

ATHABASCA UNIVERSITY

ENTERPRISE ARCHITECTURE FRAMEWORK APPLICATIONS TOWARDS  
SERVICE ORIENTED ARCHITECTURE

BY

COLLIN SMITH

A thesis essay submitted in partial fulfillment  
Of the requirements for the degree of  
MASTER OF SCIENCE in INFORMATION SYSTEMS

Athabasca, Alberta

March, 2008

© Collin Smith, 2008

## DEDICATION

I would like to dedicate this essay to the loving memory of my mother, Judy Smith (1945-2008), and all those who must face the struggle with cancer.

## ABSTRACT

A large challenge facing today's enterprises is the integration of their own disparate systems to be more competitive and react more strategically to market opportunities. Service Oriented Architecture(SOA) is a style of architecture where the use of reusable and discoverable services are used as components to meet business requirements. The promise of SOA is that the use of these services will allow the organization to increase business agility and reuse of software components. There are some challenges to implementing a SOA. One of the challenges is to present view from the enterprise perspective. Typically, projects take only a view of individual business unit without concerns for the entire organization. Enterprise Architecture(EA) describes the current business and IT processes and how they map together. The two most popular EA Frameworks, Zachman Framework and The Open Group Architecture Framework(TOGAF), are designed to help the organization clarify the current IT Architecture and help provide a roadmap to the goal or future IT architecture. It is with the enterprise view that these frameworks provide that SOA can be supported towards a more successful implementation. There is work underway to determine the relationship between SOA and EA. Whether by clarification from industry experts or direct modifications of the EA Frameworks, EA Frameworks are being used to help implement SOA. It is through the enterprise view and architectural tools provided by the EA Frameworks that SOA can be supported.

## ACKNOWLEDGMENTS

I would like to acknowledge the guidance I received from Dr. Qing Tan during this process. I gratefully appreciate the feedback provided by Jack Kerkhof during the later stages of this essay, when I found it increasingly difficult to revise.

# TABLE OF CONTENTS

1	CHAPTER 1 – Introduction.....	1
1.1	Background Review of EA and SOA .....	2
1.2	Technology Overview .....	4
1.3	Statement of the Purpose .....	5
1.4	Research Problem .....	5
1.5	Significance .....	6
1.6	Organization of this paper.....	6
2	CHAPTER 2 – Service Oriented Architecture .....	8
2.1	Services.....	9
2.2	SOA Governance .....	9
2.3	Service Level Agreements.....	12
2.4	Enterprise Service Bus .....	13
2.5	Metadata.....	15
2.6	Service Composition and Collaboration.....	15
2.7	Defining SOA.....	16
2.8	Conclusion.....	18
3	CHAPTER 3 – Web Service Status .....	20
3.1	History of WS.....	20
3.2	The WS Specification Process .....	22
3.3	Review of more important WS-Specifications.....	24
3.3.1	WS-Reliable Messaging .....	24
3.3.2	WS-Coordination .....	26
3.3.3	WS-Security .....	29
3.3.4	WS-BPEL .....	30
3.4	Conclusion.....	32
4	CHAPTER 4 – EA Frameworks .....	35
4.1	EA Background.....	35
4.2	Zachman Framework.....	37
4.3	TOGAF Framework in Detail .....	41
4.3.1	Architectural Development Method(ADM) .....	43
4.3.2	Enterprise Continuum.....	44
4.3.3	TOGAF Resource Base.....	47
4.4	Comparison of Zachman and TOGAF .....	48
4.5	Conclusion.....	50
5	CHAPTER 5 – EA & SOA Together.....	51
5.1	Why EA Frameworks and SOA are a good match.....	51
5.2	Review of the Zachman Framework towards SOA .....	51
5.3	Review of TOGAF towards SOA .....	54
5.4	Conclusion.....	59
6	CHAPTER 6 – Recommendations & Conclusions .....	63
6.1	Goals of EA and SOA.....	63
6.2	How EA Frameworks help SOA.....	63
6.3	Caveats of the EA Frameworks .....	64
6.4	Conclusion.....	65
7	REFERENCES .....	67

## LIST OF FIGURES

Figure 1 Enterprise Service Bus.....	14
Figure 2 Reliable Messaging Format.....	25
Figure 3 WS- Coordination.....	28
Figure 4 Sample BPEL process definition using ActiveBPEL™ Designer.....	31
Figure 5 EA Frameworks History.....	37
Figure 6 Zachman Framework.....	39
Figure 7 Architectural Development Method.....	43
Figure 8 Architecture & Solutions Continuum.....	45
Figure 9 Integrated Information Infrastructure Reference Model.....	47
Figure 10 Scope of TOGAF onto the Zachman Framework.....	49
Figure 11 Mapping SOA to Zachman.....	53

## **1 CHAPTER 1 – Introduction**

Today's large enterprises have vast arrays of information systems applications. The variety of applications is numerous covering applications such as core business processes, accounting, billing, human resources to name a few. These applications not only vary in functionality but also the underlying technologies. As business evolves, enterprises are seeking a faster way to harness these strategic IT assets to remain competitive and take advantage of market opportunities. The challenge facing today's enterprise is integrating these applications and positioning the enterprise for current goals and those of the future. Other important goals are increased quality, better integration and better business alignment. Enterprise Architecture deals with the description of the enterprise's current processes and information systems to align these assets with current and future goals of the organization

Service Oriented Architecture (SOA) is a style of architecture by which information systems provide services which are to be available for consumption within the organization. These services provide standardized contracts from which clients may use them either individually or within a composition of services. This architecture, if implemented correctly, is designed to provide quicker "time to market" and increase code reuse. Current important issues facing CIOs and CTOs are business and IT alignment, moving from a department to an Enterprise view of data and functionality and the navigation of vendor buzzwords and

inconsistent terminology. SOAs are most commonly implemented with the web service or WS-\* specifications. Web service specifications are driven by a formal process requiring major IT vendors to vote to create these formal specifications. Even with this specification process, it is difficult for IT management to determine the technological future of the enterprise. It is through the use of Enterprise Architecture, a systematic and technologically agnostic approach to successfully implement an SOA. Enterprise Architecture(EA) Frameworks are methodologies used to help shape IT direction. It is through an EA framework that one may describe an organizations IT assets and help position itself for its strategic goals. Two of the more popular EA Frameworks will be examined in more detail. These are the Zachman Framework and The Open Group Architecture Framework(TOGAF). In particular, the use of EA frameworks towards an SOA will be reviewed.

## **1.1 Background Review of EA and SOA**

Enterprise Architecture is “description of the current and/or future structure and behavior of an organization's processes, information systems, personnel and organizational sub-units, aligned with the organization's core goals and strategic direction” (Wikipedia, 2007a). Defining what the organization currently does accurately will allow the organization to align more effectively business processes and IT assets. Large organizations have a wide array of both business processes and IT assets. EA is designed to help manage this complex alignment process and help navigate IT to better serve the business purposes



from an enterprise viewpoint. EA is not just the current state, but includes the current state, the goal state and how helps visualize the road map to get to the goal state. Business units often create disparate sets of applications and processes with a localized viewpoint. This often leads to problems when an organization must apply an enterprise wide process. An example of this could be Sarbanes-Oxley compliance for all business units. An enterprise view can help create more harmony between the various business units. EA frameworks have been around at least two decades. The Zachman framework is credited as the Zachman framework was formally published in 1987 by John Zachman. This framework proposes a logical structure for classifying different views or dimensions of the enterprise. Another popular Enterprise Framework, TOGAF, created in 1995, presents an iterative process for defining and shaping the enterprise architecture. TOGAF's process is called Architectural Development Method(ADM). EA Frameworks help manage the complexity of aligning business processes with IT assets.

SOA is a style of IT architecture. SOA is based on the idea of building applications out of services. Traditionally, applications are made up of programs which make various calls to library code or other source code. SOA groups independent services together in a process known as orchestration. Orchestrations themselves become a service. One goal of SOA is to improve reuse by having an existing set of services which may be reused in many different contexts. Another goal is to improve the organization's agility in responding to business process changes. Service orchestration via existing

services is to allow the organization to react quicker than traditional software development process. The current most popular implementation of SOA revolves around web services.

## **1.2 Technology Overview**

Although SOA is not technology specific, the current most popular technology for SOA is web services. The specification of web services is driven by organizations such as OASIS(Organization for the Advancement of Structured Information Standards) and W3C(The World Wide Web Consortium).

OASIS is “a not-for-profit consortium that drives the development, convergence and adoption of open standards for the global information society”. (Oasis, 2007) OASIS is comprised of membership organizations from a wide variety of industries and geographies. W3C is also made up of industries all over the world and in different fields. One of the promises of SOA is to allow services to technology independent. They may be written in a variety of languages. It is the web service specifications that allow organization’s to have more vendor and product options. The web service technology allows services to be created in different technologies and be interoperable.

The core specifications of web services are SOAP(Simple Object Access Protocol), WSDL(Web Services Definition Language) and UDDI(Universal Description Discovery and Integration). SOAP messages are also formatted in XML(Extensible Markup Language). W3C has produced the specifications for

XML, SOAP and WSDL. OASIS has produced the UDDI specification. The core specifications have been extended and these additional specifications are referred to as WS-\*. These additional specifications are being formed by vendor alliances, OASIS and W3C. There is competition between various web service specifications. This makes the web service a slow process but on the other hand a successful one. Vendors are all participating in the specification process which will lead to better interoperability for web services in the long run.

### **1.3 Statement of the Purpose**

The purpose of this paper is to show the value that Enterprise Architecture Frameworks towards an Service Oriented Architecture. An overview of Service Oriented Architecture, Web Services, and Enterprise Architecture will be presented. The use of two popular EA Frameworks, Zachman and TOGAF, will be reviewed on there uses in industry. This review will be done to show how enterprises can exploit their current IT assets in strategic initiatives within a process that will be less sensitive to vendor or technology changes. Most of the current literature focuses on one aspect of either EA or SOA. This paper will focus on the aspects that make EA Frameworks, in general, useful towards a successful SOA implementation.

### **1.4 Research Problem**

The problem that faces many organizations today is to get the most from their enterprise applications. Service Oriented Architecture is being pursued by many of today's organizations as a way to leverage their existing applications to

become more competitive strategically. The pursuit of this style of architecture is not straight forward. A move towards SOA for an enterprise has very wide ramifications. There are many varying viewpoints on SOA and a wide selection of literature on SOA of varying quality. It is best to approach this type of problem with a technology and vendor agnostic manner. The problem facing IT leaders in organizations is how to transition to SOA in an effective manner to gain the most benefit in a careful and systematic manner. As the popularity of SOAs increase, the need for a structured approach to their development will follow.

## **1.5 Significance**

Service Oriented Architecture is a very popular architecture currently being presented by major IT vendors to integrate IT applications and resources. These are very costly endeavors due to the size and span of an enterprises IT assets. An appropriate methodology must be put in place to manage this change. This methodology should be vendor and technology neutral as vendors and technology change over time. The methodology offered by EA Frameworks should fill this gap to manage the complexity of application integration with SOA.

## **1.6 Organization of this paper**

This paper will attempt to bridge a few main concepts to explain how an effective approach to SOA development is possible with the use of EA Frameworks. There will be chapters devoted to Service Oriented Architecture, current web service standards, Enterprise Frameworks, and the use of Enterprise

Frameworks towards SOA. The use of Enterprise Frameworks towards SOA will be drawn from current views from it leaders, framework groups and literature. Lastly, conclusions will be drawn on the benefits and caveats to using EA Frameworks for SOA initiatives.

## **2 CHAPTER 2 – Service Oriented Architecture**

Service Oriented Architecture is an architectural style gaining in popularity through the promise of two main goals: quicker time to market and increased reuse. The first goal implies that if one has an existing set of services that can be reused for a given enterprise initiative, this will reduce the amount of net new code that is required. Reducing the amount of time to take a business idea from concept to implementation has always been crucial for business. SOA improves business agility. The second goal of increased reuse has always been a goal for IT. This implies that economies can be made by well written code being reused over and over again. Increased reuse always implies a higher level of cost effectiveness and quality. Reacting quickly to market initiatives and reduced costs help provide business benefit to the Enterprise.

Along with the definition of SOA, there are also some key concepts that are key to SOA and its implementation. Key concepts of SOA will also be touched upon to give the reader a better sense of how these fit within SOA and why they are important. Once the main concepts have been illustrated, a review of existing SOA definitions from various vendors and authors will be examined. From examination of these definitions and the key concepts, another definition will be created in hopes that it will better encapsulate SOA in general.

## **2.1 Services**

The basic unit or building block of an SOA is the concept of a service. Services are independent units of business logic that interact to achieve some business end. These services are to be loosely coupled implying minimal coupling and maximum cohesion. This means that services may be easily assembled to achieve IT initiatives. Many organizations are broken down into different business units or business silos. Each of these business units have their own goals and objectives to achieve through project based applications. Services are a different way of viewing the enterprise. Traditionally, one treats applications as the unit by which business process is automated. Applications were often integrated via some Enterprise Application Integration(EAI). Services are a more granular view with services acting as the unit by which one can automate business processes. Services are also composeable where a services can be put together to form a new business process., which may include services external to the enterprise. Services should also be discoverable based on their descriptions, service levels or other metadata.

## **2.2 SOA Governance**

Service Oriented Architecture requires proper process in order develop effectively. Anne Thomas Mane states that “Governance refers to the processes an enterprise puts in place to make sure things are done right, that is in accordance with best practices, architectural principles, legal and industry regulations and other factors” (Mane, 2005). SOA governance means putting

process in place for the management of the Enterprise SOA. Governance should reduce functionality duplication that would occur otherwise. Thought should be given to how project based work fits into the Enterprise view that the SOA should be seen from. Services must be created with care and thought to future reuse.

Marks & Bell state that “SOA governance refers to the organization, processes, policies, and metrics required to manage an SOA successfully” (Marks & Bell, 2006a). Governance is crucial to the success of an SOA implementation. I believe governance to be the most crucial portion of SOA with technology being far less important. Governance involves many different components including oversight, management and standards. As one rises above a simple project based view, issues such as ownership, stewardship become important in clarifying how the SOA will be driven. Without firm governance, it is doubtful that the benefits of an SOA can be achieved. Governance will ensure that an SOA will be supported and defined to allow it benefit the enterprise through a thoughtful and organized fashion. The danger of trying to do SOA without a good governance strategy is that the enterprise goals will be sacrificed and ignored during individual project work. This would lead to more point to point integration which is what SOA is to prevent. The definition of project success needs to change to include overall enterprise goals as well.

Paul C. Brown states that “governance is the process of risk management” (Brown, 2007). He outlines that there are four major process requiring governance. These are project portfolio planning, service design, service utilization, and service operation.



Firstly, governance for project portfolio planning involves the process of prioritizing ideas for new projects. ROI on the first use of a service is not expected on the first implementation. Governance will involve speculating on what is important to the business to schedule service creation within project schedules. Governance review should monitor progress of the service plan as well.

Secondly, governance for service design involves process of determining how and why a service should be created. There are 3 points involved with the governance of service creation or design:

- Ensuring that the creation of the service is appropriately justified and specified along with getting senior architects involved and responding in a timely manner within a proper process
- Ensuring that the service is implemented appropriately by ensuring that the service will respond to future demands as well
- Ensuring that the service development process produces the artifacts required to support its future usages including various descriptions to be stored in repositories to easily find them for reuse in the future

Thirdly, governance of Service Utilization involves ensuring that services are used and reused properly. Firstly, project teams must be educated on the enterprise's SOA strategy and be aware of how to find and use existing structures. The fact that services are being used appropriately during the architecture review for a project should be ensured. Also, ensuring that services are adequately prepared for their intended purpose is important.

Lastly, Brown mentions the Governance of Service Operation which includes three broad areas: performance monitoring, new usage planning, and upgrade planning.

There are a lot of details with respect to SOA governance. SOA Governance is the process of managing the SOA itself. It involves analysis upfront to ensure that services are built within project work based on current priority and future needs. The need for project work to be involved with the SOA governance through education and guidance cannot be overemphasized.

This is the area of SOA that requires the most effort, SOA governance is the guiding force which can help ensure that services are properly created and used. Without SOA governance, there is only service creation without proper creation, education and reuse processes.

### **2.3 Service Level Agreements**

Service Level Agreements are mandatory performance standards for given services. The service providers and consumers both are interested in the responsiveness and availability of services. Consumers want assurances that a given service can accomplish the task within their requirements. By defining a level of service, consumers can have more confidence in choosing a service and the provider can set better expectations for potential consumers.

To honor the SLAs, services must be monitored with respect to information as start times and end times. Service statistics must be published to consumers to show the level of service they are receiving. There may be penalties and

rewards with respect to the level of service provided. Monitoring and publishing of this information is vital to enforcing SLAs.

Marks & Bell(2007b) state that Producers of services should include these elements in their service contracts and SLAs:

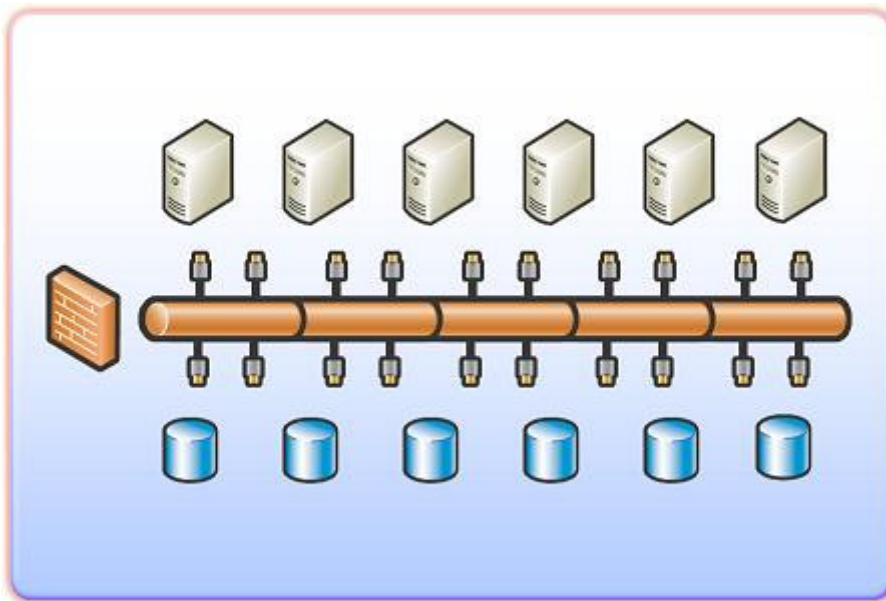
- Consumption limits and ranges. Minimal and maximum allowable transactions and utilization rates
- Reusability and utilization parameters. Reusability factors that are supported by service providers in the agreement, such as interoperability, exposure mechanisms, searchability, and so on
- Guaranteed service performance. Message response time and estimated performance benchmarks
- State management methods. Methods of message and transaction state management, such as stateless and stateful implementations
- Quality assurance. Quality guarantees (e.g., smooth transactions, free of defects, and delays)
- Interface descriptions. Description of exposed interfaces
- Service availability. Hours of operations, and restricted access time

It is through these SLAs and the publishing mechanism where consumers can be charged for their consumption and use of services.

## **2.4 Enterprise Service Bus**

When implementing SOA, an important piece of the infrastructure is the Enterprise Service Bus(ESB). The Enterprise Service Bus is designed to provide

an abstract construct usually based on recognized standards to provide a foundation built on top of the messaging system. The ESB is different from the standard Enterprise Application Integration “hub and spoke” model. The ESB is designed to reduce the amount of point to point connections which will make integration easier to implement and maintain.



**Figure 1 Enterprise Service Bus (Bateman 2005)**

An ESB is not any one product or standard which makes it hard to describe in detail. The ESB provides a central place for messages to flow without having to know the exact location of the services sought. In addition, security and service level monitoring can be implemented at the ESB level rather than at individual services.(Wikipedia, 2007b)

## **2.5 Metadata**

Metadata is defined as “data about data”(Wikipedia, 2007c). Service Oriented Architecture is about defining services and reusing services. Metadata must be stored about the services within the Enterprise. Without the metadata, finding out about existing services becomes very difficult, reduces benefit of reuse, and may lead to more redundant code. With respect to web services, UDDI(Universal Description, Discovery and Integration) is an xml based registry designed for the purpose of storing metadata on services.

The metadata in the repository must be carefully stored to allow services to be discovered and reused. Proper metadata usage is crucial to ensuring a successful SOA over the long term. Discovery of existing services allows reuse to take place and help reduce service functionality overlap or redundancy.

## **2.6 Service Composition and Collaboration**

To increase the reuse, it is desirable for an SOA to allow the construction of new services or work flows by the reuse of existing services. The composition and collaboration between services is called orchestration. Workflow logic and state is better managed within a separate layer rather than embedding this logic within individual components.

Service composition is not limited to only within the organization. The promise of SOA is to allow the reuse of services of within the organization and those which are external. There is a distinction between service collaboration within the organization and those collaborations involving outside parties.

Thomas Erl states that “an orchestration expresses organization-specific business workflow” (Erl, 2005a) whereas a choreography “assumes that there is no single owner of collaboration” (Erl, 2005b). Choreography is cross-organizational. An example of choreography might be a travel agent company booking tours involving the services of an airline company to book flights and a hotel to book a room. Both forms of collaboration, orchestration and choreographies produce reusable services which may then be used in further collaboration.

## **2.7 Defining SOA**

There is much confusion about SOA due to the many different definitions provided by software vendors and experts. Some make the mistake of assuming that if one is using web services that this implies SOA. SOA is currently a very dynamic topic and makes the exact definition difficult to determine. There are numerous definitions that are tailored towards the specific viewpoint that the source is representing.

A look various definitions will allow to get a basic idea of what SOA really is. Most definitions come from technology vendors and authors of SOA literature. There is no one universally accepted definition currently available.

IBM defines SOA as “a business-centric IT architectural approach that supports integrating your business as linked, repeatable business tasks, or services. SOA helps users build composite applications, which are applications

that draw upon functionality from multiple sources within and beyond the enterprise to support horizontal business processes.” (IBM, 2007)

Thomas Erl states that “Contemporary SOA represents an open, agile, extensible, federated, composable architecture comprised of autonomous, QoS-cabable, vendor diverse, interoperable, discoverable, and potentially reusable services, implemented as Web Services.” (Erl, 2005c)

Krafzig, Banke and Slama state that SOA is “a software architecture that is based on the key concepts of an application frontend, service, service repository, and service bus. A service consists of a contract, one or more interfaces, and an implementation” (Krafzig et al., 2005)

Jason Bloomberg and Ronald state “With an enterprise architecture grounded in Service Orientation, we’re looking for a broad set of rules and practices that govern the design and evolution of organizations that leverage business resources as services. We call that set of rules and practices Service-Oriented Architecture” (Bloomberg & Schmelzer, 2006)

SOA is “a conceptual business architecture where business functionality, or application logic, is made available to SOA users, or consumers, as shared, reusable services on an IT network.”(Marks & Bell, 2006c)

The TOGAF SOA working group defines SOA as “an architectural style that supports service orientation. It is a way of thinking in terms of services, service-based development, and the outcomes of services”(The Open Group, 2007a)

Microsoft states that “Service orientation is a means for integrating across diverse systems. Each IT resource, whether an application, system, or trading partner, can be accessed as a service.” (Microsoft, 2006a)

This is by no means a complete list of SOA definitions. There are some common terms used in these definitions but it is quite obvious that these definitions do differ in varying degrees. There is no end to the different SOA definitions that can be found. The key is to look at many definitions and identify the different concepts that make up SOA.

The dominant words found in the previous definitions are services, reusability, discoverability, business functionality, and composition.

Using the intersection of various SOA definitions we come up with a separate definition as follows:

Service Oriented Architecture is the approach of creating discoverable services to handle business logic. These discoverable services should be reusable across the enterprise either individually or in composition with other services.

## **2.8 Conclusion**

Some of the major components of SOA have been introduced. A successful SOA implementation is a combination of many of the previous concepts. An SOA is also a technology independent concept not tied to any specific tool set. It



is an architectural style. This chapter has touched on some of the major components of SOA that are discussed in current literature.

Items such as SOA Governance, metadata, and service level agreements are often driven by the business side which result in corresponding technology side changes for implementation. Other constructs lie more on the technical or technology side and result in helping business accomplish their goals. Individual services, enterprise service bus, and orchestration are derived more on the technical side to help out in business process. It is the bridging of the business side and technology side that help make a Service Oriented Architecture help enable the enterprise better achieve its goals.

Service Oriented Architecture is about the creation of services to help reduce time to market and increase reuse. Both the business and technological concepts presented here help make a successful SOA.

### **3 CHAPTER 3 – Web Service Status**

By far, the most popular and common implementation technology used for implementing an SOA is with web services or the WS specifications. To truly develop an SOA, it requires a standards-based interoperability which must be provided by some technology. Web Services have become very popular with all major IT vendors. They have been working together to develop the specifications and then the products that use them. A requirement for interoperability is vendor collaboration. Without vendors working together to create specifications, interoperability would not be possible. The World Wide Web Consortium(W3C) defines a Web Service as “a software system designed to support interoperable Machine to Machine interaction over a network” (Wikipedia, 2007d). Web Services has developed through the specification of open based standards.

#### **3.1 History of WS**

The Web Service specifications are founded on some basic technology specifications. These basic specifications include XML, SOAP, WSDL, and UDDI. The history of each specification will be given and how they helped form the core specifications of Web Services.

Extensible Markup Language(XML) is the specification that is used as to how to format web service messages. XML is a subset of the SGML(Standard

Generalized Markup Language), is a general text based markup language which allows for the creation of well defined documents with their own tags and structure. XML 1.0 became a W3C Recommendation on February 10, 1998. Some of the goals of this document standard were internet usability, formality and conciseness (Wikipedia, 2007e).

Another specification used to help define web service messages is SOAP(Simple Object Access Protocol). SOAP is a “protocol for exchanging XML-based messages over computer networks, normally using HTTP/HTTPS” (Wikipedia, 2007f) . SOAP was originally designed in 1998 with the backing of Microsoft. SOAP was submitted to the World Wide Web Consortium(W3C) in 2000 as a specification to unify proprietary RPC communication. This protocol is the foundation layer of web services messages. SOAP is currently maintained by the XML Protocol Working Group of the W3C. A SOAP message is made of some individual elements. Firstly, the SOAP message has a required Envelope element which identifies the message as a SOAP message. Header information may be contained in an optional Header element. The required Body element contains call and response information. There is also an optional Fault element which provides information on errors (W3Schools, 2007a).

Publishing a web service’s interface is useful for consumers to see how to talk to an existing web service. It is the Web Service Definition Language(WSDL) that is the specification detailing a web service’s public interface. WSDL is an XML format that describes the details of interacting with a

given web service. The WSDL specification was first submitted to W3C in 2001 and the W3C continues to revise it.

The ability to have a registry or repository to allow services to be discovered by consumers is also very important in regards to SOA. Universal Description Discovery and Integration(UDDI) is an important core specification of web services to help accomplish this. UDDI is a “platform-independent, XML-based registry for businesses worldwide to list themselves on the Internet” (Wikipedia, 2007f). This registry specification is designed to allow business or organizations to publish service definitions to be discovered. This is very important to prevent service redundancy within an organization. Unlike the previous specifications, UDDI was originally developed by UDDI.org and submitted to OASIS(Organization for the Advancement of Structured Information Standards) where this standard is currently.

SOAP, WSDL, and UDDI are considered first generation web service specifications or core specifications.

### **3.2 The WS Specification Process**

The Web Services specification requires a great collaborative effort between vendors. The Web Service specifications are in a constant state of flux. These specifications must truly provide interoperability. The Web Services Interoperability Organization(WS-I) was created in 2002 to help guide this process. WS-I is “an open industry organization that is chartered to promote Web services interoperability across differing platforms, operating

systems/middle-ware, programming languages, and tools” (Weerawarana, Sanjiva, Curbera, Fran, Leymann, Frank, Storey & Tony, Ferguson, Donald F., 2005a).

WS-I’s deliverables include profiles, sample applications, and testing tools. The board of directors is led by members from prominent members such as SAP, BEA, Fujitsu, Hewlett-Packard, Sun, IBM, Intel, Oracle, Microsoft, and webMethods. (Web Services Interoperability Organization, 2007a) A profile consists of “implementation guidelines for how related Web services specifications should be used together for best interoperability” (Web Services Interoperability Organization, 2007b) Sample applications which are provided are to provide examples of applications which are WS-I standards compliant. The testing tools help monitor and analyze message exchanges for conformance to WS-I standards.

Along with a diverse vendor participation within WS-I, the WS-I adopts specification by other organizations such as the Internet Engineering Task Force(IETF), OASIS, UDDI, W3C, and the Open Applications Group(OAGi). WS-I works towards ensuring that the needs of the web services community are met.

The basic core specifications are being extended with further specifications. These web service extension specifications are generally referred to as WS-\*. WS-\* are generally referred to as second-generation specifications. Most of these specifications are prefixed with “WS-“. These specifications are all in various stages of maturity. WS-\* specifications engage a wide variety of issues

including security, transactions, reliability, addressing and orchestration. The WS-\* specifications are maintained by various standards organizations including W3C and OASIS with widespread vendor support and input.

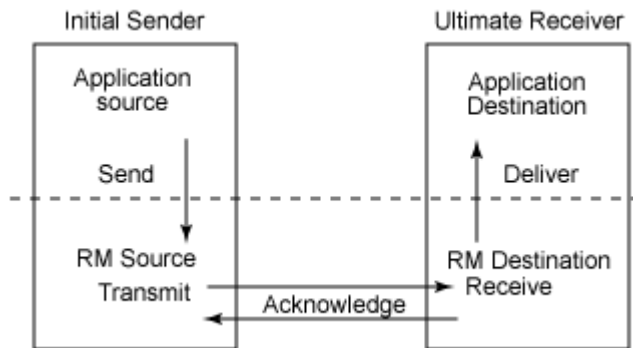
### **3.3 Review of more important WS-Specifications**

There are far too many second-generation specifications to list as they are all in various levels of maturity. There are some more important ones and some of these will be examined. We will only look at four major WS-Specifications: WS-Reliable Messaging, WS-Coordination, WS-Security, and WS-BPEL. This will help illustrate what different specifications are trying to achieve.

#### **3.3.1 WS-Reliable Messaging**

WS-Reliable Messaging is a specification designed to ensure that messages are properly delivered in distributed systems. A sending web service would like to know if its messages successfully arrived, if its message failed to arrive, or whether its messages were received in the correct order.

WS-Reliable Message distinguishes between what initiates a message and what actually performs the transmission. It is the application source which sends the message to the RM(Reliable Messaging) source. The RM Source is responsible for transmitting the message externally. On the other end, it is the RM destination which will receive the transmission and subsequently deliver it to the application destination.



**Figure 2 Reliable Messaging Format (IBM, 2005)**

When considering a set of messages, each message has a unique sequence number called a Sequence, starting with one and increasing monotonically. The receiving endpoint will indicate reception of each message using a SequenceAcknowledgement. Upon receipt of a message the RM destination will issue a sequence acknowledgement. The reliability rules are known as delivery assurances. There are four different kinds:

- **AtMostOnce** – ensures that delivery of one or zero messages are delivered with an error condition happening if more than one is delivered
- **AtLeastOnce** – ensures that a message will be delivered at least once, otherwise resulting in an error condition
- **ExactlyOnce** – ensures that a message will be delivered once and only once with a resulting error condition if not delivered or are delivered multiple times.
- **InOrder** – ensures messages are delivered in sequence otherwise resulting in an error condition

WS-Reliable Messaging has a Sequence element in the SOAP header of the message which contains three child elements: Identifier, MessageNumber, and LastMessage. The Identifier element is the unique identifier of the message set. The MessageNumber is the message position within the set of messages. The LastMessage element is included in the last message only to indicate that the last message has been received. There are further specifics to WS-Reliable Messaging but it is important to note that the Header of the SOAP message is where the WS-Reliable Messaging information is placed.

WS-Reliable Messaging is an important specification in helping dealing with the distributed nature of web services. A web service will greatly improve its quality of service by implementing WS-ReliableMessaging. By having a more reliable message delivery with this specification, enterprises can better support reliable business to business(B2B) exchanges.

### **3.3.2 WS-Coordination**

Another important specifications deals with how different web services can be brought together to make a transaction. WS-Coordination is a general foundation for coordinating outcomes between different Transactions have traditionally possessed 4 qualities: Atomicity, Consistency, Isolation, and Durability (ACID). (Wikipedia, 2007h)



- Atomicity – either all of the tasks are performed in the case of a successful transaction or none are performed
- Consistency – transactions produce consistent results in a legal state
- Isolation – No operation outside the transaction can see the data in an intermediate state
- Durability – once a transaction is successful, the transaction will persist unless a catastrophic failure occur

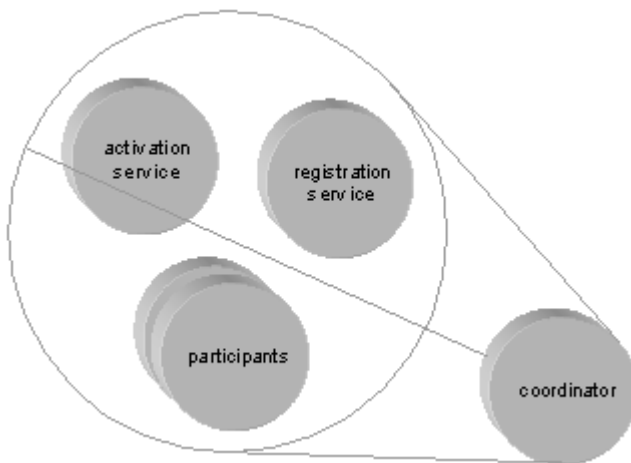
WS-Coordination defines protocols and services that do the following as outlined by (Weerawarana et al., 2005b):

- Provide a context to identify Web Service operations as part of a particular activity
- Allow Web services to register interest in participating in the activity outcome
- Allow the selection of a coordination protocol to be performed between the coordination service and participating Web services at completion of the activity

There are 2 additional specifications that can be used in conjunction with WS-Coordination: WS-Atomic Transaction and WS-BusinessActivity) WS-Atomic Transaction aids in transactions that are typically shortlived using a two phase commit protocol. WS-BusinessActivity aids with longer lived activities and

minimizing latency issues. Both of these specifications may be used in conjunction with one another.

A typical transaction within WS-Coordination will have an Activation Service which will create a new activity and context for the given work scope. The context may contain information such as a unique activity identifier, and expiration value, and coordination type information. The context will be passed along to the web services. When the web services process the CoordinationContext for the first time, they will register with the WS-Coordination's Registration Service. The Registration Service will acknowledge registration and will then return the coordinator's address. The Coordinator Service will perform the transaction via a 2 phase commit.



**Figure 3 WS- Coordination (Erl, 2007b)**

Transactional processing is a very important aspect of distributed computing in the business world. WS-Coordination allows web services to deal with this challenge in a flexible and extensible fashion.

### 3.3.3 WS-Security

Security is a very important topic when it comes to distributed computing. Organizations want to be sure that unauthorized groups are not eavesdropping on messages, tampering with messages, or posing as the organization. Note that security can be implemented at the transport layer, with https, but WS-Security provides end to end security. This is message level security where the security measures are applied directly to the message itself.

An important part of the WS-Security framework is XML-Encryption. The encryption may be applied to portions of the SOAP header or SOAP body. XML-Encryption relies on public/private keys to perform the encryption. The XML-Signature will reside in the SOAP header.

An enterprise should already have security infrastructure in place and WS-Security is designed to augment it and not replace it. WS-Security helps in a couple of ways that the existing security cannot. Firstly, Web Service messages pass through intermediaries which should not be able to see the information. Secondly, Web Services may integrate across multiple systems with different security configurations and WS-Security is a flexible approach to dealing with multiple security models.

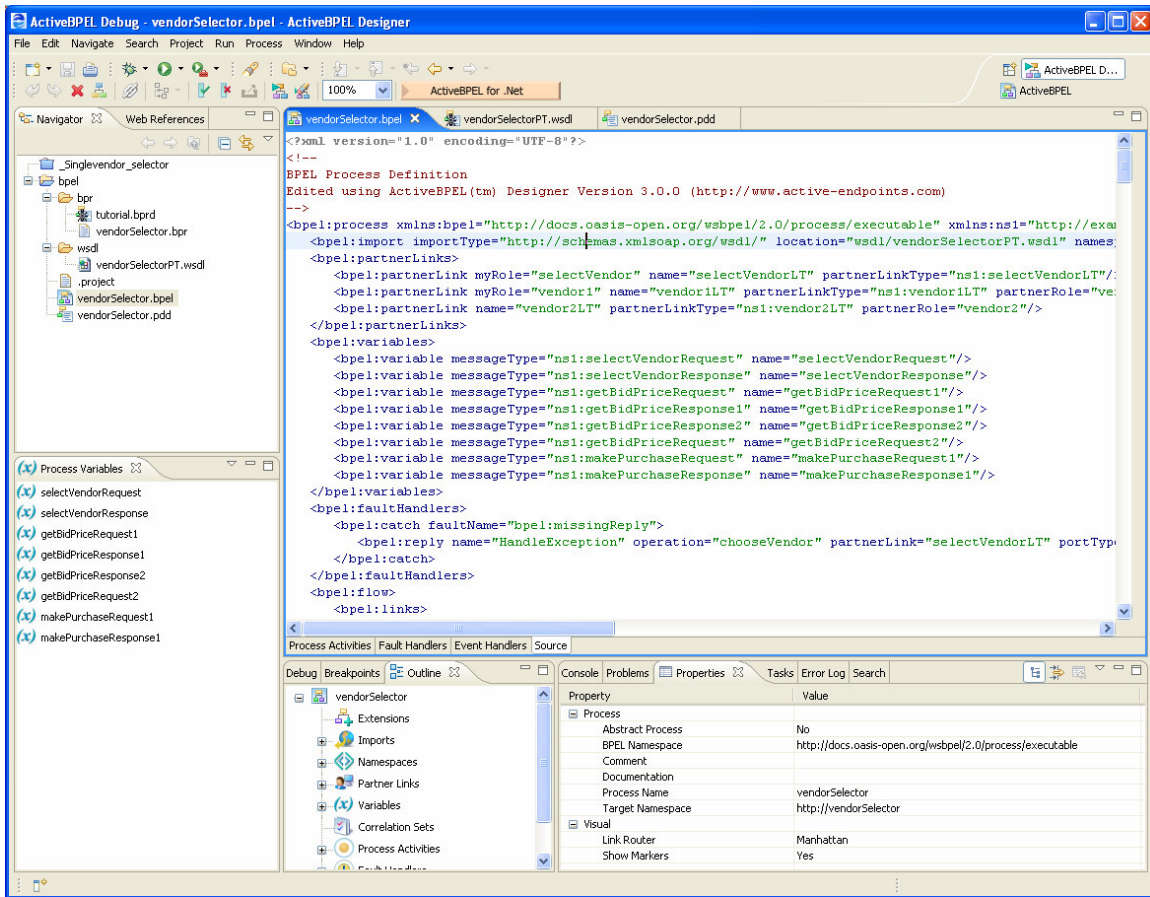
WS-Security is actually a family of specifications that also includes WS-Policy which includes WS-SecureConversation, WS-Federation, WS-Authorization, WS-Policy, WS-Trust and WS-Privacy. These additional specifications can be used in conjunction to meet the security needs of the enterprise.

Security is important to all forms of distributed messaging. It is especially important in communications between different enterprises. WS-Security is one web service extension that helps deal with secure messaging.

#### **3.3.4 WS-BPEL**

WS-BPEL or Business Process Execution is a web service specification that helps define a workflow using web services. Workflow is especially important when orchestrating a new process by reusing existing web services.

BPEL emerged from IBM's Web Service Flow Language(WSFL) and Microsoft's XLANG. It is important to allow different web services to be composed together both within the enterprise and with external partners. As reuse is important in SOA, BPEL workflows handle the stateful conversations and the lifecycle management letting the integrated web services to be relatively stateless. Composition in SOA with web services is the main goal of BPEL. BPEL allows processes to be created with a combination of WSFL's graph oriented style and XLANG's algebraic style.



**Figure 4 Sample BPEL process definition using ActiveBPEL™ Designer**

A BPEL process definition is an xml document. The root element is the process element which provides the name attribute and process definition-related namespaces. The partnerLinks and partnerLink elements define the communication exchanges between the participating web services. The variable elements define variables that can be used to store information retrieved from the participating web services. The invoke element defines how the different partners or participating web services will be called. The receive element defines the information that the BPEL process will require as input. The reply element will define the information that the BPEL process will return.

Logical flow is controlled via the elements sequence, to sequence events, and the switch, case, and otherwise elements which provide conditional logic. Exception handling is handled via the faultHandlers, catch, and catchall elements. A detailed review of BPEL is beyond the scope of this essay. A BPEL process actually composes individual web services into a new web service. BPEL allows the creation of business processes through the reuse of existing web services.

### **3.4 Conclusion**

WS-ReliableMessaging, WS-Coordination, WS-Security, WS-BPEL are some of the more important WS Specifications. There are many more specifications. The number of specifications is constantly changing along with the approval process that these specifications undergo.

The Web services platform provides a standardized framework in which organizations can implement a SOA. Web services provide a framework to provide interoperable messaging, describable and discoverable services, and a mechanism of reuse and composition of services.

The specification process is certainly beneficial for the web service. By standardizing across vendors, it is more likely that the specification will be of higher quality than a vendor specific specification. The specification process helps vet specifications that are not well received. The specification process may be seen as slow and frustrating but this is also an advantage. The advantage is

that the specifications that finally get approved will have had a chance to incorporate feedback along the process.

The future of the Web services framework does look positive due to 4 main factors as stated by Weerawarana et al.:

- “The unprecedented level of vendor support gathered so far, which has resulted in early availability of multiple interoperable implementations.
- The consistent focus (in spite of all the noise of competing standards) on solving the core technical problems, rather than attempting to address every possible requirement.
- The composeability of the specification set, which avoids a monolithic solution and permits flexible use and adaptation of the core specifications.
- A pragmatic specification development process in which technical and interoperability issues are resolved before standardization begins.”(Weerawarana et al., 2005c)

These reasons articulate some of the positive points surrounding the Web Services framework. There has been some concern about IBM and Microsoft exercising patents against some specifications but they have unequivocally stated that they will offer these specifications under royalty-free terms. The challenge for organizations will be dealing with change associated with the changing specifications, technologies and vendor products.

Organizations must be careful with the dynamic nature of the web service specifications as well. As these standards evolve, organizations will have to

adapt to the changes. This is something that must be taken into consideration with during their business and IT processes. This is especially important if the organization's services interact with external organizations as well.



## **4 CHAPTER 4 – EA Frameworks**

Similar to SOA, Enterprise Architecture(EA) can find definitions in a variety of places. EA is about developing a higher level view of the Enterprise as a whole rather than the views found at the project level. This means an architecture that enables a organization to turn their IT into a strategic asset more able to react to business events. Large organizations need a way to manage the inherent complexity of their IT. Enterprise Architecture Frameworks are documented processes that help enable organizations organize their IT assets.

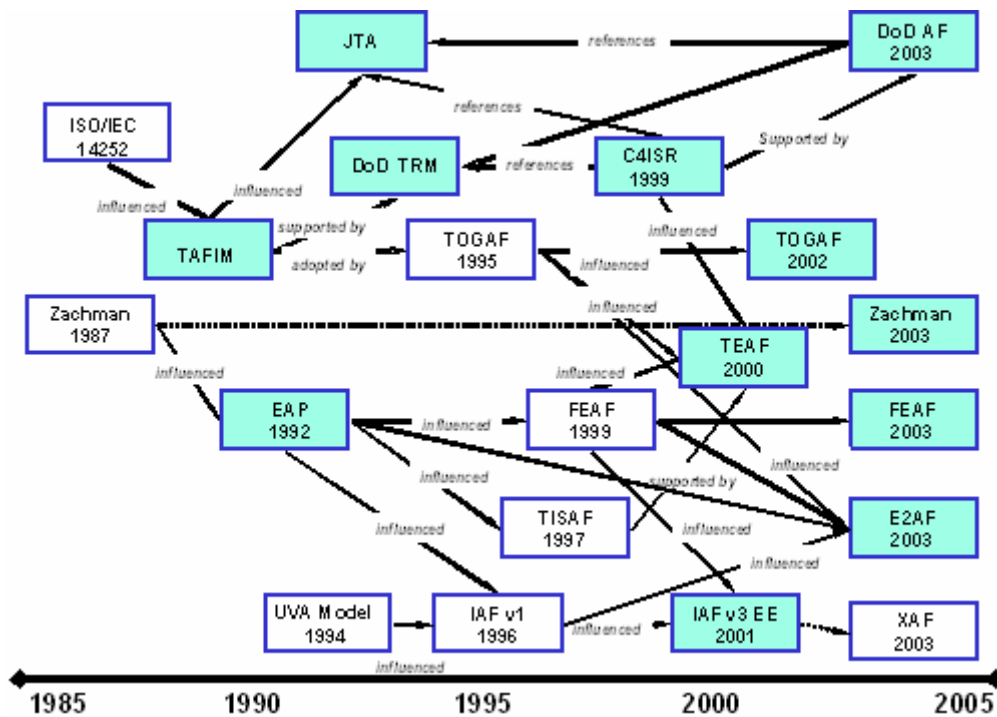
### **4.1 EA Background**

Enterprise Architecture is designed to provide a clearer vision of the organization's current IT assets and business processes. EA will also allow the road map to be developed to reach the future goal state. EA is often compared with an analogy to city planning. With a city, certain zones or neighbourhoods need to be carefully designed with respect to the rest of the city. One would not put all hospitals or police stations only in one neighbourhood. An overall vision for the city is needed to ensure that services benefit maximally. EA allows is designed to let you know where you are with respect to IT assets and current business processes. If one knows where they are, it is a lot easier to know how to get to the desired architecture vision. With EA, the focus is different from IT architecture by being at a higher or broader level. Ross, Weill, and Robertson

elaborate that “enterprise architecture boils down to these two concepts: business process integration and business process standardization...EA is not an IT issue – it’s a business issue.” (Ross, Weill, and Robertson, 2005a). EA helped identify business process issues not visible otherwise. Efficiency comes from digitizing core processes and EA allows analysis on how the business is supported by technology. EA also helps guide the organization to how technology can best support future business direction.

The EA is usually placed under the control of a group that must be involved actively to ensure that projects are aligning with the current EA vision. This is a challenge for organizations, as this is a complex challenge technically with the addition of politics between the various lines of business within the organization.

There are a variety of existing EA Frameworks to choose from. The first EA Framework created was the Zachman Framework in 1987. The following figure shows a basic timeline of EA Frameworks and their relatedness.



**Figure 5 EA Frameworks History (Schekkerman, 2004)**

The 2 most popular EA frameworks available are the Zachman Framework and the TOGAF(Schekkerman, 2004a),(Infosys, 2007). These are the 2 frameworks we will examine in this essay in respects to how they apply to SOA. We will start with a basic overview of them.

## 4.2 Zachman Framework

The concept of EA started with John Zachman’s “A framework for Information Systems Architecture” published in the IBM Systems Journal Volume 26, Issue 3. (Zachman, 1987) . Zachman states that “to keep the business from disintegrating, the concept of information systems architecture is becoming less an option and more a necessity for establishing some order and control in the

investment of information systems resources”(Zachman,1987). Before actually discussing information systems architecture, he draws parallels to the field of classical architecture.

In classical architecture, there are different deliverables or representations that an architect has during the construction of a building. The different representations are bubble charts, architect’s drawings, architect’s plans, contractor’s plans, shop plans, and the building itself. Each representation is a deliverable along the process of designing a building from the most abstract bubble charts to the actual building itself.

Zachman further explains that the models and the different representations can be mapped to cells along an x-y axis. This allows the architecture to be grouped by cells of different classifications. Zachman created the Zachman Institute for Framework Advancement(ZIFA) to further develop the Zachman framework. The Zachman Framework can be represented by a 6 by 6 grid of cells.

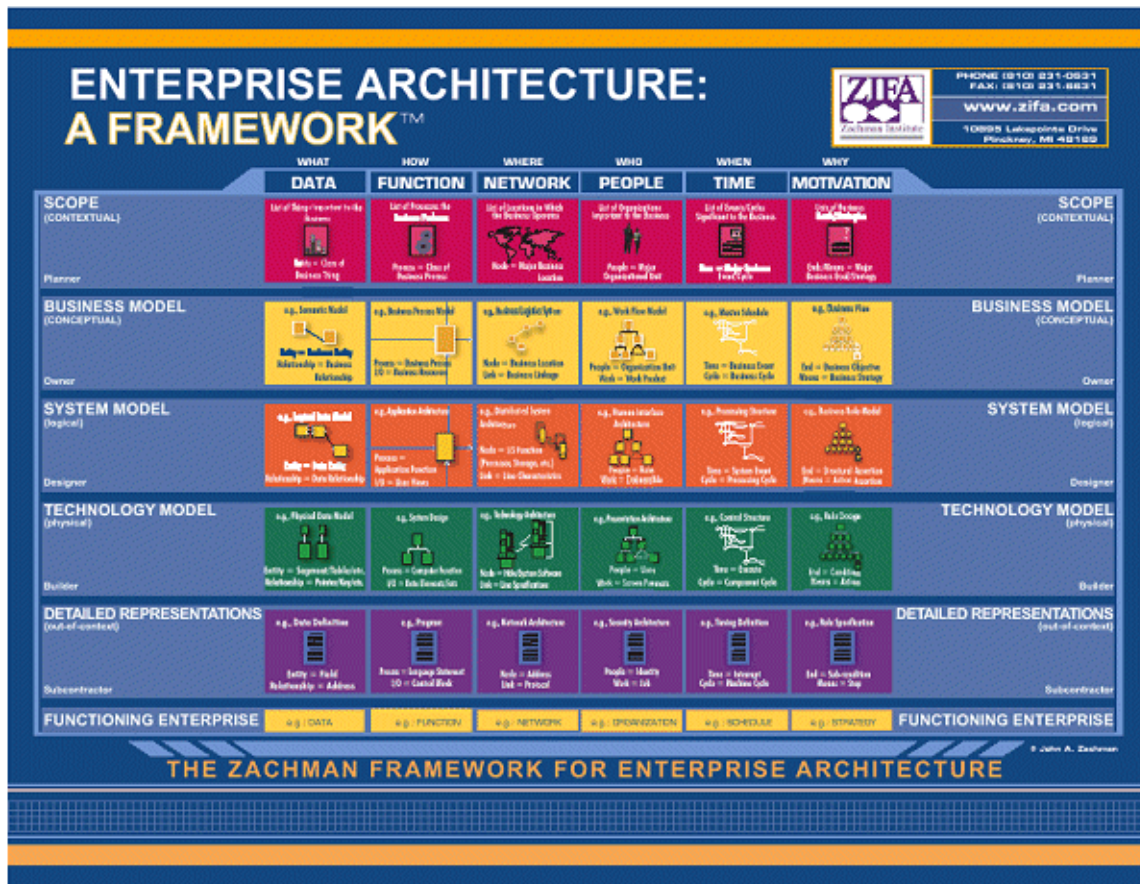


Figure 6 Zachman Framework (Zachman, 2007)

The rows represent principle perspectives: Scope, Business Model, System Model, Technology Model, Detailed Representations, and Functioning Enterprise. The columns are What(Data), How(Function), Where(Network), Who(People), When(Time), and Why(Motivation) aspects. Rows correspond to the different views of the enterprise from very abstract to the actual implementation. At the highest level of abstraction, the Scope view represents an executive summary level description for the various aspects. Business Model row represents a level of detail describing business processes. Each view becomes progressively more detailed until the Detailed Representations view.

The Detailed Representations level should be explicit enough that a contractor could develop the actual software from that view. The sixth row, Functioning Enterprise, represents the software or system itself.

The different aspects or columns represent business process items. What(Data) describes the entities involved in the enterprise. How(Function) describes functions and processes. Where(Network) illustrates geographical locations and interconnections within the enterprise. Who(People) represents the people relationships. When(Time) represents time and events within the enterprise. Why(Motivation) represents the motivations of the enterprise.

The Zachman Framework is a generic classification framework for the Enterprise. Zachman does not prescribe any specific process or tools for completing the mapping or classification process. The Zachman eBook outlines that the goal state is one where time-to-market is reduced and where assemble-to-order is occurring. “Shifting to an assemble-to-order environment. That is, the culture of the assemble-to-order (mass customization) environment is diametrically opposed to the make-to-order (job shop) environment” (Zachman, 2006a). It is much easier and more efficient to develop new products if one assembles from preexisting components. Reusable Parts are the key to quicker time to market. It is interesting to note that the goal of quicker time to market and reusability are shared by both SOA and the Zachman Framework.

Zachman states that there are four items that will not be possible unless investment in Enterprise Architecture is done:

- “Alignment(quality). To ensure that the Enterprise implementations are consistent with the Owner’s intentions
- Integration(seamlessness, interoperability, standard interchangeable parts). To eliminate redundancy, duplication, discontinuity, incoherence, etc.
- Change (flexibility, adaptability). To change the Enterprise with minimum time, disruption and cost.
- Reduced time-to-market. Architecture coupled with an assemble-to-order strategy to reduce the time it takes to produce implementations to only the time it takes to assemble.” (Zachman, 2006b)

In summary, the Zachman framework presents a classification system through different perspectives and aspects to classify artifacts for the enterprise. This EA framework approaches from a taxonomy point of view of classifying current artifacts into cells. This framework is very flexible due to its simplicity and lack of implementation details. It is also interesting to note that reduced time to market and elimination of redundancy are mentioned as items to be achieved with an Enterprise Architecture. These are the main goals of Service Oriented Architecture. Zachman helps ensure that no key aspects are missed in regards to defining the Enterprise Architecture.

### **4.3 TOGAF Framework in Detail**

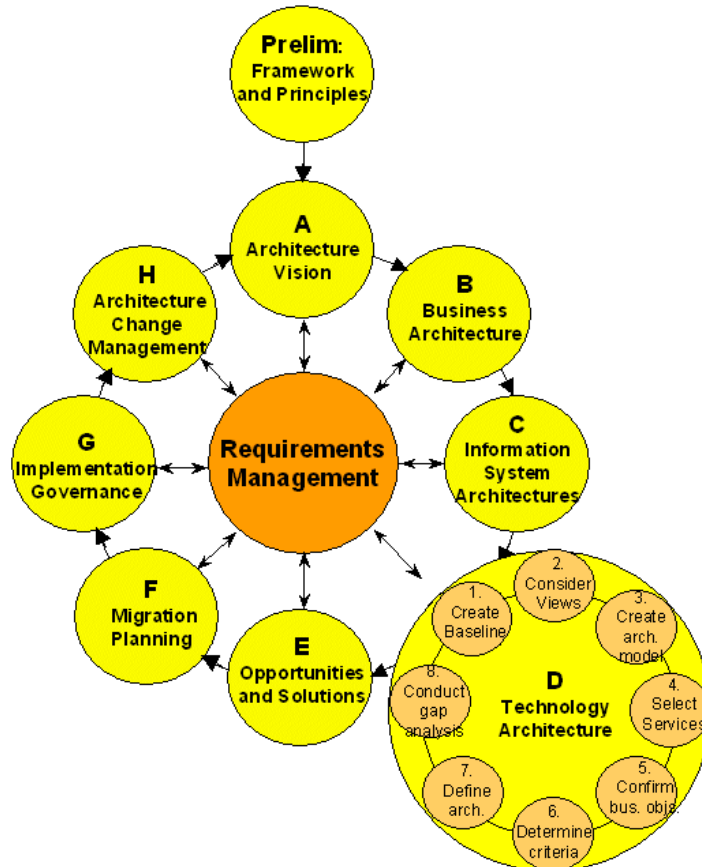
The Open Group Architecture Framework (TOGAF) is an EA Framework which describes a methodical process along with a set of supporting tools. The original version 1 of TOGAF was based on TAFIM(Technical Architecture Framework for Information Management). TAFIM was developed to guide the development of systems within the Department of Defense(DoD).

There are 3 main components of TOGAF:

- Architectural Development Method(ADM) – a process used to derive an Enterprise Architecture for an organization
- Enterprise Continuum – a “virtual repository” for architectural assets of the organization
- TOGAF Resource Base – a set of resources, including guidelines, templates, and background information to aid in the ADM



#### 4.3.1 Architectural Development Method(ADM)



**Figure 7 Architectural Development Method (Innenministerium des LNW, 2008)**

The ADM helps describe how to develop an Enterprise Architecture through the examination of business requirements. There are nine main areas to help define the Enterprise Architecture:

- Preliminary Phase – define “how we do architecture” by ensuring commitment of those involved, defining methodology and principles
- A) Architecture Vision – define scope, stakeholders, and vision

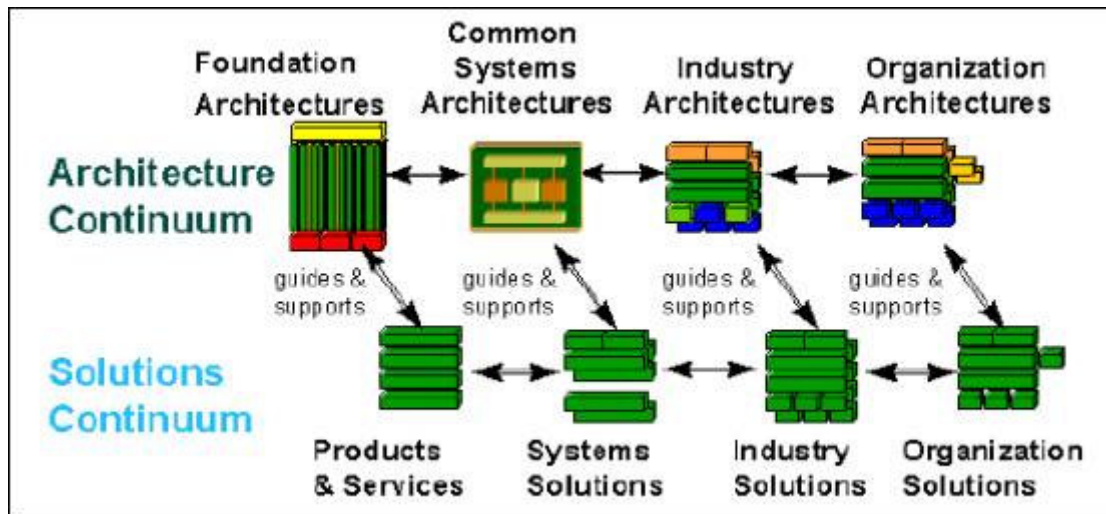
- B) Business Architecture – defining the Business Architecture including the Baseline Business Architecture and Target Business Architecture
- C) Information Systems Architecture – develop Target Architectures
  - Data Architecture – define major types and sources of data to support the business
  - Applications Architecture – define application systems to process data and support the business
- D) Technology Architecture – define current approved technology, products and vendors
- E) Opportunities and Solutions –implementing new solutions within the frameworks guidelines from previous steps
- F) Migration Planning – planning and prioritizing processes or technology
- G) Implementation Governance – ensuring the various IT implementations and the architecture are aligned
- H) Architecture Change Management – ensure changes in architecture are dealt with in a cohesive manner

Requirements Management – managing the requirements throughout all the ADM phases

#### **4.3.2 Enterprise Continuum**

The Enterprise Continuum can be thought of as a “virtual repository” of all the architecture assets including models, patterns and architecture descriptions.

There are two main concepts within the Enterprise Continuum: The Architecture Continuum and The Solutions Continuum.



**Figure 8 Architecture & Solutions Continuum (Open Group, 2007c)**

The Architecture Continuum “offers a consistent way to define and understand the generic rules, representations, and relationships in an information system” (The Open Group, 2006a) The bidirectional arrows along the top row indicate increasing levels of detail from left to right. A Foundation Architecture represents an architecture of building blocks and corresponding standards. A Common Systems Architectures include items such as a Security Architecture or a Network Architecture. An Industry Architecture represents an industry specific component such as a data model shared within the retail industry. Organization or Enterprise Architectures represent components that have been written to constitute solutions for a particular enterprise or organization.

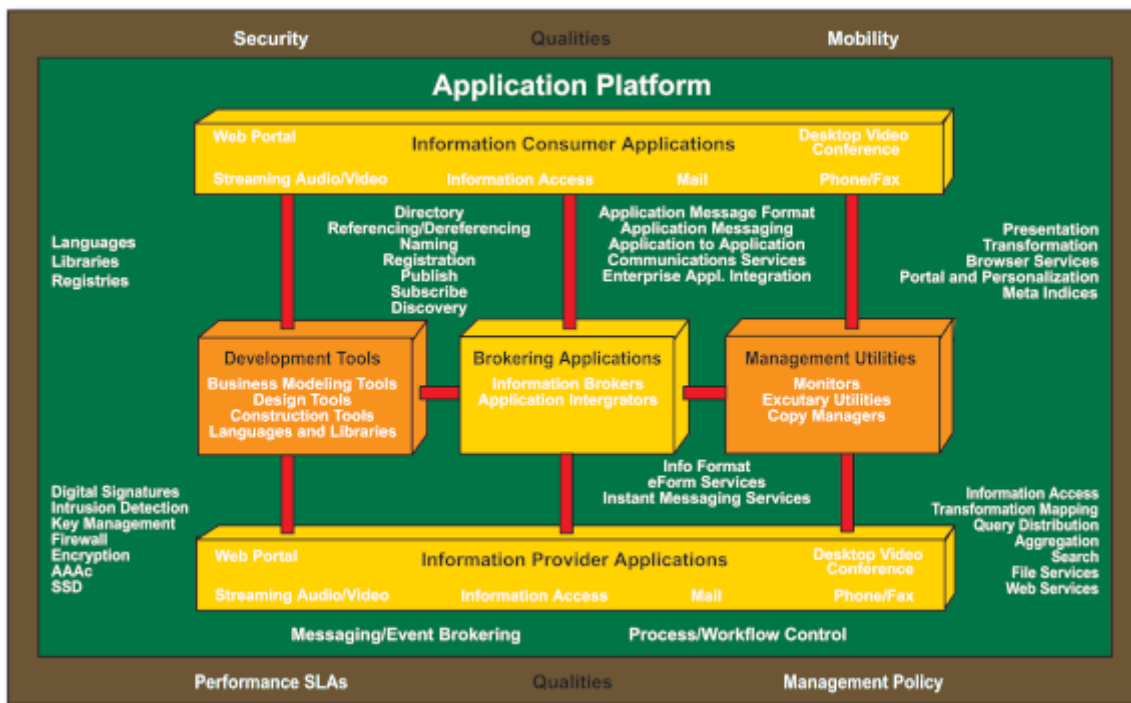
The Solutions Continuum represent the implementations of the various architectures along the Architectures Continuum. The Solutions Continuum can be seen as a solutions inventory or re-use library. Products and Services include procurable products such as hardware and software, along with professional services. Systems Solutions is an implementation of the Common Systems Architecture which may be certified or branded. Industry Solutions is an implementation of an Industry Architecture. Organization or Enterprise Solutions are an implementation of an Organization Architecture.

The TOGAF Standards Information Base (SIB) is “a database of facts and guidance about information systems standards” (The Open Group, 2006b). These may include standards from organizations such as ISO, IEE, W3C, etc. The SIB is used to document architectural standards, assure conformance, and provide information on relevant IT standards.

The TOGAF Integrated Information Infrastructure Reference Model (III-RM) is designed to help in “addressing one of the key challenges facing the enterprise architect today: the need to design an integrated information infrastructure to enable Boundaryless Information Flow” (The Open Group, 2006c). It is Jack Welch, former CEO of General Electric, who coined the term “Boundaryless Organization”. This refers to the fact that information should flow throughout the organization to make it more flexible, innovative and competitive.

III-RM has two main components:

- “1. A taxonomy, which defines terminology, and provides a coherent description of the components and conceptual structure of an integrated information infrastructure
2. An associated III-RM graphic, which provides a visual representation of the taxonomy and the inter-relationship of the components, as an aide to understanding”(The Open Group, 2006d)



**Figure 9 Integrated Information Infrastructure Reference Model (Open Group 2007d)**

The III-RM helps define components and their interrelationships.

#### 4.3.3 TOGAF Resource Base

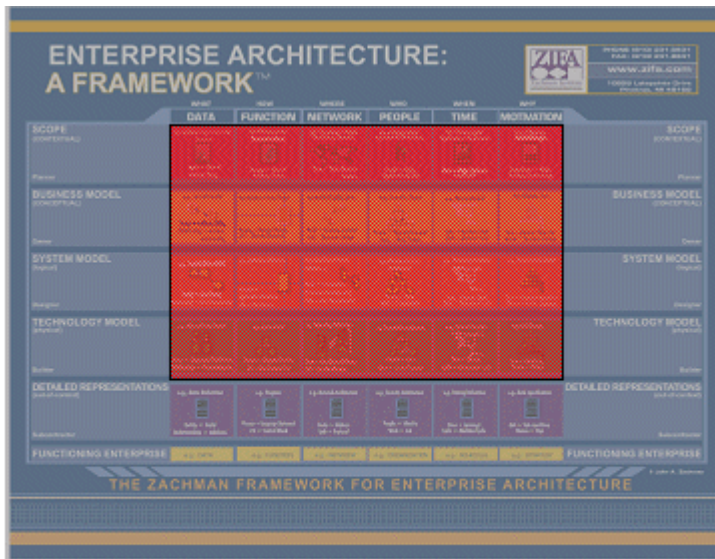
The TOGAF Resource Base is a collection of template examples and guidelines. There is a chapter on the creation of an Architecture Board to prevent on-off solutions and unconstrained development. Architecture Compliance is


covered to ensure that individual projects comply with the Enterprise Architecture. Guidelines are presented in regards to Architecture Governance in respect to Corporate, Technology, IT and Architecture Governance. A detailed review of the Resource Base is beyond the scope of this essay.

#### **4.4 Comparison of Zachman and TOGAF**

Both Zachman and TOGAF are classified as EA Frameworks which help address system complexity and the alignment of business and IT. Roger Sessions's compared the top 4 EA Frameworks in (Sessions, 2007). In this article he describes the Zachman Framework more of a taxonomy or classification of architectural assets. There is not much help in how to go about classifying the artifacts into the Zachman Framework as there is no process or instructions to go about doing this.

TOGAF is built to map onto other EA Frameworks, including the Zachman Framework. The Zachman framework is generic as "it does not prescribe or describe any particular method, representation technique, or automated tool" (The Open Group, 2006g). The open group mentions that "the scope of the four architecture domains of TOGAF align very well with the first four rows of the Zachman Framework"(The Open Group, 2006h)



 Scope of TOGAF ADM

**Figure 10 Scope of TOGAF onto the Zachman Framework (Open Group, 2007e)**

The Open Group maps out how to use the ADM to populate the Zachman Framework in various steps, from the preliminary phase and Phases A, B, C, and D. The Architecture Development Method(ADM) is generally regarded as the strength of the TOGAF EA framework. The combination of the strengths of differing EA Frameworks could be of better value for certain organizations.

The TOGAF Framework is described as a process to generate an Enterprise Architecture. The three main components of the TOGAF help an organization define their Enterprise Architecture. The ADM process may be used to incrementally define the baseline architecture and how to achieve the target architecture. The Enterprise Continuum is a repository for storing the

architectural assets. The TOGAF Resource Base provides templates and guidelines to help in the process.

There is nothing prohibiting the use of more than one EA Framework. The Open Group even lays out steps on mapping the TOGAF ADM towards the Zachman Framework. TOGAF provides more detailed steps with a set out process with templates where the Zachman Framework does not provide any implementation details.

#### **4.5 Conclusion**

EA Frameworks are designed to better coordinate IT and business and help the enterprise become more agile. Different frameworks will better suit different organizations. There will have to be a review of the different candidates and how they will benefit the organization. Different EA Frameworks may be used to complement each other in the organization's quest for the best Enterprise Architecture.

Making a better fit between business process and IT process is the key to good Enterprise Architecture. EA Frameworks help organizations define their IT processes to help make the organization's business processes better succeed. These frameworks are usually very flexible. This flexibility is required as organizations differ greatly on so many levels. The two most popular EA Frameworks, Zachman and TOGAF, can help an organization define the current Enterprise Architecture as well as map the future EA directions.



## **5 CHAPTER 5 – EA & SOA Together**

### **5.1 Why EA Frameworks and SOA are a good match**

Enterprise Architecture and Service Oriented Architecture take similar stances to viewing Enterprise IT assets. Both require an enterprise view of the organization to streamline any individual business unit projects to be better aligned. There are SOA architects who are turning to EA Frameworks to help them with their SOA implementations.

The Enterprise Architecture Frameworks help guide the architect to a higher, more abstract view of the enterprise's IT assets and business processes. There are common themes between both the EA world and SOA world. The following chapter will look at how the Zachman and TOGAF frameworks being adapted in the implementation of SOA.

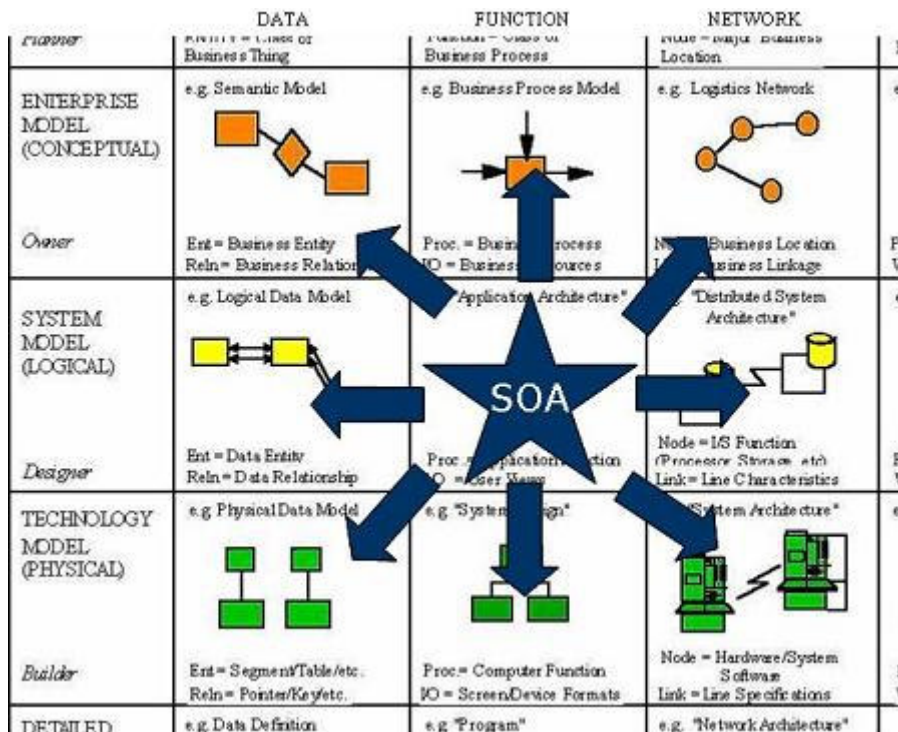
### **5.2 Review of the Zachman Framework towards SOA**

The most popular EA Framework is the Zachman Framework. Enterprise Architecture helps look at IT from an Enterprise level rising above project or line of business views. SOA can benefit by classifying existing IT assets and business processes together. The Zachman Framework, as previously outlined, is presented in two dimensions to map views of the Enterprise. SOA shares similar goals to EA in respects to different views to identify enterprise concerns

and interests. Reuse & consolidation through some form of governance are made easier through this visualization. The Zachman Framework provides this matrix to help visualize the Enterprise as a whole.

Zapthink, an IT advisory firm, has published some thoughts on how the Zachman Framework can map towards SOA. They state that the Zachman Framework “helps companies organize and prioritize the various perspectives of EA, and this organization applies just as well when the EA is SOA” (Schmelzer, 2006). Due to the similar goals of EA and SOA, it can be seen that the use of an existing framework like Zachman can provide value towards SOA.

In mapping this EA framework to SOA, Zapthink outlines that the understanding to be gained among the various Zachman perspectives is the key strength of the framework. They go on to mention that the Zachman framework must be tailored to the specifics of SOA.



**Figure 11 Mapping SOA to Zachman (Schmelzer,2006)**

Zapthink mentions that the “logical starting point for applying the Framework to SOA is the Application Architecture portion at the intersection of the “Function” column and “Logical System Model” row” (Schmelzer, 2006) This is the logical starting point for services which “tackle the problems of integration, asset reuse, and loose coupling of systems” (Schmelzer, 2006) The services may be mapped into this cell. The adjoining 8 cells then may be described in further iterations. The Zachman framework does not explicitly provide the process of populating the individual cells. The Zachman eBook outlines that slivers of the enterprise may be documented, the same may be done with the services and its affected assets including business processes, networks, and data. These slivers may be either vertical or horizontal in respect to the

Zachman Framework. The vertical sliver relates to different levels of detail of an application. The horizontal sliver relates to a certain level of detail across the enterprise. Zapthink states “For SOA to be the architecture that enables such broad agility and change in organizations, it’s essential for architects to eventually apply the entire Zachman Framework to SOA” (Schmelzer, 2006) .

It is also pointed out that what is provided is that “the Zachman Framework shows that an EA perspective on SOA broadens the reach of SOA and doesn’t require us to consider an overly narrow view of SOA” (Schmelzer, 2006). SOA is often a confusing world of buzzwords, technology and inconsistencies. Zapthink further articulates that the Zachman Framework can help the architects develop techniques of organizing and identifying relationships to better “respond to change, and leverage change for competitive advantage” (Schmelzer, 2006). Simplicity is key for the Zachman framework. From a classification process, the Zachman framework allows one to see relationships emerge among identified assets. The cells which are not populated are gaps to be filled. The organization can populate the cells with known artifacts to create a better picture of what the current architecture is.

### **5.3 Review of TOGAF towards SOA**

The Open Group has officially been trying to incorporate SOA as part of its TOGAF Framework. The SOA Working Group, formed in October 2005, was set up to facilitate SOA and to “to enable a trained TOGAF practitioner to use

TOGAF 8 and the SOA adjustments 'out of the box' to develop a service-oriented architecture.” (The Open Group, 2007b) This is not an easy task given that SOA is such a hard area to create SOA specifications that everyone will agree to.

The Open Group’s initial SOA work involved the setup of 3 initial projects:

- Definition of SOA
- SOA Case Studies
- Evaluation of the Value that The Open Group can add to the Evolution of SOA

The first project defined SOA as “an architectural style that supports service orientation. It is a way of thinking in terms of services, service-based development, and the outcomes of services”(The Open Group, 2006e).

The SOA Case Studies produced a set of SOA Case studies to help make SOA more understandable. It must be noted that these are SOA Case studies alone and not case studies reflecting TOGAF & SOA together.

The last project led to recommendations where the Open Group could add value to the Evolution of SOA. These recommendations led to the formation of a number of new projects. Some of the SOA Working Group Projects include Ontologies for SOA, SOA Governance, SOA/TOGAF Practical Guide, Service-Oriented Infrastructure, SOA Reference Architecture, SOA and Security and The SOA Maturity Model Project.

"SOA is an emerging architectural style that needs to be supported by TOGAF," said Dr. Chris Harding, forum director for SOA and semantic interoperability at The Open Group. "The SOA/TOGAF Practical Guide Project will make a major contribution to the industry-wide understanding of SOA and the use of TOGAF to better align business with IT." (The Open Group, 2006f)

In July 2007, the Open Group SOA Working Group published a white paper to "create a high-level understanding of SOA and its relation to enterprise architecture, and in particular to TOGAF" (The Open Group, 2007a). This paper is a starting point for incorporating SOA into TOGAF explicitly.

This white paper defines SOA as "an architectural style that has recently come to prominence" (The Open Group, 2007a). This guide is designed to support EA architects in their understanding of Enterprise Architecture and SOA. This white paper lays out:

- its definition of the nature of SOA within a typical enterprise
- a discussion of changes within TOGAF's Architectural Development Method(ADM)

Firstly, The Open Group's vision is that of "Boundaryless Information Flow". This vision entails a need to "enable information to flow freely across the permeable organizational boundaries"(The Open Group, 2007a). A service is defined as "a logical representation of a repeatable business activity that has a specified outcome, such as "check customer credit", "provide weather data", or

“consolidate drilling reports”. It is self-contained, may be composed of other services, and is a “black box” to its consumers” (The Open Group, 2007a).

They outline that the applications will be replaced by interacting services. These services will exchange messages an Enterprise Service Bus (ESB). It is stated that “A major benefit of SOA is that it delivers enterprise agility, by enabling rapid development and modification of the software that supports the business processes” (The Open Group, 2007a).

A distinction is drawn between SOA and software architecture in that the building blocks of SOA are loosely coupled services rather than subroutines or scripts calling each other directly. Definitions of the various elements of SOA are necessary to define approaches to implementing an SOA. This is one of the steps required for an EA framework to be useful in providing an adequate methodology.

Secondly, this white paper describes how TOGAF can be applied to SOA, by a focus on the eight iterative phases of the TOGAF ADM. This paper will only outline some of the changes provided to these phases:

- Architecture Vision – the general approach is not changed
- Business Architecture – 2 major differences
  - An SOA project, it will be natural and desirable to describe the business operations as services
  - This phase may require describing new business operations and methods for future use, aiding in SOA business agility

- Information Systems Architecture –
  - Must consider how services will interact with applications that will not be converted (Legacy)
- Technology Architecture –
  - Specification of SOA specific infrastructure (service bus, registry, etc.) will be required
  - Specification of tools to support the methods of phases B & C
- Opportunities and Solutions – SOA has little impact
- Migration Planning – SOA has little impact
- Governance –
  - “focus on governing the service lifecycle, supporting service infrastructure, and compliance with the SOA of the organization” (The Open Group, 2007a).
- Architecture Change Management –
  - Changes to services should not be considered architecture change and can be covered by Implementation Governance

It is no simple task to attempt to incorporate SOA into an existing EA framework. The Open Group has started to attempt to fold SOA into EA. This will hopefully help clarify SOA’s relationship to EA more clearly. The Open Group is clearly concerned about this relationship by the published white paper. The SOA group is continually providing updates on its progress with SOA. This is a slow process as gaining consensus among the members can be challenging.



## 5.4 Conclusion

There are many EA Frameworks to choose from to help in the SOA process. Zachman and TOGAF are currently the most common ones to choose from. Existing frameworks can provide a ready toolset and must be evaluated for a best fit for the organization.

Roger Sessions, did a comparison of 4 EA Frameworks, Zachman, TOGAF, FEA and the Gartner Framework. (Sessions, 2007) In summary, the Zachman framework can often be thought of more of a taxonomy rather and a framework. “Zachman tells you how to categorize your artifacts. TOGAF gives you a process for creating them” (Sessions, 2007). This is the essential difference between both frameworks. This difference also makes them complementary as well.

Sessions goes on to say that choosing between these frameworks can be difficult and “for many organizations, the best choice is all of these methodologies, blended together in a way that works well within the organization’s constraints” (Sessions, 2007). The job of an EA Framework should be to deliver real business value as quick as possible.

“Benefits to be gained from Enterprise Architecture are listed as

- Improvements in using IT to drive business adaptability.
- Closer partnership between business and IT groups.
- Improved focus on organizational goals.
- Improved morale, as more individuals see a direct correlation between their work and the organization's success.

- Reduced numbers of failed IT systems.
- Reduced complexity of existing IT systems.
- Improved agility of new IT systems.
- Closer alignment between IT deliverables and business requirements.”

(Sessions, 2007)

The goals of Enterprise Architecture and Service Oriented Architecture are very similar and it can be seen how the use of an EA Framework or some combination of EA Frameworks can be used to support SOA as well. The loosely coupled services and governance required by SOA can benefit from a technology independent methodology as offered by the EA frameworks.

Chris Harding, who heads the TOGAF SOA Working group states “ By working together, the proponents of SOA and the traditional enterprise architecture community can smooth the evolution of the practice, and help SOA deliver its revolution, so that the organizations that use SOA can deliver better business value.” (Harding, 2007).

The application of EA towards SOA is not hard to envision. “In five years I don’t think there will be SOA ... it’s all going to fold back into enterprise architecture,’ said Dave Linthicum, CEO of Linthicum Group, in a keynote address today at The Open Group’s Enterprise Architecture Practitioner’s Conference in Austin, Texas. “SOA is a subpattern of EA.”(Gardner, 2007)

As illustrated, there are people using the frameworks towards SOA implementation and it is a best fit to the organization. Challenges lie in not only the style of architecture but also in the changing technologies from which the

services are built. An EA Framework can provide a consistent methodology to see an SOA evolve within an organization through business process and technology changes. The strength of EA Frameworks strengths lie in that they are not tied to specific technologies. They structure IT solutions to business problems regardless of technology. This allows the organization to better separate themselves from specific technology specific details when developing their SOA. Longevity of solutions is achieved through this architectural backbone.

EA “provides a basis for service orientation through a top-down approach” (Banerjee, J. & Aziz, S.,2007). Philip Allega contends that many Service Oriented Architecture endeavors “have been project-specific, not guided by enterprise architecture (EA)” (Allega, 2004) but states that additional investment in SOA should be guided by EA. To do this project teams should interact with the EA teams to ensure that some suitable balance between project specific goals and enterprise architecture goals is met.

Trying to do SOA without EA, will usually mean that artifacts of EA will exist but not be properly recorded and more importantly the SOA will develop with “no clear planning process for IT architecture” (Banerjee, J. & Aziz, S.,2007a).

SOA adoption without an EA program “is often “technology” focused and restricted to individual projects” (Banerjee, J. & Aziz, S.,2007a). SOA must be adopted across projects to be successful. “Without the right focus, SOA adoption can easily lead to an application portfolio in which all applications/services

depend on each other in enterprise-level spaghetti architecture” (Banerjee, J. & Aziz, S.,2007a). EA is to provide the focus necessary for a more successful SOA implementation. The use of an EA framework helps an SOA implementation.

## **6 CHAPTER 6 – Recommendations & Conclusions**

### **6.1 Goals of EA and SOA**

There are many goals shared between EA and SOA. Some of these include reduced time to market, increase code reuse, business agility and alignment of business processes to IT. SOA is a newer concept when compared to Enterprise Architecture. The SOA and EA share similar goals and do fit together in some fashion. Both are relatively young and evolving which presents an opportunity for their use together.

SOA is being developed along with Web Services specifications. The Web Services specifications are still being worked on and developed. EA Frameworks provide structure in achieving the goals of SOA. This is due to the fact that an SOA requires a vision of the Enterprise as a whole. EA Frameworks provide this.

### **6.2 How EA Frameworks help SOA**

The EA Frameworks help by framing IT assets within the enterprise. The services created within the SOA fall within the Enterprise Architecture as reusable and discoverable components. I tend to regard SOA as a concern within the Enterprise Architecture. This means that if you can address your Enterprise Architecture, the SOA concerns will be addressed within it.

SOA is an architectural style, not unlike procedural programming or object oriented programming. Enterprise Architecture Frameworks allow an enterprise to develop process and views which are separated from the exact technologies. EA Frameworks are abstract to the point of being insulated from technology, architectural and business changes. This is important for the goal of SOA implementation as this field is currently very dynamic. The changes occur very quickly within the different components including SOA components, Web Service specifications or vendor products. Enterprise Architecture Frameworks can capture the Enterprise business processes and IT processes and minimize the ripple affect of these other changing pieces. Technologies and architectural styles will change. Since SOA is an architectural style, it makes most sense to document the Enterprise IT in terms of the most abstract form. This is Enterprise Architecture defined by an EA Framework.

### **6.3 Caveats of the EA Frameworks**

Every organization is different. Each EA Framework brings with it different strengths and weaknesses. The Zachman Framework is especially strong when mapping or classifying the IT artifacts. It is not very strong when it comes do defining a specific process to implement it. TOGAF has its very strong process oriented approach to defining the Enterprise Architecture with it's Architectural Definition Methodology. Being more process oriented, it may involve much more

structure than the Zachman Framework. TOGAF provides many tools that can be taken as IT leaders see benefit.

These are choices to be made carefully for the given enterprise. There is nothing against implementing more than one framework or a combination of individual parts of frameworks. It is up to the IT leaders to decide what will work for the given organization in implementing the SOA.

#### **6.4 Conclusion**

The style of Service Oriented Architecture is designed to increase agility for the organization through the use of reusable services. Services must be designed correctly in accordance to the needs of the business as a whole. Often projects are based at the line of business level without taking into account the enterprise's needs as a whole. To eliminate future rework, it is valuable to consider the enterprise's needs as a whole to make IT more valuable to the business.

It is an enterprise view that is required to help SOA succeed. EA Frameworks help clarify the organizations current state, future state and road map by examining current assets and processes. The use of EA Frameworks means that SOA services can be created in a manner that best benefits the organization's business or long term goals. Zachman and TOGAF are both being used to help out with SOA in the industry. TOGAF is being developed specifically to support SOA. EA Frameworks help provide architectural tools for

classification or process development to give SOA increased support to be more successful.



## 7 REFERENCES

- Allega, Philip. (2004). EA and SOA: Balancing project implementation and top-down guidance. Retrieved November 23, 2007 from <http://www.itweb.co.za/sections/business/2004/0409300600.asp?A=BUS&S=Business&T=News&O=C>
- Banerjee, J. & Aziz, S.. (2007). SOA: The missing link between Enterprise Architecture and Solution Architecture. Retrieved December 21, 2007 from <http://www.infosys.com/IT-services/systems-integration/white-papers/SOA-link-between-EA-SOLA.pdf>
- Bateman Adrian. (2005). WS-\* vs. Enterprise Service Bus (ESB). Retrieved January 3, 2008 from <http://adrianba.net/archive/ecad356e8d97482c8c891e083cf07b6c.aspx>
- Bloomberg, Jason & Schmelzer, Ronald (2006). Service Orient or Be Doomed! – How Service Orientation Will Change Your Business. Hobonoken, New Jersey: John Wiley & Sons, Inc.(pp. 126)
- Brown, Paul C. (2007). Succeeding With SOA – Realizing Business Value Through Total Architecture. Boston: Addison Wesley. (pp. 217 – 226)
- Erl, Thomas. (2007a). SOA – *Principles of Service Design*. Boston: Prentice Hall
- Erl, Thomas. (2007b). The WS-Coordination Context Management Framework. Retrieved December 14, 2007 from <http://www.ws-standards.com/ws-coordination.asp>
- Erl, Thomas. (2005a). Service-Oriented Architecture – Concepts, Technology, and Design. Crawfordsville, Indiana: Prentice Hall(pp. 211)
- Erl, Thomas. (2005b). Service-Oriented Architecture – Concepts, Technology, and Design. Crawfordsville, Indiana: Prentice Hall(pp. 212)
- Erl, Thomas. (2005c). Service-Oriented Architecture – Concepts, Technology, and Design. Crawfordsville, Indiana: Prentice Hall(pp. 54)
- Gardner, Dana. (2007). SOA will fold into general enterprise architecture, Linthicum tells Open Group crowd. Retrieved December 23, 2007 from <http://blogs.zdnet.com/Gardner/?p=2506>

- Harding, Chris. (2007). Enterprise Architecture and SOA: Two tribes. Retrieved October 31, 2007 from [http://www.cioupdate.com/reports/article.php/11050\\_3695556\\_3](http://www.cioupdate.com/reports/article.php/11050_3695556_3)
- IBM. (2005). Web Services Reliable Messaging reloaded. Retrieved November 29, 2007 from <http://www.ibm.com/developerworks/webservices/library/ws-rmreload/>
- IBM. (2007). Service Oriented Architecture. Retrieved December 15, 2007 from <http://www-306.ibm.com/software/solutions/soa/>
- Infosys. (2007). Enterprise Architecture Is Maturing. Retrieved December 12, 2007 from <http://www.infosys.com/cio/ea-maturing.pdf>
- Innenministerium des Lands Nordrhein-Westfalen. (2008). Vorgehensmodell nach TOGAF. Retrieved December 16, 2007 from <http://www.im.nrw.de/inn/446.htm>
- Krafzig, Dirk, Banke, Karl & Slama, Dirk (2005). Enterprise SOA – Service-Oriented Architecture Best Practices. Crawfordsville, Indiana: Pearson Education Inc.. (pp. 57)
- Manes, Anne Thomas. (2005). The Elephant Has Left the Building. Retrieved January 24, 2008 from <http://www.intelligententerprise.com/showArticle.jhtml?articleID=164301126&pgno=3>
- Marks, Eric A. & Bell, Michael. (2006a). Service-Oriented Architecture – A Planning And Implementation Guide For Business And Technology. Hobonoken, New Jersey: John Wiley & Sons, Inc.(pp. 248 – 249)
- Marks, Eric A. & Bell, Michael. (2006b). Service-Oriented Architecture – A Planning And Implementation Guide For Business And Technology. Hobonoken, New Jersey: John Wiley & Sons, Inc.(pp. 238)
- Marks, Eric A. & Bell, Michael. (2006c). Service-Oriented Architecture – A Planning And Implementation Guide For Business And Technology. Hobonoken, New Jersey: John Wiley & Sons, Inc.(pp. 1)
- Microsoft. (2006a). Service Oriented Architecture Overview. Retrieved December 11, 2007 from <http://www.microsoft.com/biztalk/solutions/soa/overview.msp#EFB>
- Oasis. (2007). OASIS. Retrieved December 1, 2007 from <http://www.oasis-open.org/>
- The Open Group. (2006a). TOGAF 2006 Edition (Incorporating 8.1.1) . US : Open Group. (pp. 129)

- The Open Group. (2006b). TOGAF 2006 Edition (Incorporating 8.1.1) . US : Open Group. (pp. 174)
- The Open Group. (2006c). TOGAF 2006 Edition (Incorporating 8.1.1) . US : Open Group. (pp. 185)
- The Open Group. (2006d). TOGAF 2006 Edition (Incorporating 8.1.1) . US : Open Group. (pp. 186)
- The Open Group. (2006e). Definition of SOA. Retrieved November 21, 2007 from <http://www.opengroup.org/projects/soa/doc.tpl?gdid=10632>
- The Open Group. (2006f). Definition of SOA. Retrieved December 12, 2007 from <http://www.cyperus.com/cgi-bin/stories.pl?ACCT=104&STORY=/www/story/10-30-2006/0004462133&EDATE>
- The Open Group. (2006g). TOGAF 2006 Edition (Incorporating 8.1.1) . US : Open Group. (pp. 475)
- The Open Group. (2006h). TOGAF 2006 Edition (Incorporating 8.1.1) . US : Open Group. (pp. 476)
- The Open Group. (2007a). Service Oriented Architecture Overview (SOA). Retrieved December 27, 2007 from <http://www.opengroup.org/onlinepubs/7699929599/toc.pdf>
- The Open Group. (2007b). SOA/TOGAF Practical Guide. Retrieved December 27, 2007 from <http://www.opengroup.org/projects/soa-togaf/>
- The Open Group. (2007c). Analyzing The Open Group Architecture Framework from the GERAM Perspective. Retrieved February 1, 2008 from [http://www.opengroup.org/architecture/wp/saha/TOGAF\\_GERAM\\_Mapping.htm](http://www.opengroup.org/architecture/wp/saha/TOGAF_GERAM_Mapping.htm)
- The Open Group. (2007d). Integrated Information Infrastructure Reference Model. Retrieved February 1, 2008 from <http://www.opengroup.org/architecture/togaf8-doc/arch/chap22.html>
- The Open Group. (2007e). ADM and the Zachman Framework. Retrieved February 1, 2008 from <http://www.opengroup.org/architecture/togaf8-doc/arch/chap39.html>
- Ross, Jeanne W., Weill, Peter, Robertson, David C.. (2006a). Enterprise Architecture As Strategy. Boston, Massachusetts: Harvard Business School Press. (pp. vii)
- Ross, Jeanne W., Weill, Peter, Robertson, David C.. (2006a). Enterprise Architecture As Strategy. Boston, Massachusetts: Harvard Business School Press. (pp. vii)

- Schekkerman, Jans. (2004). A Comparative Survey of Enterprise Architecture Frameworks. Retrieved December 12, 2007 from [http://www.enterprise-architecture.info/Images/Documents/Comparative\\_Survey\\_of\\_EA\\_Frameworks.pps](http://www.enterprise-architecture.info/Images/Documents/Comparative_Survey_of_EA_Frameworks.pps)
- Schmelzer, Ronald. (2006). SOA and the Zachman Framework. Retrieved December 12, 2007 from <http://www.zaphink.com/report.html?id=ZAPFLASH-2006615>
- Sessions, Roger. (2007). A Comparison of the Top Four Enterprise-Architecture Methodologies. Retrieved December 19, 2007 from <http://msdn2.microsoft.com/en-us/library/bb466232.aspx>
- Web Services Interoperability Organization. (2007a). Leadership. Retrieved December 20, 2007 from <http://www.ws-i.org/about/leadership.aspx>
- Web Services Interoperability Organization. (2007b). Leadership. Retrieved December 20, 2007 from <http://www.ws-i.org/deliverables/Default.aspx>
- W3Schools. (2007a). Soap Syntax. Retrieved November 21, 2007 from [http://www.w3schools.com/soap/soap\\_syntax.asp](http://www.w3schools.com/soap/soap_syntax.asp)
- Weerawarana, Sanjiva, Curbera, Fran, Leymann, Frank, Storey & Tony, Ferguson, Donald F. (2005a). Web Services Platform Architecture. Stoughton, Massachusetts: Pearson Education Inc. (pp. 53)
- Weerawarana, Sanjiva, Curbera, Fran, Leymann, Frank, Storey & Tony, Ferguson, Donald F. (2005b). Web Services Platform Architecture. Stoughton, Massachusetts: Pearson Education Inc. (pp. 228)
- Weerawarana, Sanjiva, Curbera, Fran, Leymann, Frank, Storey & Tony, Ferguson, Donald F. (2005c). Web Services Platform Architecture. Stoughton, Massachusetts: Pearson Education Inc. (pp. 387-388)
- Wikipedia. (2007a). Enterprise Architecture. Retrieved November 21, 2007 from [http://en.wikipedia.org/wiki/Enterprise\\_Architecture](http://en.wikipedia.org/wiki/Enterprise_Architecture)
- Wikipedia. (2007b). Enterprise Service Bus. Retrieved December 3, 2007 from [http://en.wikipedia.org/wiki/Enterprise\\_service\\_bus](http://en.wikipedia.org/wiki/Enterprise_service_bus)
- Wikipedia. (2007c). Metadata. Retrieved December 4, 2007 from <http://en.wikipedia.org/wiki/Metadata>
- Wikipedia. (2007d). Web Service. Retrieved December 4, 2007 from [http://en.wikipedia.org/wiki/Web\\_service](http://en.wikipedia.org/wiki/Web_service)

- Wikipedia. (2007e). XML. Retrieved December 5, 2007 from <http://en.wikipedia.org/wiki/XML>
- Wikipedia. (2007f). SOAP. Retrieved December 6, 2007 from <http://en.wikipedia.org/wiki/SOAP>
- Wikipedia. (2007g). Universal Description Discover and Integration. Retrieved December 7, 2007 from [http://en.wikipedia.org/wiki/Universal\\_Description\\_Discovery\\_and\\_Integration](http://en.wikipedia.org/wiki/Universal_Description_Discovery_and_Integration)
- Wikipedia. (2007h). ACID. Retrieved December 8, 2007 from <http://en.wikipedia.org/wiki/ACID>
- Web Services Interoperability Organization. (2007a). Leadership. Retrieved December 5, 2007 from <http://www.ws-i.org/about/leadership.aspx>
- Zachman, John A.. (1987) A framework for information systems architecture, IBM Systems Journal, 26(3), 1987. Retrieved November 15, 2007 from <http://www.research.ibm.com/journal/sj/263/ibmsj2603E.pdf>
- Zachman, John A.. (2006a). Zachman Framework for Enterprise Architecture(Zachman eBook). Zachman Framework Associates Metadata Systems Software Inc. (Chapter 6)
- Zachman, John A.. (2006b). Zachman Framework for Enterprise Architecture(Zachman eBook). Zachman Framework Associates Metadata Systems Software Inc. (Chapter 8)
- Zachman, John A.. (2007) Zachman Framework. Retrieved December 5, 2007 from <http://www.zifa.com/framework.html>