A Comparison of the Top Four Enterprise-Architecture Methodologies

Roger Sessions
ObjectWatch, Inc.

May 2007

Applies to:
Enterprise Architecture

Summary: Twenty years ago, a new field was born that soon came to be known as *enterprise architecture*. This paper covers a broad introduction to the field of enterprise architecture. Although the history of the field goes back 20 years, the field is still evolving—and rapidly so. (36 printed pages)

Contents

Executive Summary

Twenty years ago, a new field was born that soon came to be known as *enterprise architecture*. The field initially began to address two problems:

- **System complexity**—Organizations were spending more and more money building IT systems; and
- **Poor business alignment**—Organizations were finding it more and more difficult to keep those increasingly expensive IT systems aligned with business need.

The bottom line: more cost, less value. These problems, first recognized 20 years ago, have today reached a crisis point. The cost and complexity of IT systems have exponentially increased, while the chances of deriving real value from those systems have dramatically decreased.

Today's bottom line: even more cost, even less value. Large organizations can no longer afford to ignore these problems. The field of enterprise architecture that 20 years ago seemed quaintly quixotic today seems powerfully prophetic.

Many enterprise-architectural methodologies have come and gone in the last 20 years. At this point, perhaps 90 percent of the field use one of these four methodologies:

- The Zachman Framework for Enterprise Architectures—Although self-described as a *framework*, is actually more accurately defined as a *taxonomy*
- The Open Group Architectural Framework (TOGAF)—Although called a *framework*, is actually more accurately defined as a *process*
- The Federal Enterprise Architecture—Can be viewed as either an *implemented enterprise architecture* or a *proscriptive methodology* for creating an enterprise architecture
- The Gartner Methodology—Can be best described as an *enterprise architectural practice*

This white paper discusses these four approaches to enterprise architecture. It does so within the context of a fictional company that is facing some very nonfictional operations problems. These problems include:

- IT systems that have become unmanageably complex and increasingly costly to maintain.
- IT systems that are hindering the organization's ability to respond to current, and future, market conditions in a timely and cost-effective manner.
- Mission-critical information that is consistently out-of-date and/or just plain wrong.
- A culture of distrust between the business and technology sides of the organization.

How should this company choose from among these four very different approaches to enterprise architecture? This white paper traces the journey the company is likely to face in using any one of these methodologies.

When examining each of these methodologies in depth, one is struck by the fact that none of these approaches is really complete. Each has strengths in some areas and weaknesses in others.

For many enterprises, none of these methodologies will therefore be a complete solution. For such organizations, this white paper proposes another approach, one that might be called a *blended methodology*. Choose bits and pieces from each of these methodologies, and modify and merge them according to the specific needs of your organization. This white paper gives an approach to creating such a blended methodology that is a best fit for your organization's needs.

But even a blended methodology will only be as good as an organization's commitment to making changes. This commitment must be driven by the highest level of the organization. The good news is that, with a real commitment to change and a tailored methodology for guiding that change, the 20-year-old promise of enterprise architecture is within reach.

That promise hasn't changed: reducing IT cost and complexity, while increasing business value and effectiveness—or, to put it even more simply, improving your competitiveness in an increasingly competitive world.

Introduction

The year 2007 marks the 20-year anniversary of enterprise architecture. In that time, a number of enterprise-architectural methodologies have come and gone. Today, four dominate the field: the Zachman Framework for Enterprise Architectures, The Open Group Architecture Framework (TOGAF), the Federal Enterprise Architecture (FEA), and Gartner (formerly, the Meta Framework).

Should you care about a field that is 20 years old? It depends. This field was inaugurated to address two major problems in information technology (IT) that were then already becoming apparent. The first problem was managing the increasing complexity of information-technology systems. The second problem was the increasing difficulty in delivering real business value with those systems.

As you can imagine, these problems are related. The more complex a system, the less likely it is that it will deliver maximum business value. As you better manage complexity, you improve your chances of delivering real business value.

So, should you care about this field? It depends on how you feel about positively affecting your organization's bottom line. If managing system complexity and delivering business value are key priorities for you, you should care about enterprise-architecture methodologies. If you are focused on maintaining, or rebuilding, IT's credibility in your organization, or if you strive to promote the use of IT to maintain a competitive position in your industry, you should continue reading this white paper. If these issues don't concern you, these methodologies have little to offer.

As systems become more complex, they generally require more planning. It is easy to see this in buildings. When Henry David Thoreau built his little cabin on Walden Pond (shown in Figure 1), he embraced simplicity and needed no architects. If you are building New York City (shown in Figure 2), simplicity is out of the question, and you will need many architects.

Figure 1. Thoreau's cabin at Walden Pond, as drawn by Thoreau's sister, Sophia Thoreau

Figure 2. New York City

The relationship between complexity and planning for buildings and cities is similar for information systems. If you are building a simple, single-user, nondistributed system, you might not need architects at all. If you are building an enterprise-wide, mission critical, highly distributed system, you might need a database architect, a solutions architect, an infrastructure architect, a business architect, and an enterprise architect.

This paper is about the methodologies needed to develop the overall architectural vision for an organization. This is the responsibility of the *enterprise architect*. This is the architect who specializes in the broadest possible view of architecture within the enterprise. This is the architect's architect, the architect who is responsible for coordinating the work of all of the other architects. Do you need such an architect? It all depends on what you are building: Thoreau's cabin or New York City.

Building a large, complex, enterprise-wide information system without an enterprise architect is like trying to build a city without a city planner. Can you build a city without a city planner? Probably. Would you want to live in such a city? Probably not.

Of course, hiring a city planner does not guarantee a livable city; it merely improves your chances. Similarly, having an enterprise architect does not guarantee a successful enterprise architecture. There are many examples of failed enterprise architectures in the world today, and all of them had enterprise architects (probably dozens!). Architectural methodologies can help, but they go only so far. I'll discuss some of the reasons for these failures, and how to avoid them, also in this paper.

Before I get too far into comparing the methodologies that make up the enterprise architect's toolkit, I need to define some terms. This is especially important in an article that is comparing methodologies, because the different methodologies sometimes use similar terms to mean different things.

For example, we have two methodologies that describe themselves as enterprise-architectural frameworks: the Zachman Framework for Enterprise Architectures and The Open Group Architectural Framework (TOGAF). Yet these two methodologies share little in common other than the words *enterprise*, *architecture*, and *framework*.

So, I will start by defining the terms as I will use them in this white paper. Those definitions marked with an asterisk (*) are taken mostly from IEEE-1471-2000 [01], whose definitions I use where they exist and make sense.

architect—One whose responsibility is the design of an architecture and the creation of an architectural description

architectural artifact—A specific document, report, analysis, model, or other tangible that contributes to an architectural description

architectural description*—A collection of products (artifacts) to document an architecture

architectural framework—A skeletal structure that defines suggested architectural artifacts, describes how those artifacts are related to each other, and provides generic definitions for what those artifacts might look like

architectural methodology—A generic term that can describe any structured approach to solving some or all of the problems related to architecture

architectural process—A defined series of actions directed to the goal of producing either an architecture or an architectural description

architectural taxonomy—A methodology for organizing and categorizing architectural artifacts

architecture*—The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution

enterprise architecture—An architecture in which the system in question is the whole enterprise, especially the business processes, technologies, and information systems of the enterprise

Now that we have a common understanding of these key terms, I can take you through the history of enterprise-architecture methodologies, discuss the problems these methodologies are trying to solve, and compare the top four methodologies in terms of their approach and their relationship to each other.

A Brief History of Enterprise Architecture

The field of enterprise architecture essentially started in 1987, with the publication in the *IBM Systems Journal* of an article titled "A Framework for Information Systems Architecture," by J.A. Zachman. In that paper, Zachman laid out both the challenge and the vision of enterprise architectures that would guide the field for the next 20 years. The challenge was to manage the complexity of increasingly distributed systems. As Zachman said:

The cost involved and the success of the business depending increasingly on its information systems require a disciplined approach to the management of those systems. [02]

Zachman's vision was that business value and agility could best be realized by a holistic approach to systems architecture that explicitly looked at every important issue from every important perspective. His multiperspective approach to architecting systems is what Zachman originally described as an *information systems architectural framework* and soon renamed to be an *enterprise-architecture framework*.

Zachman was a major influence on one of the earliest attempts by a branch of the U.S. Government, the Department of Defense, to create an enterprise architecture. This attempt was known as the Technical Architecture Framework for Information Management (TAFIM) [03] and was introduced in 1994.

The promise of enterprise architectures, such as TAFIM, to better align technical projects with business need was noticed by no less a body than the U.S. Congress. Most likely influenced by the promised benefits of TAFIM, Congress in 1996 passed a bill known as the Clinger-Cohen Act of 1996 [04], also known as the Information Technology Management Reform Act, which mandated that all federal agencies take steps to improve the effectiveness of their IT investments. A CIO Council, consisting of CIOs from all major governmental bodies, was created to oversee this effort.

In April 1998, the CIO Council began work on its first major project, the Federal Enterprise Architecture Framework (FEAF). Version 1.1 [05] of this framework was released in September of 1999. This document contained some innovate ideas, such as "segmented architectures"—that is, architectural focus on segmented subsets of the larger enterprise.

Over time, responsibility for federal enterprise architecture moved from the CIO Council to the Office of Management and Budget (OMB). In 2002, the OMB evolved and renamed the FEAF methodology as the Federal Enterprise Architecture (FEA). I will describe FEA in greater detail, in the section dedicated to it.

Despite the very significant enterprise-architectural activity in the Federal Government (one could argue that no organization has spent more money attempting to develop an enterprise architecture than the U.S. Government), progress has been slow and success stories are overshadowed by higher-profile failures. In 2004, a full eight years after the Clinger-Cohen Act mandated the use of effective IT planning processes, the General Accounting Office (GAO) reported the following:

Only 20 of 96 agencies examined had established at least the foundation for effective architecture management. Further, while 22 agencies increased in maturity since 2001, 24 agencies decreased in maturity and 47 agencies remained the same. [06]

Since January of 2005, the General Accounting Office (GAO, not to be confused with the OMB) has severely chastised a number of U.S. agencies for failures in their adoption or use of enterprise architecture. A few examples include the FBI [07], the Department of Defense [08], the Department of Homeland Security [09], and NASA [10].

In 1998, four years after TAFIM (remember TAFIM?) was introduced and two years after it became codified as Clinger-Cohen, TAFIM was officially retired by the Department of Defense.

The work done on TAFIM was turned over to The Open Group. They morphed it into a new standard that is today known as The Open Group Architectural Framework—better known by its acronym, TOGAF. I will discuss the TOGAF work in the section dedicated to that topic.

In 2005, about the same time that OMB was becoming the dominant EA force in the public sector, another organization was taking steps to become a dominant force in the private sector. This group was Gartner.

By 2005, Gartner was already one of the most influential organizations specializing in CIO-level consulting. However, in the specific area of enterprise architecture, the best known IT research and advisory group was not Gartner, but Meta Group.

Gartner had struggled to build an enterprise-architecture practice, but never achieved the status of the Meta Group. In 2005, Gartner decided that if they couldn't compete with Meta Group, they would do the next best thing: They would buy it.

Following the purchase of Meta Group, Gartner/Meta spent a year looking at what each company brought to the table as far as enterprise-architecture experience and methodologies. The two companies discussed how best to reconcile their often quite different approaches.

In the end, a fairly simple algorithm was applied: If Meta Group liked it, it was in; if Meta Group didn't like it, it was out. Gartner liked architectural frameworks. The Meta Group liked architectural process. So, frameworks were out; processes were in. I'll discuss this Gartner/Meta process in detail, in the section devoted to Gartner.

Figure 3 summarizes this history with an enterprise-architecture timeline. This brings us up to date in the history of enterprise architecture. Now, let's look more closely at today's main methodologies and introduce a case study that will be used in this white paper.

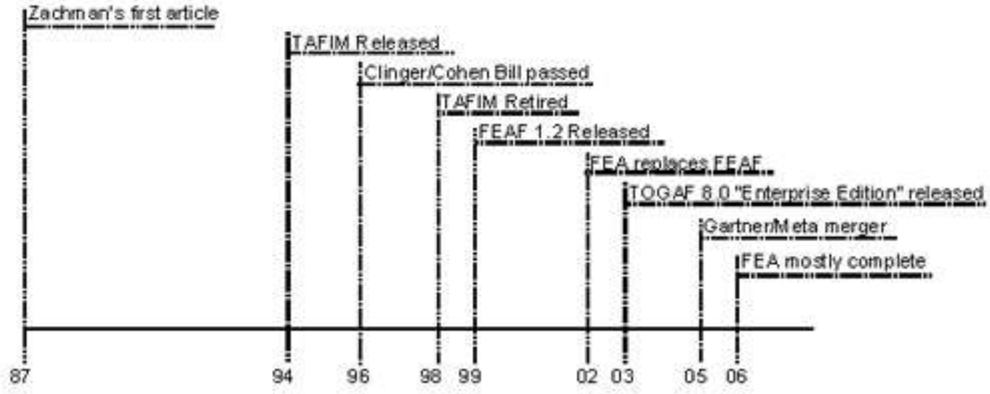


Figure 3. Enterprise-architecture timeline

Case Study

So that we can compare and contrast the four major approaches to enterprise architectures, I am going to illustrate how each would approach a similar scenario. This fictitious scenario is a composite of several enterprises with which I have worked over the past several years. So, while it is fictitious, it is very realistic. I'll first describe the scenario.

MedAMore is a chain of drug stores. It started as a regional chain in 1960. In 1995, it developed an innovative software system that enabled it to run drug stores very efficiently. It called this system MedAManage, or MAM. MAM incorporated some innovative business ideas, such as patient-relationship management, inventory management, automated insurance billing, and even utility optimization.

MAM consisted of three programs: MAM/Store, which ran on a small computer at a drug store; MAM/Warehouse, which ran on a server in a regional warehouse; and MAM/Home, which ran on a large server at the home office.

These three programs communicated through files that were transferred from one location (for example, a store) to another (for example, a regional warehouse). When reliable communications lines existed, file transfers could occur through FTP. The system was also flexible enough to accommodate transfers through courier, where necessary.

By 2000, MedAMore was doing quite well—in part, because of the cost-cutting moves enabled by the MAM system. MedAMore decided to begin expansion. To do this, it purchased three regional chains. With these purchases, MedAMore extended its reach through the southeast quadrant of the U.S.

By 2002, it was clear that the same software systems that had initially fueled MedAMore's success were now hampering its future. Some of the problems MedAMore was running into were the following:

- MAM/Store required regional specializations. For example, different insurance plans needed to be supported in different regions, and these all required changes to the MAM/Store module.
- The regional warehouses that had been acquired through acquisition each had different ways of receiving orders from the retail stores and different procedures from ordering supplies from the wholesalers. Each of these differences required changes to the MAM/Warehouse module.
- The file-transfer approach to information sharing that had worked so well when MedAMore consisted of 30 drugstores, one regional warehouse, and one home office were turning out to be difficult to coordinate among 200 drugstores, four regional warehouses, two geographic offices, and one home office. Files were often delivered late, sometimes not at all, and occasionally multiple times. This made it difficult for the home office to access reliable, up-to-date financial information, especially in the areas of sales and inventory.

It was clear to MedAMore management that the MAM system needed many enhancements. However, upgrading this system was difficult. Each of the three modules (store, warehouse, and home office) was huge, inefficient, and cumbersome, and each included functionality for everything that each entity might need.

The modules had grown to over 1 million lines of code each. It was difficult to change one function without affecting others. All of the functions accessed a single database, and changes to one record definition could ripple through the system in an unpredictable fashion. Changing even a single line of code required a rebuild of the entire multimillion-line module.

MedAManage had become MedANightmare. Debugging was difficult. Software builds were torturous. Installing new systems was hugely disruptive.

These technical problems soon created internal conflicts within the home office of MedAMore. The business side of MedAMore wanted to acquire two more regional chains, but IT was still struggling to bring the existing acquisitions online.

This resulted in a rapidly growing divide between the business and the technical sides of MedAMore. The business side saw IT as reducing business agility. The technical side saw the business side as making impossible demands and blamed it for refusing to consult IT before entering into acquisition discussions.

The distrust had reached such a point that, by 2005, the CIO was no longer considered part of the executive team of MedAMore. The business side distrusted IT and tried to circumvent it at every opportunity. The technical side built its IT systems with little input from the business folks. Several large and expensive IT initiatives were ignored by the business side and were eventually abandoned.

By 2006, MedAMore was in crisis. It clearly needed to revamp its technical systems to make them easier to specialize for regional requirements. This was going to be an expensive proposition, and MedAMore couldn't afford for the effort to fail.

Just as importantly, MedAMore also had to rebuild its internal relationships. The constant bickering and distrust between business and IT was affecting morale, efficiency, and profitability. A company that only five years earlier was an industry leader in profitability—in large part, because of its innovative use of IT—was now struggling to stay out of the red—in large part, because of the inflexibility of those same IT systems.

Cathy, the CEO of MedAMore, desperately needed a solution. At a CEO conference, she heard how many of her peers were using enterprise architectures to build stronger partnerships between their technical and business groups and deliver more cost-effective IT systems that enabled business agility.

Cathy decided that this approach merited further investigation. She asked Irma, her CIO, to prepare a recommendation on the use of an enterprise architecture within MedAMore. Irma was impressed with the approach, but recognized that any such initiative needed to be driven from the top and needed to involve the business side from the start.

On Irma's recommendation, Cathy called a meeting with Bret, the Vice-President of Business, and Irma. Cathy announced that she had decided to create a common enterprise architecture for MedAMore that would unite its technical and business people. This common enterprise architecture would be named MedAMore-Enterprise Architecture, or MAM-EA. After it was completed, MAM-EA would drive all new IT investment and ensure that every dollar invested in IT was delivering the maximum value to the business.

Cathy knew that MAM-EA was a bet-the-company decision for MedAMore. The MAM-EA vision *had* to work. Cathy was depending on Bret (the business side) and Irma (the IT side) to make it work.

So, that is the problem. Now, let's see how each of the EA approaches might provide a solution for MedAMore.

The Zachman Framework for Enterprise Architectures

The first thing we need to understand about the Zachman Framework is that it isn't a framework—at least, by my definition of a framework. According to the American Heritage Dictionary, a framework is defined as:

A structure for supporting or enclosing something else, especially a skeletal support used as the basis for something being constructed; An external work platform; a scaffold; A fundamental structure, as for a written work; A set of assumptions, concepts, values, and practices that constitutes a way of viewing reality. [11].

A *taxonomy*, on the other hand, is defined as:

The classification of organisms in an ordered system that indicates natural relationships; The science, laws, or principles of classification; systematics; Division into ordered groups or categories [12]

The Zachman "Framework" is actually a taxonomy for organizing architectural artifacts (in other words, design documents, specifications, and models) that takes into account both who the artifact targets (for example, business owner and builder) and what particular issue (for example, data and functionality) is being addressed.

As John Zachman retrospectively described his work:

The [Enterprise Architecture] Framework as it applies to Enterprises is simply a logical structure for classifying and organizing the descriptive representations of an Enterprise that are significant to the management of the Enterprise, as well as to the development of the Enterprise's systems. [13]

Many proponents of the Zachman Framework see it as cross-disciplinary, with influence extending far beyond IT. One popular book on Zachman, for example, says:

...in due course, you will discover that the Framework exists in everything you do, not only IT projects. When you thoroughly understand the Framework, you can become more effective in everything you do. This means everything. This statement is not made lightly. [14]

John Zachman himself told me, in an interview that I recently conducted with him:

...the Framework schema has been around for thousands of years and I am sure it will be around for a few more thousands of years. What changes is our understanding of it and how to use it for Enterprise engineering and manufacturing. [15]

Zachman originally explained his IT taxonomy using the building industry as an analogy. In that industry, architectural artifacts are implicitly organized using a two-dimensional organization. One dimension is the various "players in the game." For a physical building, some of these players are the owner (who is paying for the project), the builder (who is coordinating the overall construction), and a zoning board (who is ensuring that construction follows local building regulations).

A building architect prepares different artifacts for each of these players. Every player demands complete information, but what constitutes completeness differs for the different players. The *owner* is interested in a complete description of the functionality and aesthetics of the building. The *builder* is interested in a complete description of the materials and construction process. The owner doesn't care about the placement of studs in the walls. The builder doesn't care how the bedroom windows line up with the morning sun.

As Zachman said in his original article:

...each of the architectural representations differs from the others in essence, not merely in level of detail. [16]

The second dimension for architectural artifact organization is the descriptive focus of the artifact: the what, how, where, who, when, and why of the project. This dimension is independent of the first. Both the builder and the owner need to know *what*, but the owner's need to know *what* is different from the builder's need to know *what*. What *what* is what depends on who is asking the question.

In his first paper and Zachman's subsequent elaboration in 1992 [17], Zachman proposed that there are six descriptive foci (data, function, network, people, time, and motivation) and six player perspectives (planner, owner, designer, builder, subcontractor, and enterprise.) These two dimensions can be arranged in a grid, as shown in Figure 4.

From the business owner's perspective, "data" means business entities. This can include information about the entities themselves, such as customers and products, or information about relationships between those entities, such as demographic groups and inventories. If you are talking to a business owner about data, this is the language you should use.

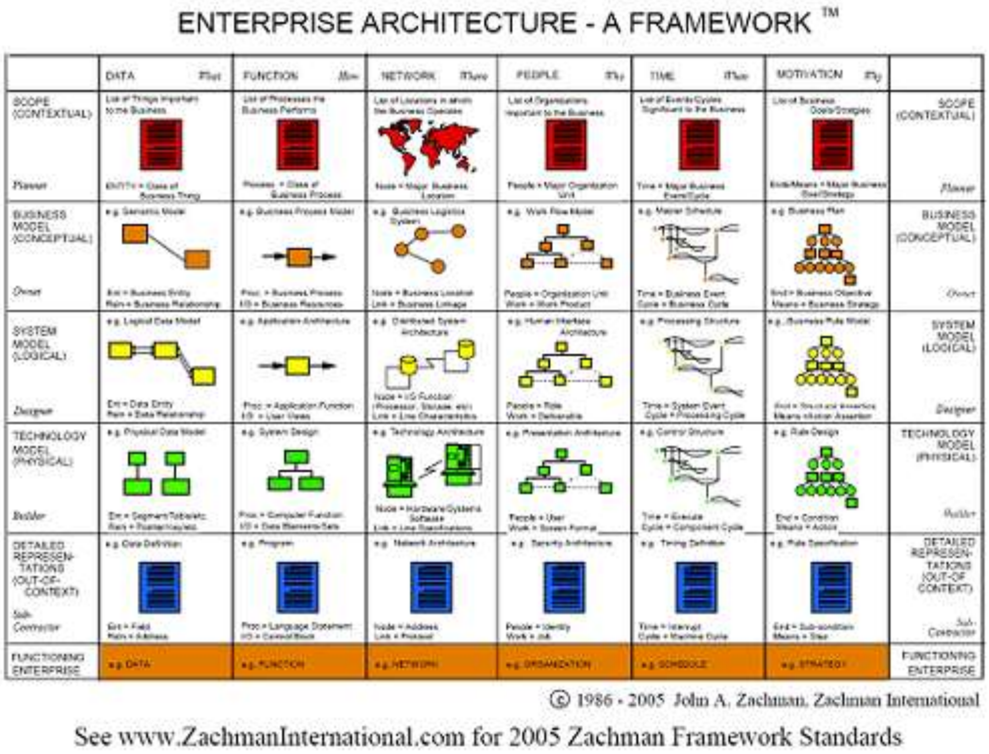
From the perspective of the person implementing the database, "data" does not mean business entities, but rows and columns organized into tables and linked together by mathematical joins and projections. If you are talking to a database designer about data, don't talk about customer demographic groups, but talk about third-normal relational tables.

It's not that one of these perspectives is better than the other or more detailed than the other or of a higher priority than the other. *Both* of these perspectives on data are critical to a holistic understanding of the system's architecture. As Zachman said:

We are having difficulties communicating with one another about information systems architecture, because a set of architectural representations exists, instead of a single architecture. One is not right and another wrong. The architectures are different. They are additive and complementary. There are reasons for electing to expend the resources for developing each architectural representation. And there are risks associated with not developing any one of the architectural representations. [18]

I discussed the historical importance of the Zachman Framework in the history section. Here, I will discuss the actual framework itself and how it could be used to help build MAM-EA, the problem proposed in the case-study section.

As I mentioned earlier, the Zachman Framework consists of six functional foci, each considered from the perspective of a major player. The Zachman Framework as it is portrayed today is shown in Figure 4.



ENTERPRISE ARCHITECTURE - A FRAMEWORK ™

	DATA What	FUNCTION How	NETWORK Where	PEOPLE Who	TIME When	MOTIVATION Why	
SCOPE (CONTEXTUAL) Planner	List of Things Important to the Business ENTITY = Class of Business Thing	List of Processes the Business Performs Process = Class of Business Process	List of Locations in which the Business Operates Node = Major Business Location	List of Organizations Important to the Business People = Major Organization Unit	List of Events/Cycles Significant to the Business Time = Major Business Event/Cycle	List of Business Goals/Strategies Ends/Means = Major Business Goal/Strategy	SCOPE (CONTEXTUAL) Planner
BUSINESS MODEL (CONCEPTUAL) Owner	e.g. Semantic Model Ent = Business Entity Reln = Business Relationship	e.g. Business Process Model Proc. = Business Process I/O = Business Resources	e.g. Business Logistics System Node = Business Location Link = Business Linkage	e.g. Work Flow Model People = Organization Unit Work = Work Product	e.g. Master Schedule Time = Business Event Cycle = Business Cycle	e.g. Business Plan End = Business Objective Means = Business Strategy	BUSINESS MODEL (CONCEPTUAL) Owner
SYSTEM MODEL (LOGICAL) Designer	e.g. Logical Data Model Ent = Data Entity Reln = Data Relationship	e.g. Application Architecture Proc. = Application Function I/O = User Views	e.g. Distributed System Architecture Node = I/S Function (Processor, Storage, etc) Link = Line Characteristics	e.g. Human Interface Architecture People = Role Work = Deliverable	e.g. Processing Structure Time = System Event Cycle = Processing Cycle	e.g. Business Rule Model End = Structural Assertion Means = Action Assertion	SYSTEM MODEL (LOGICAL) Designer
TECHNOLOGY MODEL (PHYSICAL) Builder	e.g. Physical Data Model Ent = Segment/Table/etc. Reln = Pointer/Key/etc.	e.g. System Design Proc. = Computer Function I/O = Data Elements/Sets	e.g. Technology Architecture Node = Hardware/Systems Software Link = Line Specifications	e.g. Presentation Architecture People = User Work = Screen Format	e.g. Control Structure Time = Execute Cycle = Component Cycle	e.g. Rule Design End = Condition Means = Action	TECHNOLOGY MODEL (PHYSICAL) Builder
DETAILED REPRESENTATIONS (OUT-OF-CONTEXT) Sub-Contractor	e.g. Data Definition Ent = Field Reln = Address	e.g. Program Proc. = Language Statement I/O = Control Block	e.g. Network Architecture Node = Address Link = Protocol	e.g. Security Architecture People = Identity Work = Job	e.g. Timing Definition Time = Interrupt Cycle = Machine Cycle	e.g. Rule Specification End = Sub-condition Means = Step	DETAILED REPRESENTATIONS (OUT-OF-CONTEXT) Sub-Contractor
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

© 1986 - 2005 John A. Zachman, Zachman International

See www.ZachmanInternational.com for 2005 Zachman Framework Standards

Figure 4. Zachman grid

As you can see from Figure 4, there are 36 intersecting cells in a Zachman grid—one for each meeting point between a player's perspective (for example, business owner) and a descriptive focus (for example, data.). As we move horizontally (for example, left to right) in the grid, we see different descriptions of the system—all from the same player's perspective. As we move vertically in the grid (for example, top to bottom), we see a single focus, but change the player from whose perspective we are viewing that focus.

There are three suggestions of the Zachman grid that can help MedAMore in the development of MAM-EA.

The first suggestion of the Zachman taxonomy is that every architectural artifact should live in one and only one cell. There should be no ambiguity about where a particular artifact lives. If it is not clear in which cell a particular artifact lives, there is most likely a problem with the artifact itself.

As MedAMore begins accumulating artifacts in the development of MAM-EA, it can use the Zachman grid to clarify the focus of each of these artifacts. For example, artifacts relating to a service-oriented architecture live mostly in the third row (designer's perspective). They generally will not be of interest to the business owner (Bret, in the MedAMore case study).

The second suggestion of the Zachman taxonomy is that an architecture can be considered a *complete* architecture only when every cell in that architecture is complete. A cell is complete when it contains sufficient artifacts to fully define the system for one specific player looking at one specific descriptive focus.

When every cell is populated with appropriate artifacts, there is a sufficient amount of detail to fully describe the system from the perspective of every player (what we might today call a *stakeholder*) looking at the system from every possible angle (descriptive focus). So, MedAMore can use the Zachman grid to ensure that appropriate discussions are occurring between all of the important stakeholders of MAM-EA.

The third suggestion of the Zachman grid is that cells in columns should be related to each other. Consider, for example, the data column (the first column) of the Zachman grid. From the business owner's (Bret's) perspective, *data* is information about the business. From the database administrator's perspective, data is rows and columns in the database.

While the business owner thinks about data quite differently from the database administrator, there should be some relationship between these perspectives. Somebody should be able to follow Bret's business requirements and show that the database design is, in fact, being driven by those requirements. If Bret has requirements that are not traceable down to the database design, we must ask if the business needs will be met by this architecture. On the other hand, it there are database-design elements that do not trace back to business requirements, we might ask if we have included unnecessary design at the database level.

So, we can see five ways in which the Zachman grid can help in the development of MAM-EA. It can help:

1. Ensure that every stakeholder's perspective has been considered for every descriptive focal point.
2. Improve the MAM-EA artifacts themselves by sharpening each of their focus points to one particular concern for one particular audience.
3. Ensure that all of Bret's business requirements can be traced down to some technical implementation.
4. Convince Bret that Irma's technical team isn't planning on building a bunch of useless functionality.
5. Convince Irma that the business folks are including her IT folks in their planning.

But Zachman by itself is not a complete solution for MedAMore. There are far too many issues that will be critical to MAM-EA's success that Zachman does not address. Zachman does not give us a step-by-step process for creating a new architecture. Zachman doesn't even give us much help in deciding if the future architecture we are creating is the best architecture possible. For that matter, Zachman doesn't even give us an approach to show a need for a future architecture. For these and other issues, we are going to need to look at other methodologies.

The Open Group Architecture Framework (TOGAF)

The Open Group Architecture Framework is best known by its acronym, TOGAF. TOGAF is owned by The Open Group [19]. TOGAF's view of an enterprise architecture is shown in Figure 5.

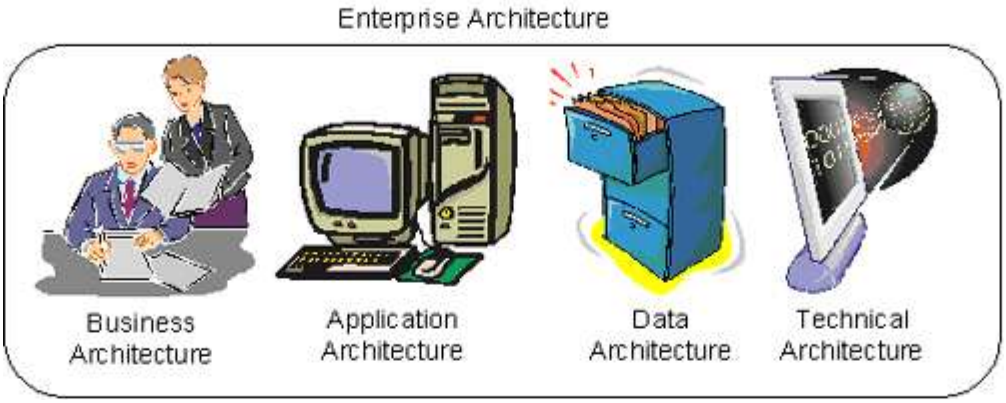


Figure 5. TOGAF's enterprise architecture

As shown in the figure, TOGAF divides an enterprise architecture into four categories, as follows:

1. **Business architecture**—Describes the processes the business uses to meet its goals
2. **Application architecture**—Describes how specific applications are designed and how they interact with each other
3. **Data architecture**—Describes how the enterprise datastores are organized and accessed
4. **Technical architecture**—Describes the hardware and software infrastructure that supports applications and their interactions

TOGAF describes itself as a "framework," but the most important part of TOGAF is the Architecture Development Method, better known as ADM. ADM is a recipe for creating architecture. A recipe can be categorized as a*process*. Given that ADM is the most visible part of TOGAF, I categorize TOGAF overall as an *architectural process*, instead of either an *architectural framework* (as The Open Group describes TOGAF) or a *methodology* (as it describes ADM).

Viewed as an architectural *process*, TOGAF complements Zachman—which, recall, I categorized as an architectural *taxonomy*. Zachman tells you how to categorize your artifacts. TOGAF gives you a process for creating them.

TOGAF views the world of enterprise architecture as a continuum of architectures, ranging from highly generic to highly specific. It calls this continuum the Enterprise Continuum. It views the process of creating a specific enterprise architecture, such as MAM-EA, as moving from the generic to the specific. TOGAF's ADM provides a process for driving this movement from the generic to the specific.

TOGAF calls most generic architectures *Foundation Architectures*. These are architectural principles that can, theoretically, be used by any IT organization in the universe.

TOGAF calls the next level of specificity *Common Systems Architectures*. These are principles that one would expect to see in many—but, perhaps, not all—types of enterprises.

TOGAF calls the next level of specificity *Industry Architectures*. These are principles that are specific across many enterprises that are part of the same domain—such as, in our MedAMore case study, all pharmaceutical enterprises.

TOGAF calls the most specific level the *Organizational Architectures*. These are the architectures that are specific to a given enterprise, such as MedAMore.

Figure 6 shows the relationship between the Enterprise Continuum and the Architecture Development Method (ADM).

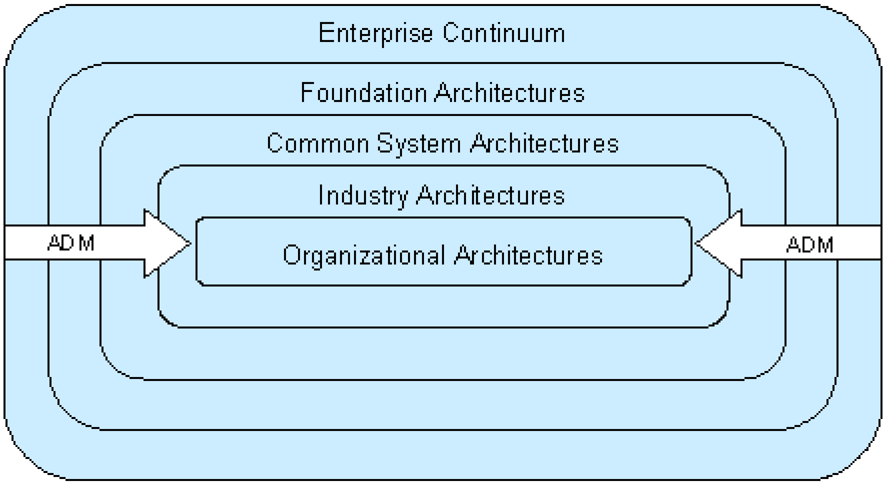


Figure 6. The TOGAF Enterprise Continuum

TOGAF defines the various knowledge bases that live in the Foundation Architecture. Two that you might run into are the *Technical Reference Model* (TRM) and the *Standards Information Base* (SIB). The TRM is a suggested description of a generic IT architecture. The SIB is a collection of standards and pseudo-standards that The Open Group recommends that you consider in building an IT architecture.

TOGAF presents both the TRM and the SIB as suggestions; neither is required. In my view, both the TRM and the SIB are flawed for the same reason: They are biased toward application *portability*, at the expense of application *interoperability* and application *autonomy*. I consider this an outdated view of technical architectures.

For an organization such as MedAMore, TOGAF largely boils down to the Architecture Development Method (ADM). Individuals within MedAMore will be exposed to the Enterprise Continuum, the SIB, and the TRM (as well as a few other TOGAF features), which is why I discussed them. But the day-to-day experience of creating an enterprise architecture will be driven by the ADM, a high-level view of which is shown in Figure 7.

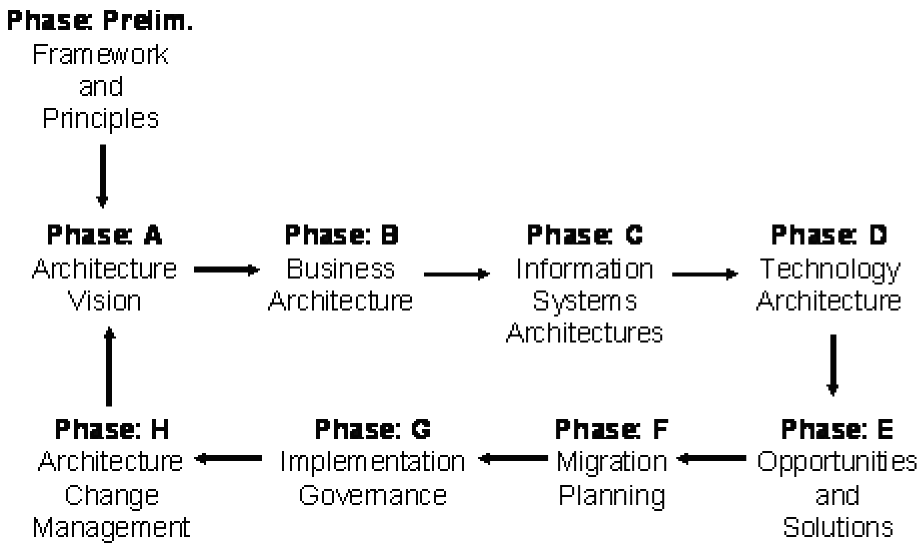


Figure 7. The TOGAF Architecture Development Method (ADM)

As shown in Figure 7, the TOGAF ADM consists of eight phases that are cycled through after an initial "priming of the pump." I'll take you through these phases as they could be applied to the MedAMore case study. But, before MedAMore can start the ADM, it needs to gain some experience with TOGAF. MedAMore will have two choices on how it can get this experience.

First, MedAMore can train itself in TOGAF. MedAMore can download the TOGAF documentation [20]—which describes all of TOGAF, including the ADM, in considerable detail. It can purchase books on TOGAF [21]. There is probably more free and inexpensive available information about TOGAF than about all other architectural methodologies combined.

Second, MedAMore can buy expertise in TOGAF. There are consultants who specialize in TOGAF and have earned Open Group certification [22]. Because MedAMore wants to minimize any chances of failure, it has chosen to call in a TOGAF consultant. MedAMore has brought in Teri, an Open Group–certified TOGAF architect. Remember that the other players at MedAMore are Cath, the CEO of MedAMore; Bret, the Business VP; and Irma, the CIO.

In the Preliminary Phase, Teri meets with the major players at MedAMore to introduce the TOGAF process. Her three goals in the preliminary phase are to:

1. Make sure everybody is comfortable with the process.
2. Modify the TOGAF process, as necessary, to fit within the MedAMore culture.
3. Set up the governance system that will oversee future architectural work at MedAMore.

Teri will work closely with Bret to understand the business philosophy, business models, and strategic drivers of MedAMore. She will work closely with Irma to define the architectural principles that drive technological architectures at MedAMore and document those principles using the TOGAF-recommended format.

In some organizations, achieving buy-in on the need for an enterprise architecture could be very difficult. This is especially true when the effort is driven from the IT organization, and even more so when there is a history of discord between the business and the technical sides of the organization. MedAMore does have this history of animosity. However, it has another fact going for it from which Teri should take heart: The effort is not driven by IT, but is driven by Cath, the CEO. This gives the project high visibility and creates a positive incentive for cooperation from all sides.

As soon as Teri and MedAMore have completed the Preliminary Phase, they are ready to start Phase A. Phase A begins, at least in theory, with a *Request for Architecture Work* from some organization within MedAMore. This document includes the business reasons for the request, budget and personnel information, and any constraints that need to be considered. Because MedAMore has never done a *Request for Architecture Work*, Teri will probably need to work with the sponsoring organization in creating such a request.

As soon as the Request for Architecture Work (or some equivalent) has been received, Teri (the TOGAF consultant) starts MedAMore on Phase A. In Phase A, Teri will ensure that the project has the necessary support within MedAMore, define the scope of the project, identify constraints, document the business requirements, and establish high-level definitions for both the baseline (starting) architecture and target (desired) architecture.

These baseline and target definitions will include high-level definitions on all four of the EA sub-architectures shown back in Figure 5—namely, business, technology, data, and application architectures.

The culmination of Phase A will be a Statement of Architecture Work, which must be approved by the various stakeholders before the next phase of the ADM begins. The output of this phase is to create an architectural vision for the first pass through the ADM cycle. Teri will guide MedAMore into choosing the project, validating the project against the architectural principles established in the Preliminary Phase, and ensure that the appropriate stakeholders have been identified and their issues have been addressed.

The Architectural Vision created in Phase A will be the main input into Phase B. Teri's goal in Phase B is to create a detailed baseline and target business architecture and perform a full analysis of the gaps between them. She will work primarily with Bret (or Bret's team) to achieve this.

Phase B is quite involved—involving business modeling, highly detailed business analysis, and technical-requirements documentation. A successful Phase B requires input from many stakeholders. The major outputs will be a detailed description of the baseline and target business objectives, and gap descriptions of the business architecture.

Phase C does for the information-systems architecture what Phase B does for the business architecture. In this phase, Teri works primarily with Irma (or her team). TOGAF defines nine specific steps, each with multiple sub-steps:

1. Develop baseline data-architecture description
2. Review and validate principles, reference models, viewpoints, and tools
3. Create architecture models, including logical data models, data-management process models, and relationship models that map business functions to CRUD (Create, Read, Update, Delete) data operations
4. Select data-architecture building blocks

5. Conduct formal checkpoint reviews of the architecture model and building blocks with stakeholders
6. Review qualitative criteria (for example, performance, reliability, security, integrity)
7. Complete data architecture
8. Conduct checkpoint/impact analysis
9. Perform gap analysis

The most important deliverable from this phase will be the Target Information and Applications Architecture.

Phase D completes the technical architecture—the infrastructure necessary to support the proposed new architecture. This phase is completed mostly by engaging with Irma's technical team.

Phase E evaluates the various implementation possibilities, identifies the major implementation projects that might be undertaken, and evaluates the business opportunity associated with each. The TOGAF standard recommends that Teri's first pass at Phase E "focus on projects that will deliver short-term payoffs and so create an impetus for proceeding with longer-term projects."

This is good advice in any architectural methodology. Therefore, Teri should be looking for projects that can be completed as cheaply as possible, while delivering the highest perceived value. A good starting place to look for such projects is the organizational pain-points that initially convinced Cath (the MedAMore CEO) to adopt an enterprise architectural-based strategy in the first place. These pain-points, described earlier, included difficulties in completing regional/warehouse specialization and unreliability in data sharing.

Phase F is closely related to Phase E. In this phase, Teri works with MedAMore's governance body to sort the projects identified in Phase E into priority order that include not only the cost and benefits (identified in Phase E), but also the risk factors.

In Phase G, Teri takes the prioritized list of projects and creates architectural specifications for the implementation projects. These specifications will include acceptance criteria and lists of risks and issues.

The final phase is H. In this phase, Teri modifies the architectural change-management process with any new artifacts created in this last iteration and with new information that becomes available.

Teri is then ready to start the cycle again. One of the goals from the first cycle should be information transfer, so that Teri's services are required less and less as more and more iterations of the cycle are completed.

Much of the results of the TOGAF process will be determined as much by the Teri/MedAMore relationship as it will by the TOGAF specification itself. TOGAF is meant to be highly adaptable, and details for the various architectural artifacts is sparse. As one book on TOGAF says:

TOGAF is not wholly specific with respect to generated documents; in fact, it provides very little in the way of prescriptive document templates—merely guidelines for inputs and outputs. [23]

The TOGAF specification is also flexible with respect to the phases. As the specification itself says:

One of the tasks before applying the ADM is to review its components for applicability, and then tailor them as appropriate to the circumstances of the individual enterprise. This activity might well produce an "enterprise-specific" ADM. [24]

TOGAF allows phases to be done incompletely, skipped, combined, reordered, or reshaped to fit the needs of the situation. So, it should be no surprise if two different TOGAF-certified consultants end up using two very different processes—even when working with the same organization.

TOGAF is even more flexible about the actual generated architecture. In fact, TOGAF is, to a surprising degree, "architecture-agnostic". The final architecture might be good, bad, or indifferent. TOGAF merely describes *how*to generate an enterprise architecture, not necessarily how to generate a *good* enterprise architecture. For this, you are dependent on the experience of your staff and/or TOGAF consultant. People adopting TOGAF in the hopes of acquiring a magic bullet will be sorely disappointed (as they will be with any of the methodologies).

Federal Enterprise Architecture (FEA)

The Federal Enterprise Architecture (FEA) is the latest attempt by the federal government to unite its myriad agencies and functions under a single common and ubiquitous enterprise architecture. FEA is still in its infancy, as most of the major pieces have been available only since 2006. However, as I discussed in the history section, it has a long tradition behind it and, if nothing else, has many failures from which it has hopefully learned some valuable lessons.

FEA is the most complete of all the methodologies discussed so far. It has both a comprehensive taxonomy, like Zachman, and an architectural process, like TOGAF. FEA can be viewed as either a methodology for creating an enterprise architecture or the result of applying that process to a particular enterprise—namely, the U.S. Government. I will be looking at FEA from the methodology perspective. My particular interest here is how we can apply the FEA methodology to private enterprises.

Most writers describe FEA as simply consisting of five reference models, one each for performance: business, service, components, technical, and data. It is true that FEA has these five references models, but there is much more to FEA than just the reference models. A full treatment of FEA needs to include all of the following:

- A perspective on how enterprise architectures should be viewed (the *segment* model, that I will describe shortly)
- A set of reference models for describing different perspectives of the enterprise architecture (the five models, mentioned earlier)
- A process for creating an enterprise architecture
- A transitional process for migrating from a pre-EA to a post-EA paradigm
- A taxonomy for cataloging assets that fall within the purview of the enterprise architecture
- An approach to measuring the success of using the enterprise architecture to drive business value

You can see that the FEA is about much more than models. It includes everything necessary to build an enterprise architecture for probably the most complex organization on earth: the U.S. Government. As the FEA-Program Management Office (FEAPMO) says, FEA, taken *in toto*, provides:

...a common language and framework to describe and analyze IT investments, enhance collaboration and ultimately transform the Federal government into a citizen-centered, results-oriented, and market-based organization as set forth in the President's Management Agenda. [25].

While it might be a stretch to imagine that anything short of divine intervention could "transform the Federal government into a citizen-centered, results-oriented, and market-based organization," there is at least hope that some of the FEA methodology could help our beleaguered MedAMore corporation deal with its much more mundane problems. So, let's take a look at what FEA has to offer.

The FEA Perspective on EA

The FEA Perspective on EA is that an enterprise is built of *segments*, an idea first introduced by FEAF [26]. A segment is a major line-of-business functionality, such as human resources. There are two types of segments: *core mission-area segments* and *business-services segments*.

A *core mission-area segment* is one that is central to the mission or purpose of a particular political boundary within the enterprise. For example, in the Health and Human Services (HHS) agency of the federal government,*health* is a core mission-area segment.

A *business-services segment* is one that is foundational to most, if not all, political organizations. For example, *financial management* is a business-services segment that is required by all federal agencies.

Another type of enterprise-architecture asset is an *enterprise service*. An enterprise service is a well-defined function that spans political boundaries. An example of an enterprise service is *security management*. Security management is a service that works in a unified manner across the whole swath of the enterprise.

The difference between *enterprise services* and *segments*, especially *business-service segments*, is confusing. Both are shared across the entire enterprise. The difference is that business-service segments have a scope that encompasses only a single political organization. Enterprise services have a scope that encompasses the entire enterprise.

In the federal government, for example, both HHS and the Environmental Protection Agency (EPA) use the *human resources* business-service segment. However, the people who are managed by human resources are a different group for HHS from what they are for the EPA.

Both HHS and the EPA also use the *security management* enterprise service. But the security credentials that are managed by the security-management service are not specific to either of those agencies. Security credentials are managed effectively only when they are managed at the scope of the enterprise.

Resist the temptation to equate either *segments* or *services* with services, as in *service-oriented architectures*. There are two reasons such a comparison would be flawed. Firstly, enterprise services, business-service segments, and core mission-area segments are all much broader in focus than services found in service-oriented architectures. Secondly, segments are an organizational unit for an *enterprise architecture*, whereas services are an organizational unit for *technical implementations*. As organizational units for an enterprise architecture, their depth includes not just the technical, but also the business and the data architectures.

One final note about segments: Although segments *function* at the political (that is, agency) level, they are *defined* at the enterprise (that is, government) level. Enterprise services, of course, both function and are defined at the enterprise level.

The fact that segments are defined globally facilitates their reuse across political boundaries. One can map out the usage of segments across the political boundaries of the enterprise, then use that map to seek opportunities for architectural reuse. Figure 8, for example, shows a segment map of the federal government from the FEA Practice Guide [27]. As you can see, there are many segments (the vertical columns) that are used in multiple agencies, and any or all of these are good candidates for sharing.

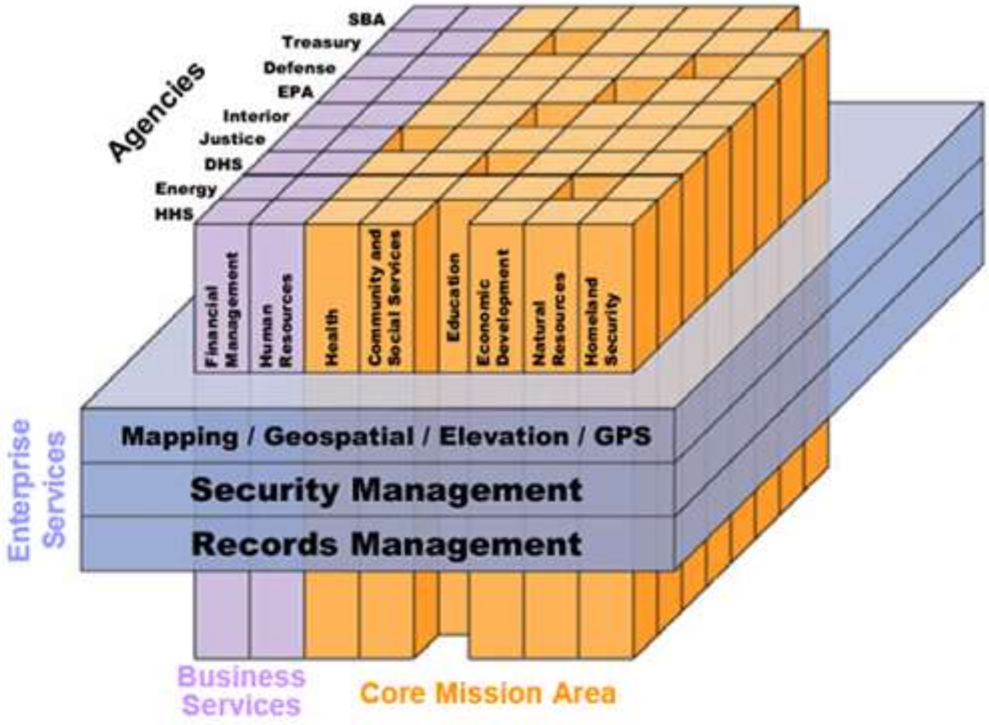


Figure 8. Segment map of the federal government

FEA Reference Models

The five FEA reference models are all about establishing common languages. The goal here is to facilitate communication, cooperation, and collaboration across political boundaries. According to the FEAPMO:

The FEA consists of a set of interrelated "reference models" designed to facilitate cross-agency analysis and the identification of duplicative investments, gaps, and opportunities for collaboration within and across agencies. Collectively, the reference models [compose] a framework for describing important elements of the FEA in a common and consistent way. [28]

Why do we need a common language? Consider this exchange:

James: Do you have a torch I can borrow?

Roger: No, I'm afraid not.

James: Do you know where I can get one?

Roger: The hardware store in town should have one.

So, James goes out to the hardware store and buys himself a torch. He returns.

Roger: Did you get your torch?

James: Yes, here it is.

Roger: That's not a torch! That's a flashlight. Why didn't you say so? I have one you could have borrowed.

James: Well, why didn't you say so?

The problem, of course, is that James comes from England, where what I call a *flashlight* they call a *torch*. And when I hear *torch*, I think of a *blowtorch*. Although we both speak English, we don't necessarily speak the same English. The result is that James goes out and unnecessarily spends money on something that I could have lent him.

This is exactly the problem that the FEA Reference Models are trying to solve on a much larger scale. Suppose the Internal Revenue Service (IRS) decides it needs a *demographics* system to track taxpayer data. They ask around to see if anybody has one they can modify for their purposes. Nobody does.

Little do they know that, right next door, the Government Printing Office (GPO) has a perfectly good demographics system that is almost exactly what the IRS needs. They just happen to call it a *customer-analytics* system.

So, the IRS goes out and builds its system from scratch, instead of just modifying the one already built (and paid for) by the GPO. And, in doing so, the IRS will waste considerably more money than James spent on his unnecessary flashlight.

This, in a nutshell, is the goal of the five FEA reference models: to give standard terms and definitions for the domains of enterprise architecture and, thereby, facilitate collaboration and sharing across the federal government. The five reference models are as follows:

1. The *Business Reference Model (BRM)* gives a business view of the various functions of the federal government. For example, the BRM defines a standard business capability called *water resource management* that is a subfunction of *natural resources* that is considered a *line-of-business* of the broader *services for citizens* business area. [29]
2. The *Components Reference Model (CRM)* gives a more IT view of systems that can support business functionality. For example, the CRM defines a *customer-analytics* system that I described earlier in the hypothetical interchange between the IRS and the GPO. [30]
3. The *Technical Reference Model (TRM)* defines the various technologies and standards that can be used in building IT systems. For example, the TRM defines *HTTP* as a *protocol* that is a subset of a *service transport* that is a subset of *service access and delivery*. [31]
4. The *Data Reference Model (DRM)* defines standard ways of describing data. For example, the DRM defines an *entity* as something that *contains attributes* and *participates in relationships*. [32]
5. The *Performance Reference Model (PRM)* defines standard ways of describing the value delivered by enterprise architectures. For example, the PRM describes *quality* as a technology measurement

area that is defined as "the extent to which technology satisfies functionality or capability requirements." [33]

FEA Process

The FEA Process is primarily focused on creating a segment architecture for a subset of the overall enterprise (in FEA's case, the enterprise is the federal government and the subset is a governmental agency) and is described in the FEA Practice Guidance [34]. I discussed the FEA vision on enterprise segments earlier. The overall segment-architecture development process is (at a very high level) as follows:

- Step 1: Architectural Analysis—Define a simple and concise vision for the segment, and relate it back to the organizational plan.
- Step 2: Architectural Definition—Define the desired architectural state of the segment, document the performance goals, consider design alternatives, and develop an enterprise architecture for the segment, including business, data, services, and technology architectures.
- Step 3: Investment and Funding Strategy—Consider how the project will be funded.
- Step 4: Program-Management Plan and Execute Projects—Create a plan for managing and executing the project, including milestones and performance measures that will assess project success.

FEA Success Measurement

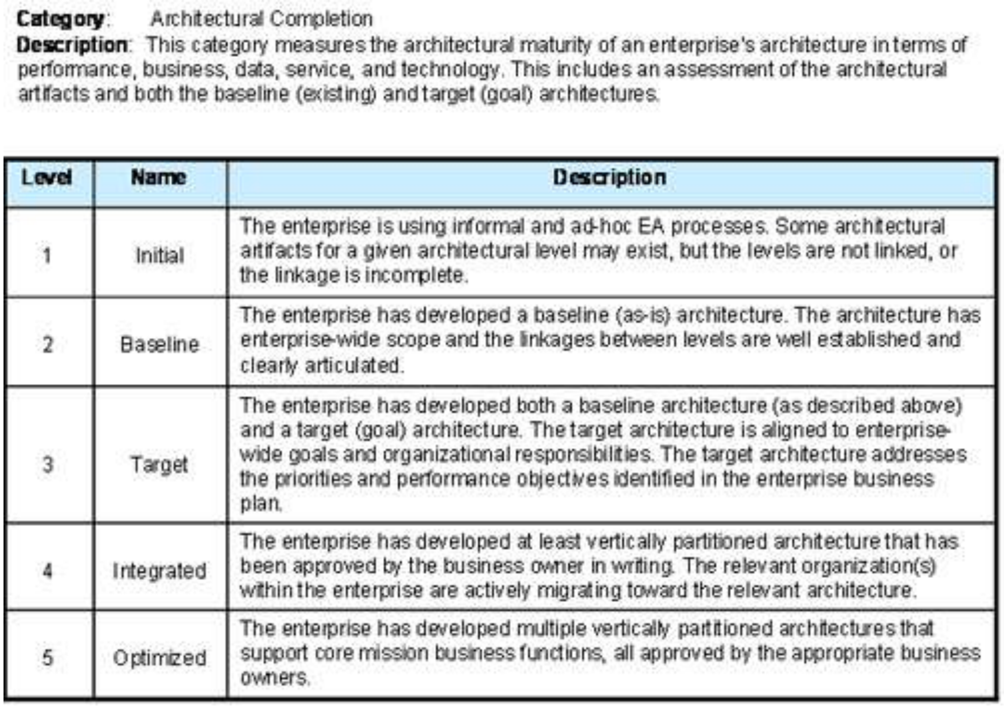
The FEA framework for measuring organizational success in using enterprise architecture is defined in the Federal Enterprise Architecture Program EA Assessment Framework 2.1 [35]. Federal agencies are rated on their overall maturity levels in three main categories:

1. Architectural completion—Maturity level of the architecture itself
2. Architectural use—How effectively the agency uses its architecture to drive decision-making
3. Architectural results—The benefits being realized by the use of the architecture

OMB assigns each agency a success rating, based on its scores in each category and a cumulative score, as follows:

- Green—The agency rates quite well in the *completion* area (it has a quite mature enterprise architecture). It also rates well in both the *use* area (it is effectively using that enterprise architecture to drive ongoing strategy) and the *results* area (the usage of that architecture is driving business value).
- Yellow—The agency rates quite well in the *completion* area. It also rates well in either the *use* area or the *results* area.
- Red—The agency either does not have a completed architecture and/or is not effectively using that architecture.

The framework is interesting beyond the confines of the public sector. The category ratings can be fruitfully adapted by many enterprises to assess the maturity level of their own architectural efforts. Figure 9, for example, shows my own interpretation of the OMB maturity rankings for *architectural completion*, as I adapt them for the private sector. Similar adaptations can be created for *architectural usage* and *architectural results*.



Category: Architectural Completion

Description: This category measures the architectural maturity of an enterprise's architecture in terms of performance, business, data, service, and technology. This includes an assessment of the architectural artifacts and both the baseline (existing) and target (goal) architectures.

Level	Name	Description
1	Initial	The enterprise is using informal and ad-hoc EA processes. Some architectural artifacts for a given architectural level may exist, but the levels are not linked, or the linkage is incomplete.
2	Baseline	The enterprise has developed a baseline (as-is) architecture. The architecture has enterprise-wide scope and the linkages between levels are well established and clearly articulated.
3	Target	The enterprise has developed both a baseline architecture (as described above) and a target (goal) architecture. The target architecture is aligned to enterprise-wide goals and organizational responsibilities. The target architecture addresses the priorities and performance objectives identified in the enterprise business plan.
4	Integrated	The enterprise has developed at least vertically partitioned architecture that has been approved by the business owner in writing. The relevant organization(s) within the enterprise are actively migrating toward the relevant architecture.
5	Optimized	The enterprise has developed multiple vertically partitioned architectures that support core mission business functions, all approved by the appropriate business owners.

Figure 9. OMB ranking of architectural completion, adapted for private sector by author (Roger Sessions)

FEA Applied to MedAMore

Now that I have taken you through the FEA approach, let's see what this might mean to MedAMore. Let's assume that Cath (MedAMore's CEO) has heard about FEA and how it is promising to streamline the federal government. If it can do this for the federal government, she reasons, surely it can do this for her company.

Cath hires a consultant, Fred, who is an expert on FEA (if such a thing can be said to exist for a methodology that is, at the time of this writing, less than a year old!). Fred's job is to show MedAMore how to do FEA—of course, not the real FEA, but FEA as it might be applied to the private sector. Cath introduces Fred to Bret (the business VP) and Irma (the CIO), and tells them to build her anMEA—FEA adapted for MedAMore.

Keep in mind that Cath has taken quite a risk. No other company to date has attempted to apply FEA to the private sector; and even the experience of using FEA within the public sector is nominal, at best.

The first thing that Fred will want to do is build enthusiasm for MEA. Keep in mind that he is coming into an organization in which the business folks barely speak to IT folks. If MEA is going to succeed, it needs to transform not only processes, but people. He will want to create a series of seminars explaining the value of the soon-to-be-defined MEA and how MEA will benefit not only MedAMore as a whole, but the individual units specifically.

Fred will next build a governance structure—MedAMore's equivalent to FEAPMO. I'll call this group MEAPMO. MEAPMO will own MEA, including the processes, the models, and the architecture itself.

The next thing that Fred will likely do is create reference models that can be used by all of the organizations across MedAMore. The five reference models from FEA can serve as a starting point. Some, such as the Technical Reference Model, might be usable with few modifications. Others, such as the Business Reference Model, will require extensive renovation. He shouldn't do these in excruciating detail, but create starting points and build them up as MEA evolves.

Next, Fred will probably want to create a description of the segment architecture as it applies to MedAMore. Note that he will not be doing a complete segment architecture—just a high-level description. The actual process of completing the architecture will be a constantly evolving project.

By this point, a lot of work will have been done with few results. Fred will probably want to take a first pass at a segment-architecture process. FEA's process will be a good starting point, but will require specialization to MedAMore at the detail level (such as who the team members are and what form the generated artifacts should take).

Now, Fred will test-drive the process with the first segment delivery. He will need to build a team, and then lead this team in analyzing and prioritizing the segments—mapping them to business value, determining their architectural options, delivering the work, and, perhaps most importantly, measuring the results. These measurements will be critical in building momentum for future work.

Soon after completing the first segment, Fred might decide that it is time to measure the progress of the different groups in MedAMore in using MEA effectively. To do so, Fred needs a yardstick to measure the success of the different groups within MedAMore in driving business value with MEA. Fred thus leads MEAPMO in building a MedAMore equivalent to the Federal Enterprise Architecture Program EA Assessment Framework [35]. This yardstick will be Cath's main tool for ensuring that both the different groups are taking MEA seriously and her investment is paying off.

And, finally, after Fred has completed this process, he will start the process again. Each iteration will result in new segments being delivered, more business value being generated, and more substance being added to the MEA methodology. At least, this is the theory. As I said earlier, with MEA, we are working at the bleeding edge.

Gartner

So far, I have written about three different methodologies that come together under the banner of enterprise architectures. This last methodology is a little different. It isn't a taxonomy (like Zachman), a process (like TOGAF), or a complete methodology (like FEA). Instead, it is what I define as a *practice*. It is the enterprise-architecture practice of one of the best known IT research and consulting organizations in the world: Gartner.

Let me spend a moment exploring my use of the word *practice*. I use the word "practice" much like I would to describe a physician's practice. A physician—say, Dr. Pérez—does not diagnose a disease by studying taxonomies, although taxonomies do help him communicate to other healthcare providers. He does not diagnose a disease by following a process, although he might go through an informal process in his head. He diagnoses a disease by applying his practice skills. These practice skills include his experience, training, and ongoing relationships with his colleagues.

How do you choose a physician? Do you grill candidates on how well they know the taxonomy of medicine? Do you sit candidates down and ask for a detailed description of the methodology each follows to diagnose illness? Probably not. You might ask your friends, but they probably only know a limited pool of candidates.

One approach to choosing a physician is to go to a well-known institution (a hospital or medical school) and choose from among their staff. In this approach, you are counting on the institution to choose highly qualified physicians and to have developed a community that encourages collaboration and best practices.

Does that institution insist on a rigid methodology for its physicians to follow? Probably not. Even if it does, it is not your primary concern. You are not even concerned with who the physicians in the institution are—although, in time, that will be of more interest to you. Your initial concern is only the reputation of the institution.

This is very similar to the Gartner approach to enterprise architecture. You don't bring in Gartner because they do or don't use TOGAF. You don't go to Gartner because they do or don't follow Zachman's taxonomy. You go to Gartner because they are well-known in their field. You assume both that they hire well-qualified specialists and that they have developed a community that encourages collaboration and best practice.

If you are a Gartner customer and you check the Garner library for research notes describing their enterprise-architecture practice, you can find many such documents. For example, there is "Gartner Enterprise Architecture Process: Evolution 2005" [36] and "Gartner Enterprise Architecture Framework: Evolution 2005" [37]. However, these documents contain little descriptive information and, in any case, are dated in the late-2005 timeframe. Gartner contends that these best practices are timeless, and they continue to augment them as appropriate. The current Gartner methodology was not solidified until probably April of 2006, after the Gartner/Meta merger that I described in the history section.

The best summation of the Gartner practice that I have heard is the following:

Architecture is a verb, not a noun.

What does it mean to say that architecture is a verb, not a noun? It means that it is the ongoing process of creating, maintaining, and, especially, leveraging an enterprise architecture that gives an enterprise architecture its vitality. An architecture that is just a bunch of stiff artifacts that sit in a corner gathering dust is useless, regardless of how sophisticated your taxonomy is for categorizing those artifacts or how brilliant your process is that guided their development.

Gartner believes that enterprise architecture is about bringing together three constituents: business owners, information specialists, the technology implementers. If you can bring these three groups together and unify them behind a common vision that drives business value, you have succeeded; if not, you have failed. Success is measured in pragmatic terms, such as driving profitability, not by checking off items on a process matrix.

Gartner believes that the enterprise architectures must start with where an organization is going, not with where it is. If we are going to clean house, we don't need to exhaustively document everything we are throwing out. Let's focus our energy on what we want to end up with. As soon as we know our goal, we can see how what we have relates to that goal.

Gartner recommends that an organization begin by telling the story of where its strategic direction is heading and what the business drivers are to which it is responding. Gartner will want this story in plain language, without worrying about prescribed documentation standards, acronyms, or techno-babble. The only goal is making sure that everybody understands and shares a single vision.

Most organizations are facing major changes in their business processes. The process of creating an enterprise-architecture vision is the organization's opportunity to sit down, take a collective breath, and ensure that everybody understands the nature, the scope, and the impact of those changes.

As soon as an organization has this single shared vision of the future, it can consider the implications of this vision on the business, technical, information, and solutions architectures of the enterprise. The shared vision of the future will dictate changes in all of these architectures, assign priorities to those changes, and keep those changes grounded in business value.

Enterprise architecture, in the Gartner view, is about strategy, not about engineering. It is focused on the destination. The two things that are most important to Gartner are *where an organization is going* and *how it will get there*. Any architectural activity that is extraneous to these questions is irrelevant. "Just enough enterprise architecture, just in time," is another saying you will hear from the Gartner analyst.

Let's say Cath (MedAMore's CEO) likes what she hears. How is a Gartner engagement likely to proceed? With FEA, TOGAF, or Zachman, Cath needs to start by finding a qualified consultant who understands the methodology. With Gartner, this step is much easier: She merely calls Gartner.

Let's say Gartner sends Greg, the Gartner EA consultant. The first thing Greg will want to do is make sure the architecture is driven from the highest levels of the corporation. The fact that he is being called by MedAMore's CEO will be very reassuring.

Exactly how Greg will proceed is difficult to predict, because Gartner does not have a firm, step-by-step process. However, it is likely that he will start by focusing on Cath's strategic vision for MedAMore. He will want her to specify her vision in business terms and resist any temptation to discuss technology. Here are some possible business-vision statements Greg might elicit:

- MedAMore will have stores in at least 30 states, spread out over 8 geographic regions, by the year 2010. It will accomplish this mainly through acquisition of regional pharmacies.
- MedAMore will be able to assimilate new regional systems within 120 days of finalization of purchase.
- MedAMore will reduce its purchasing costs by 10 percent, by consolidating all regional purchasing into a central system.
- MedAMore's central office will be able to view consolidated sales and inventory reports from all stores that include data up to and including the previous day.
- MedAMore will be able to reduce its inventory on-hand to no more than a five-day supply.
- MedAMore will be able to invoice insurance companies by the end of the day on which the prescription was delivered to the patient.
- Patients will be able to transfer prescriptions from any MedAMore pharmacy to any other.

- Patients will be able to request prescription refills though a Web interface and receive e-mail notification of their availability for pick-up.

Notice that none of these visionary statements mentions technology (except as a delivery mechanism, in the last statement). Greg is purposely keeping these early discussions focused on business strategy.

Any one of Cath's vision "bullets" will have major ramifications across the business, information, and technical architectures. Part of Greg's job will be to prioritize the bulleted items. Let's say Cath decides that her highest priority is consolidating purchasing, because this will improve profitability in the near term.

Greg will soon work to turn Cath's idea about consolidated purchasing into a common-requirements vision (CRV). The CRV is where we will see some of the changes that will be required to drive Cath's vision for MedAMore. Greg will be going over the business changes with Bret and the technical and information changes with Irma, but he will also be working to bring everybody together as a unified team.

Greg will work with Bret (the business VP) to develop a target business architecture that supports consolidated purchasing. As soon as they have spec'd out the future system, they will also look at their current architecture to see what can be recycled.

Greg will work with Irma (the CIO) to develop a target information architecture that allows the home office to track regional inventories and consolidate procurement. They will also work on the technical architecture for the IT systems that will support the new business architecture. After they understand the future, they will look at current architectures for opportunities to reuse existing technology assets.

After Greg has completed the broad-brush architecture for their strategic vision, he will probably step back from the picture until the consolidated purchasing system has been implemented. If Cath needs help with the implementation of the architecture, she will likely look outside of Gartner, because Gartner does not do implementations.

As soon as the implementation of consolidated purchasing has been completed, Greg will step back in to help with the next iteration. His approach will be to keep the architecture at a high level, business-focused, and hone in on details only when and where necessary. He will continue to see his role not as creating an enterprise architecture for MedAMore, but helping them institute a process for allowing an enterprise architecture to emerge and evolve from the business strategy.

Comparison

As you can see, the leading enterprise-architecture methodologies are very different in their approaches. Which one is best for your organization? There is no one answer to this question. I'll take you through the 12 criteria that I most often use for comparing and evaluating enterprise-architectural methodologies. Not all of these criteria might be relevant to your organization, and some might be more important than others. But, at least, this section can serve as a starting point for your own evaluation.

I'll rank each methodology in each criteria. The ratings will be assigned as follows:

- 1: Does a very poor job in this area

- 2: Does an inadequate job in this area
- 3: Does an acceptable job in this area
- 4: Does a very good job in this area

Keep in mind that these ratings are subjective. I'm sure most people would disagree with at least one of my ratings.

Taxonomy completeness refers to how well you can use the methodology to classify the various architectural artifacts. This is almost the entire focus of Zachman. None of the other methodologies focuses as much on this area. Ratings:

- Zachman: 4
- TOGAF: 2
- FEA: 2
- Gartner: 1

Process completeness refers to how fully the methodology guides you through a step-by-step process for creating an enterprise architecture. This is almost the entire focus of TOGAF, with its Architecture Development Method (ADM). Ratings:

- Zachman: 1
- TOGAF: 4
- FEA: 2
- Gartner: 3

Reference-model guidance refers to how useful the methodology is in helping you build a relevant set of reference models. This is almost the entire focus of FEA. TOGAF also provides support; however, I am less impressed with the TOGAF reference models. Ratings:

- Zachman: 1
- TOGAF: 3
- FEA: 4
- Gartner: 1

Practice guidance refers to how much the methodology helps you assimilate the mindset of enterprise architecture into your organization and develop a culture in which it is valued and used. This is a primary focus of Gartner's architectural practice. Ratings:

- Zachman: 1
- TOGAF: 2
- FEA: 2
- Gartner: 4

Maturity model refers to how much guidance the methodology gives you in assessing the effectiveness and maturity of different organizations within your enterprise in using enterprise architecture. Ratings:

- Zachman: 1
- TOGAF: 1

- FEA: 3
- Gartner: 2

Business focus refers to whether the methodology will focus on using technology to drive business value, in which business value is specifically defined as either reduced expenses and/or increased income. Ratings:

- Zachman: 1
- TOGAF: 2
- FEA: 1
- Gartner: 4

Governance guidance refers to how much help the methodology will be in understanding and creating an effective governance model for enterprise architecture. Ratings:

- Zachman: 1
- TOGAF: 2
- FEA: 3
- Gartner: 3

Partitioning guidance refers to how well the methodology will guide you into effective autonomous partitions of the enterprise, which is an important approach to managing complexity. Ratings:

- Zachman: 1
- TOGAF: 2
- FEA: 4
- Gartner: 3

Prescriptive catalog refers to how well the methodology guides you in setting up a catalogue of architectural assets that can be reused in future activities. Ratings

- Zachman: 1
- TOGAF: 2
- FEA: 4
- Gartner: 2

Vendor neutrality refers to how likely you are to get locked-in to a specific consulting organization by adopting this methodology. A high rating here indicates low vendor lock-in. Ratings:

- Zachman: 2
- TOGAF: 4
- FEA: 3
- Gartner: 1

Information availability refers to the amount and quality of free or inexpensive information about this methodology. Ratings:

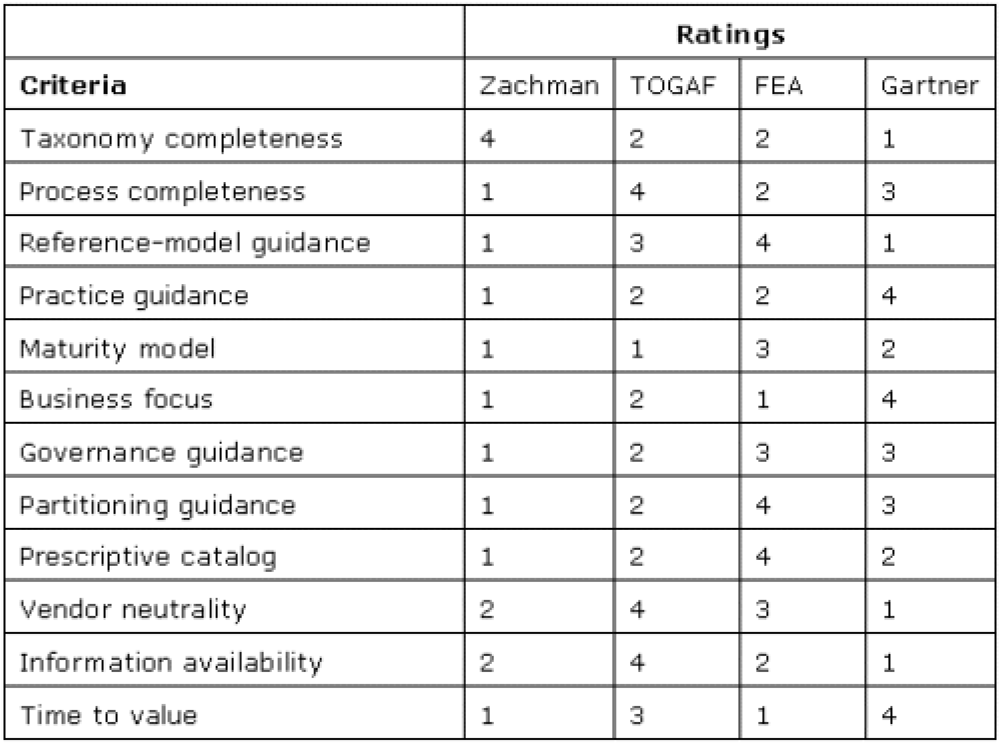
- Zachman: 2

- TOGAF: 4
- FEA: 2
- Gartner: 1

Time to value refers to the length of time you will likely be using this methodology before you start using it to build solutions that deliver high business value. Ratings:

- Zachman: 1
- TOGAF: 3
- FEA: 1
- Gartner: 4

The criteria and ratings are summarized in Figure 10.



	Ratings			
Criteria	Zachman	TOGAF	FEA	Gartner
Taxonomy completeness	4	2	2	1
Process completeness	1	4	2	3
Reference-model guidance	1	3	4	1
Practice guidance	1	2	2	4
Maturity model	1	1	3	2
Business focus	1	2	1	4
Governance guidance	1	2	3	3
Partitioning guidance	1	2	4	3
Prescriptive catalog	1	2	4	2
Vendor neutrality	2	4	3	1
Information availability	2	4	2	1
Time to value	1	3	1	4

Figure 10. Criteria and ratings for each methodology

One of the important points of Figure 10 is that none of the enterprise-architecture methodologies is really complete. Each has its strengths and weaknesses.

How do you choose which methodology is best for you? Here is my recommended approach:

1. Go through the rows (criteria) in Figure 10, eliminating any that you feel are not important to your organization.
2. Add any additional rows (criteria) that you feel are important, and rate each of the methodologies in that area.
3. Change any of my ratings with which you disagree.

At the end of this exercise, you should have a good idea about the strengths and weaknesses of each methodology with respect to your enterprise's needs. If a clear winner emerges, count yourself lucky. Find a consultant who specializes in helping enterprises implement that methodology, and go for it.

For many organizations, there will be no clear winner. For such organizations, I recommend you use a blended approach, in which you create your own enterprise-architectural methodology consisting of bits and pieces of each of the methodologies that provide the highest value in your specific areas of concern.

You will want a different kind of consultant—one who has a broad perspective of all of these methodologies and specializes in helping enterprises create a methodology that works best, given the specific needs and political realities of that enterprise.

Conclusion

This paper has covered a broad introduction to the field of enterprise architecture. The history of the field goes back 20 years, but the field is still evolving—and rapidly so. Two of the four major methodologies (Gartner and FEA) have undergone major changes in the last two years alone.

As I have shown, these methodologies are quite different from each other, both in goals and in approach. This is good news and bad. It is bad news, in that it increases the difficulty for many organizations in choosing one single enterprise-architectural methodology. How do you choose between methodologies that have so little in common? Choosing between Zachman and TOGAF, for example, is like choosing between spinach and hammers.

But the good news is that these methodologies can be seen as complementing each other. For many organizations, the best choice is all of these methodologies, blended together in a way that works well within that organization's constraints. This white paper should provide a good starting place for understanding the value of each of these methodologies and how they can complement each other.

Whichever route you choose, remember that enterprise architecture is a path, not a destination. An enterprise architecture has no value unless it delivers real business value as quickly as possible. One of the most important goals of any enterprise architecture is to bring the business side and the technology sides together, so that both are working effectively toward the same goals.

In many organizations, there is a culture of distrust between the technology and business folks. No enterprise-architecture methodology can bridge this divide unless there is a genuine commitment to change. *That commitment must come from the highest level of the organization.* Methodologies cannot solve people problems; they can only provide a framework in which those problems can be solved.

But, as soon as you have that commitment to change, an enterprise-architecture methodology can be a valuable tool for guiding that change. This change can manifest itself in many ways. Some of the predicted benefits from a successfully implemented enterprise architectural include:

- 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.

It is obvious that an organization that does well in these key areas will be more successful than one that doesn't. This is true regardless of whether success is measured with tangibles, such as profitability and return on investment, or intangibles, such as customer satisfaction and employee turnover.

The starting point for any enterprise architecture is some critical self-analysis. Does your organization spend too much money building IT systems that deliver inadequate business value? Is IT seen as improving or hampering business agility? Is there a growing divide between your business and IT folks? And, finally, perhaps the most important question of all: Is your organization truly committed to solving these problems, and does that commitment come from the highest levels of the organization? If the answer to all of these questions is "yes," enterprise architecture is your path. It's up to you to take that next step.

Glossary

- **ADM (Architecture Development Method)**—A process for creating an enterprise architecture that is part of the TOGAF standard.
- **application architecture**—The architecture of a specific application.
- **architect**—One whose responsibility is the design of an architecture and the creation of an architectural description.
- **architectural artifact**—A specific document, report, analysis, model, or other tangible that contributes to an architectural description.
- **architectural description**—A collection of products (artifacts) to document an architecture.
- **architectural framework**—A skeletal structure that defines suggested architectural artifacts, describes how those artifacts are related to each other, and provides generic definitions for what those artifacts might look like.
- **architectural methodology**—A generic term that can describe any structured approach to solving some or all of the problems related to architecture.
- **architectural process**—A defined series of actions directed to the goal of producing either an architecture or an architectural description.
- **architectural taxonomy**—A methodology for organizing and categorizing architectural artifacts.
- **architecture**—The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution (from IEEE-1471-2000).
- **business architecture**—An architecture that deals specifically with business processes and business flow.
- **business reference model (BRM)**—An FEA term that gives a business view of the various functions of the federal government.
- **business services segment**—An FEA term that refers to a *segment* that is foundational to most, if not all, political organizations, such as financial management.

- **CIO**—Chief Information Officer, the executive in charge of information technology in a corporation.
- **CIO Council**—A council consisting of CIOs from each of the federal governmental agencies that coordinates work related to common interests.
- **Clinger-Cohen Act of 1996**—See *Information Technology Management Reform Act*.
- **common-systems architectures**—A TOGAF term referring to an architecture that is common to many (but not all) types of enterprises, in contrast to *foundation architectures* and *industry architectures*.
- **component reference model (CRM)**—An FEA term that gives an IT view of systems that support business functionality.
- **data architecture**—The architecture of the data (typically stored in databases) owned by the enterprise.
- **enterprise architect**—An architect who specializes in enterprise architectures.
- **enterprise architecture**—An architecture in which the system in question is the whole enterprise, especially the business processes, technologies, and information systems of the enterprise.
- **enterprise service**—An FEA term referring to a well-defined function that spans political boundaries, such as security management.
- **FEA**—See *Federal Enterprise Architecture (FEA)*.
- **FEAF**—See *Federal Enterprise Architectural Framework (FEAF)*.
- **FEAPMO**—The organization within the OMB that owns and administers the Federal Enterprise Architecture.
- **Federal Architecture Program EA Assessment Framework**—A benchmark used by the OMB to measure the effectiveness of governmental bodies in using enterprise architecture.
- **Federal Enterprise Architectural Framework (FEAF)**—An enterprise-architectural framework used by the U.S. federal government to describe how the various governmental agencies and their IT systems are related to each other.
- **Federal Enterprise Architecture (FEA)**—An architectural description of the enterprise architecture of the U.S. federal government that includes various reference models, processes for creating organizational architectures that fit in with the federal enterprise architecture, and a methodology for measuring the success of an organization in using enterprise architectures.
- **foundation architecture**—A term used by TOGAF to refer to the most generic of architectures that can be used by any IT organization, in contrast to *common systems architectures*.
- **GAO**—See *General Accountability Office (GAO).*
- **Gartner**—An IT research and advisory organization.
- **gateway**—A transfer point of an autonomous system from which messages from the outside world are received or through which messages to the outside world are sent.
- **General Accountability Office (GAO)**—A branch of the U.S. Government that is responsible for monitoring the effectiveness of different organizations within the U.S. Government.
- **industry architecture**—A TOGAF term that refers to a architecture that is common to most enterprises within an industry, in contrast to a *common-systems architecture* and an *organizational architecture*.
- **Information Technology Management Reform Act**—An act passed by the U.S. Congress in 1996 that requires all governmental organizations to use effective strategies and frameworks for developing and maintaining IT resources.
- **OMB (Office of Management and Budget)**—Part of the Executive Office of the President of the U.S. that serves the function of presidential oversight on federal agencies.
- **The Open Group Architectural Framework**—See *TOGAF (The Open Group Architectural Framework) 8.1*.

- **organizational architecture**—A TOGAF term that applies to an architecture that is specific to a particular organization, in contrast to an *industry architecture*.
- **performance reference model (PRM)**—An FEA term that gives standard ways of describing terms related to measuring value.
- **Return on Investment (ROI)**—A measure (in percent) of the business value of a project, based on the increase in profit (either because of increased income or decreased expenses) divided by the cost of the project. For example, a project with a cost of $100,000 that returned $200,000 in increased profit has an ROI of 200 percent.
- **ROI**—See *Return on Investment (ROI)*.
- **segment**—An FEA term that refers to a major line-of-business functionality, such as human resources, that might be shared across organizations.
- **standards information base (SIB)**—A TOGAF term that refers to a collection of information about standards, particularly in the area of open-source.
- **TAFIM (Technical Architecture Framework for Information Management)**—An architectural framework developed by the Department of Defense and officially discontinued in 2000.
- **technical architecture**—Usually refers to the architecture of the technical infrastructure within which applications run and interact.
- **technical reference model (TRM)**—Part of TOGAF, a reference model that gives a common language for various pieces of IT architecture. This term is also used for a similar meaning within *FEA*.
- **TOGAF (The Open Group Architectural Framework) 8.1**—An architectural methodology that is controlled by The Open Group.
- **Zachman Framework for Enterprise Architectures**—An architectural framework in which an enterprise is modeled as 30 or 36 cells, each of which represents an intersection between a stakeholder perspective and an abstraction.

References

- [01] IEEE Standard 1471-2000: IEEE Recommended Practice for Architectural Description of Software-Intensive Systems.
- [02] Zachman, J.A. "A Framework for Information Systems Architecture." *IBM Systems Journal*, Volume 26, Number 3, 1987.
- [03] U.S. Department of Defense. *Technical Architecture Framework for Information Management (TAFIM) Volumes 1-8*. Version 2.0. Reston, VA: DISA Center for Architecture, 1994.
- [04] Clinger-Cohen Act of 1996 (PL 107-347) (See THOMAS (Library of Congress).)
- [05] The Chief Information Officers Council A04. *Federal Enterprise Architecture Framework Version 1.1*. September 1999.
- [06] Information Technology: The Federal Enterprise Architecture and Agencies' Enterprise Architectures Are Still Maturing, GAO Testimony Before the Subcommittee on Technology, Information Policy, Intergovernmental Relations and the Census, Committee on Government Reform, House of Representatives. May 19, 2004.
- [07] Information Technology: FBI Is Taking Steps to Develop an Enterprise Architecture, but Much Remains to Be Accomplished. GAO-05-363. September 9, 2005.
- [08] DOD Business Systems Modernization: Long-Standing Weaknesses in Enterprise-Architecture Development Need to Be Addressed. GAO-05-702. July 22, 2005
- [09] Homeland Security: Progress Continues, but Challenges Remain on Department's Management of Information Technology, Testimony Before Congressional Subcommittees,

Randolph C. Hite, Director, Information Technology Architecture and Systems Issues. GAO. March 29, 2006

- [10] Business Modernization: Some Progress Made Toward Implementing GAO Recommendations Related to NASA's Integrated Financial Management Program. GAO-05-799R. September 9, 2005
- [11] "framework". *The American Heritage Dictionary of the English Language*. Fourth Edition. Boston, MA: Houghton Mifflin Company, 2006.
- [12] "taxonomy". *The American Heritage Dictionary of the English Language*. Fourth Edition. Boston, MA: Houghton Mifflin Company, 2006.
- [13] Zachman, John A. "The Framework for Enterprise Architecture: Background, Description and Utility." Zachman Institute for Framework Advancement (ZIFA). Document ID: 810-231-0531
- [14] O'Rourke, Carol, Neal Fishman, and Warren Selkow. *Enterprise Architecture Using the Zachman Framework*. Boston, MA: Course Technology, 2003. ISBN: 0-619-06446-3
- [15] Interview with John Zachman by Roger Sessions, Editor-in-Chief, in *Perspectives of the International Association of Software Architects*, Issue 6.
- [16] Zachman, J.A. "A Framework for Information Systems Architecture." *IBM Systems Journal*, Volume 26, Number 3, 1987.
- [17] Zachman, J.A., and J.F. Sowa. "Extending and Formalizing the Framework for Information Systems Architecture." *IBM Systems Journal*, Volume 31, Number 3, 1992.
- [18] Zachman, J.A. "A Framework for Information Systems Architecture." *IBM Systems Journal*, Volume 26, Number 3, 1987.
- [19] www.opengroup.org
- [20] www.opengroup.org/togaf
- [21] Perks, Col, and Tony Beveridge. *Guide to Enterprise IT Architecture*. New York, NY: Springer, 2003. ISBN: 0-387-95132-6
- [22] www.opengroup.org/togaf/cert
- [23] Perks, Col, and Tony Beveridge. *Guide to Enterprise IT Architecture*. New York, NY: Springer, 2003. ISBN: 0-387-95132-6
- [24] TOGAF Version 8.1.1.
- [25] "FEA Consolidated Reference Model Document Version 2.1," December 2006, published by the Federal Enterprise Architecture Program Management Office, Office of Management of Budget.
- [26] A Practical Guide to Federal Enterprise Architecture by the CIO Council, Version 1.0, February 2001.
- [27] "FEA Practice Guidance," December 2006, published by the Federal Enterprise Architecture Program Management Office, Office of Management of Budget.
- [28] "FEA Consolidated Reference Model Document Version 2.1," December 2006, published by the Federal Enterprise Architecture Program Management Office, Office of Management of Budget.
- [29] Ibid.
- [30] Ibid.
- [31] Ibid.
- [32] "The Data Reference Model, Version 2.0," November 2005, published by the Federal Enterprise Architecture Program Management Office, Office of Management of Budget.
- [33] "FEA Consolidated Reference Model Document Version 2.1," December 2006, published by the Federal Enterprise Architecture Program Management Office, Office of Management of Budget.
- [34] "FEA Practice Guidance," December 2006, published by the Federal Enterprise Architecture Program Management Office, Office of Management of Budget.

- [35] Federal Enterprise Architecture Program EA Assessment Framework 2.1, December 2006.
- [35] Ibid.
- [36] Bittler, R. Scott, and Gregg Kreizman. "Gartner Enterprise Architecture Process: Evolution 2005." October 21, 2005. Gartner ID: G00130849
- [37] James, Greta A., Robert A. Handler, Anne Lapkin, and Nicholas Gall. "Gartner Enterprise Architecture Framework: Evolution 2005." October 25, 2005. Gartner ID: G00130855

About the author

Roger Sessions is the CTO of ObjectWatch. His ObjectWatch Newsletter is now in its 13th year of publication. He has written six books, including *Software Fortresses: Modeling Enterprise Architectures* (Addison-Wesley, 2003), and many articles. He is on the Board of Directors of the International Association of Software Architects (IASA), is Editor-in-Chief of Perspectives of the International Association of Software Architects, and is a Microsoft-recognized MVP in Enterprise Architecture. Roger holds multiple patents in both software and enterprise-architecture process. He has given talks at more than 100 conferences around the world, covering a wide range of topics in enterprise architecture. Learn more about Roger Sessions and ObjectWatch at www.objectwatch.com.