products
- Organization's methodology – through the use of consistent templates and information-driven wizards for capture of user's input
- For orchestration of services – by taking a information-driven approach to sequencing and invoking functions throughout the enterprise, and at the enterprise level

Appendix E: Homeland Security Example

OASIS CAM TC is developing XML templating that allows users to construct information exchanges that are adaptive while remaining consistent and coherent. The exact interchange information is driven by business context parameters. This is ideal for reporting where the exact requirements are only finalized at runtime.

The technology has already proved valuable for managing Telco trouble ticket reporting with hundreds of different messages, partners and service offerings that change rapidly in a highly commercial production environment.

For more information on the OASIS CAM standards technology see:
http://www.oasis-open.org/committees/cam/

The CAM approach combined with secure internet based messaging using the OASIS ebMS that recently US Gov CDC developed an engine for –provides an ideal mechanism for delivery and processing using a low-cost and open platform model.

OASIS CAM for Notifications Example

The syntax shown here is a fragment of an OASIS CAM template that could be applied to a variety of alert and dynamic notification problems.

```
<CAM xmlns:as="http://www.oasis-open.org/committees/cam">
   <AssemblyStructure>
      <Structure as:choiceID="TR-0001">
         <as:include>SecurityReport.xml</as:include>
      </Structure>
      <Structure as:choiceID="TR-0002">
         <as:include>SecurityNotificationRequest - Clear Request.xml</as:include>
      </Structure>
      <Structure as:choiceID="TR-0003">
         <as:include>SecurityNotificationRequest - ClosureRequest.xml</as:include>
      </Structure>
      <Structure as:choiceID="TR-0004">
         <as:include>SecurityNotificationRequest - StatusQuery.xml</as:include>
      </Structure>
      <Structure as:choiceID="TR-0005">
         <as:include>SecurityNotificationRequest - Change.xml</as:include>
      </Structure>
      <Structure as:choiceID="TR-1022">
         <as:include>SecurityNotification - ClearConfirm Accepted.xml</as:include>
      </Structure>
      <Structure as:choiceID="TR-1023">
         <as:include>SecurityNotification - ClearConfirm Rejected.xml</as:include>
      </Structure>
      <Structure as:choiceID="TR-1024">
         <as:include>SecurityNotification - ClearReject Accepted.xml</as:include>
      </Structure>
      <Structure as:choiceID="TR-1025">
         <as:include>SecurityNotification - ClearReject Rejected.xml</as:include>
      </Structure>
   </AssemblyStructure>
   <BusinessUseContext>
```

```
      <Rules>
         <default/>
         <context condition="token='%AlertCondition%' and contains(value,'TR-0001')">
            <constraint action="useChoiceByID(TR-0001)"/>
         </context>
         <context condition="token='%AlertCondition%' and contains(value,'TR-0002')">
            <constraint action="useChoiceByID(TR-0002)"/>
         </context>
         <context condition="token='%AlertCondition%' and contains(value,'TR-0003')">
            <constraint action="useChoiceByID(TR-0003)"/>
         </context>
         <context condition="token='%AlertCondition%' and contains(value,'TR-0004')">
            <constraint action="useChoiceByID(TR-0004)"/>
         </context>
         <context condition="token='%AlertCondition%' and contains(value,'TR-0005')">
            <constraint action="useChoiceByID(TR-0005)"/>
         </context>
         <context condition="token='%AlertCondition%' and contains(value,'TR-1022')">
            <constraint action="useChoiceByID(TR-1022)"/>
         </context>
         <context condition="token='%AlertCondition%' and contains(value,'TR-1023')">
            <constraint action="useChoiceByID(TR-1023)"/>
         </context>
         <context condition="token='%AlertCondition%' and contains(value,'TR-1024')">
            <constraint action="useChoiceByID(TR-1024)"/>
         </context>
         <context condition="token='%AlertCondition%' and contains(value,'TR-1025')">
            <constraint action="useChoiceByID(TR-1025)"/>
         </context>
      </Rules>
   </BusinessUseContext>
   <ContentReference>
      <Addressing>
         <registry name="hls" access="http://xml.gov/hls/alert/CDC" method="URL"
description="CDC Security Alert Central Repository"/>
      </Addressing><!--78 Items-->
      <item type="noun" name="AccessDetails" UIDRef="hls000007" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="AccessDetails/@Notes" UIDRef="hls000877" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="Agency" UIDRef="hls000207" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="Agency/@AgencyID" UIDRef="hls000910" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="Code" UIDRef="hls000724" taxonomy="UID" registry="hls"/>
      <item type="noun" name="ContactName" UIDRef="hls000171" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="CurrentStatus" UIDRef="hls000327" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="Description" UIDRef="hls001001" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="DetailedContact" UIDRef="hls000041" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="EndTime" UIDRef="hls000786" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="ErrorInfo" UIDRef="hls000284" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="FaultCode" UIDRef="hls000941" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="Ident" UIDRef="hls000209" taxonomy="UID"
registry="hls"/>
```

```
      <item type="noun" name="ItemNumber" UIDRef="hls000584" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="ListOfSecurityNotificationDetail" UIDRef="hls000635"
taxonomy="UID" registry="hls"/>
      <item type="noun" name="ListOfSecurityNotificationRequestDetail"
UIDRef="hls000617" taxonomy="UID" registry="hls"/>
      <item type="noun" name="ListOfSecurityReportDetail" UIDRef="hls000590"
taxonomy="UID" registry="hls"/>
      <item type="noun" name="ListOfSecurityReportResponseDetail" UIDRef="hls000593"
taxonomy="UID" registry="hls"/>
      <item type="noun" name="MessageInfo" UIDRef="hls000310" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="Msg" UIDRef="hls000598" taxonomy="UID" registry="hls"/>
      <item type="noun" name="Note" UIDRef="hls000224" taxonomy="UID" registry="hls"/>
      <item type="noun" name="Party" UIDRef="hls000195" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="Party/@PartyID" UIDRef="hls000909" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="ReceivedDate" UIDRef="hls000595" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="Reference" UIDRef="hls000730" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="RefNum" UIDRef="hls000193" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="ReportingParty" UIDRef="hls000608" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="ReportingReference" UIDRef="hls000610" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="ServiceID" UIDRef="hls000612" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="ServiceLevel" UIDRef="hls000780" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="ShortDescription" UIDRef="hls000938" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="StandardQuestions" UIDRef="hls000585" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="StandardQuestions/@DisruptiveTestsAllowed"
UIDRef="hls000929" taxonomy="UID" registry="hls"/>
      <item type="noun" name="StandardQuestions/@EquipmentChecked" UIDRef="hls000928"
taxonomy="UID" registry="hls"/>
      <item type="noun" name="StandardQuestions/@PowerChecked" UIDRef="hls000927"
taxonomy="UID" registry="hls"/>
      <item type="noun" name="StandardQuestions/@ServiceCurrentlyInUse"
UIDRef="hls000925" taxonomy="UID" registry="hls"/>
      <item type="noun" name="StandardQuestions/@ServiceHasWorked" UIDRef="hls000926"
taxonomy="UID" registry="hls"/>
      <item type="noun" name="StartTime" UIDRef="hls000785" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="Status" UIDRef="hls000792" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="Telephone" UIDRef="hls000172" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="TestResults" UIDRef="hls000794" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="SecurityNotification" UIDRef="hls000632" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="SecurityNotificationDetail" UIDRef="hls000636"
taxonomy="UID" registry="hls"/>
      <item type="noun" name="SecurityNotificationHeader" UIDRef="hls000633"
taxonomy="UID" registry="hls"/>
      <item type="noun" name="SecurityNotificationRequestItemMessageInfo"
UIDRef="hls000628" taxonomy="UID" registry="hls"/>
```

```
      <item type="noun" name="SecurityNotificationRequestMessageInfo"
UIDRef="hls000627" taxonomy="UID" registry="hls"/>
      <item type="noun" name="SecurityReport" UIDRef="hls000579" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="SecurityReportContact" UIDRef="hls000556" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="SecurityReportDate" UIDRef="hls000581" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="SecurityReportDetail" UIDRef="hls000583" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="SecurityReportHeader" UIDRef="hls000582" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="SecurityReportMessageInfo" UIDRef="hls000603"
taxonomy="UID" registry="hls"/>
      <item type="noun" name="SecurityReportParty" UIDRef="hls000580" taxonomy="UID"
registry="hls"/>
      <item type="noun" name="SecurityReportReference" UIDRef="hls000597"
taxonomy="UID" registry="hls"/>
      <item type="noun" name="SecurityReportResponse" UIDRef="hls000591"
taxonomy="UID" registry="hls"/>
      <item type="noun" name="SecurityReportResponseDetail" UIDRef="hls000594"
taxonomy="UID" registry="hls"/>
      <item type="noun" name="SecurityReportResponseErrorInfo" UIDRef="hls000606"
taxonomy="UID" registry="hls"/>
      <item type="noun" name="SecurityReportResponseHeader" UIDRef="hls000592"
taxonomy="UID" registry="hls"/>
      <item type="noun" name="SecurityReportResponseMessageInfo" UIDRef="hls000605"
taxonomy="UID" registry="hls"/>
   &lt;/ContentReference&gt;
&lt;/AssemblyDoc&gt;
```

Figure E-1: OASIS CAM XML example for 8 different alert format.

The example shows how content is driven by context rules, and also by a standard dictionary of content "nouns" that allow sender and receiver to explicitly define the details of the information to avoid discrepancies and errors.