Winter 12-9-2003

Contextual Criteria to Select a Framework for Enterprise Architecture

Ali Fatolahi

Fereidoon Shams

Contextual Criteria to Select a Framework for Enterprise Architecture

Authors :

Ali Fatolahi

MSc Student in Software Engineering

Electrical and Computer Engineering Faculty

Shahid Beheshti University

Fereidoon Shams

Assistant Professor in Computer Department

Electrical and Computer Engineering Faculty

Shahid Beheshti University

Contacts

Address : Mr. Ali Fatolahi, 3rd Floor, No. 104, Malikian DeadEnd, BaharShiraz St., Shariati Ave., Tehran, Iran.

Telephone Numbers : Days (at work) : (98) (21) (2808520), Nights (home) : (98) (21) (7510862).

Email : Ali_Fatolahi@noavar.com

Selected Track

Other Research Subjects

Contextual Criteria to Select a Framework for Enterprise Architecture

Abstract

Enterprise Architecture is considered to be an efficient tool to overcome on managerial problems, especially those which come from information technology environment of the enterprise. One of the steps must to be taken to have an Enterprise Architecture is to select an appropriate Framework. Frameworks are the most important components in any Enterprise Architecture. Those are essentially needed to create an integrated Enterprise Architecture. There are many frameworks proposed for specific use in certain enterprises. However there are some more general frameworks which are applied in different situations. These general frameworks are FEAF, TEAF and C4ISR. In this paper some criteria to select a framework are introduced. These criteria are based on the context for Enterprise, Architecture and Framework. Although all of the examples here, include only these three frameworks, the provided criteria are applicable on other frameworks in the area of Enterprise Archietcure.

Keywords

Framework, Selection Criteria, Enterprise Architecture, Zachman.

1. Introduction

Many of the concepts in the area of Enterprise Architecture could be found in other areas such as civilization or informatics, so it seems no further explanantion is needed. Unfortunately this is not true in this area and almost all of the concepts have to be redefined. This need comes from lack of unique definitions for some of them or this fact that most of these concepts have migrated from a domain to a new completely different domain. For instance *Architecture* that is a well-known keyword for most of us, has a distinct meaning in this new area.

What is an Enterprise?

There are many definitions for the term *enterprise* which could be categorized in two broad classes. The first class includes those who see an enterprise from the viewpoint of its information systems. Mary Johnson and Larry Whitman of the University of Texas Automation & Robotics Institute, define an enterprise as *a complex system of cultural, process and technology components engineered to accomplish organizational goals* [1].

The second class includes those who define an enterprise from an organizational viewpoint. In *TEAF documents*, an enterprise is described as *an organization supporting a defined business scope and mission. An enterprise is comprised of interdependent resources (people, organizations, and technology). These resources must coordinate their functions and share information in support of a common mission (or set of related missions)* [2].

We may categorize the views in defining an enterprise into *Systematic View* and *Organizational View*. Although the second one is in more interest in the enterprise architecture area but both views could be seen equally important. This is true specially when we notice that an architecture is the bridge connecting them to each other :

Figure 1 : Architecture maps organization into information systems.

What is an Architecture?

Originally architecture comes from civilization. It is used in the area of informatics today in some different ways but the way we deal with is *enterprise archietcure*.

Zachman offers enterprise architecture to be *the set of primitive, descriptive, artifacts that constitue the knowledge infrastructure of the enterprise* [3]. However this is a general definition could be applied on many areas, there are more specific definitions such as one which presented by *CIO*[1]. EA[2] is *a strategic information asset base that defines the business, information necessary to operate the business, technologies necessary to support the business operations, and transitional processes for implementing new technologies in response to the changing needs of the business.*

---

[1] Chief Information Officers Council in the United States

[2] Enterprise Architecture (will be used this point forward)

Ultimately an architecture effort will result in an IT environment for the enterprise or conforms its current IT environment to a new one. Therefore, any enterprise architecture may have up to three time-phased views [5]:

- *Baseline or Current Architecture* – Describes the current state of the environment (Often called "As-Is").
- *Target Architecture* – Describes the future state of the environment (Often called "To-Be").
- *Transition Architecture* – Includes: Business Improvement Efforts, Technology Migration Strategies, Project Development Initiatives and Deployment Plans which transform the enterprise architecture from its current state to a target state.

What is a Framework?

Framework is considered to play the most important role in enterprise architecture. For any architecture project it is vital to select a framework. It guarantees final products to be interoperable and supporting for business needs.

Generally speaking a framework *is a comprehensive, logical structure for descriptive representations (i.e., models) of any complex objects* [4]. Specifically EA framework 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* [6]. Frameworks steer architecture, organize architecture products and promote interoperability within the products during an EA project.

Generally, when we speak of frameworks in the area of EA, we mean Zachman framework, which is the fundamental framework in this area. It is first introduced by *Zachman* (*1987*) and then extended and formalized by *Sowa & Zachman* (*1992*).

Zachman framework is a table, consisting of six rows and six columns. There are two different ideas for rows and columns (*Zachman, 1987*):

1. Rows represent different perspectives of the different participants in building enterprise architecture.
2. Columns are different ways in which we describe the same product for different purposes.

There is a cell, at the cross point of each row and each column, which contains a unique model.

Figure 3 depicts Zachman framework, as it is known today. (Model names are ignored for the sake of brevity):

	Data	Function	Network	People	Time	Motivation
Scope						
Business						
System						
Technology						
Detailed Specifications						
Functional Entepris						

Figure 3 : Zachman Framweork

2. EA Frameworks

One of the steps must to be taken in any enterprise architucture effort would be selecting a framework. Using a framework will ensure uniformity and standardaization when migrating and integrating information systems [9].

There are many frameworks introduced in order to build an enterprise architecture, from which those that are candidates in doing federal architectures are selected here to describe about and use as samples.

Most federal organizations (in the US) have standardized on the following three frameworks [9]:

Federal Enterprise Architecture Framework (FEAF) [4]: The Clinger-Cohen Act of 1996 mandated that Federal Agencies develop and maintain an enterprise IT architecture. The FEAF was established in 1999 by the Chief Information Officers (CIO) in response to this mandate. The purpose of the FEAF is to facilitate shared development of common processes and information among Federal Agencies and other government agencies [10].

In designing the Framework, the CIO Council identified eight components vital for developing and maintaining the Federal Enterprise Architecture, then drilled down to a further granularity of detail. These components are [10]:

- **Architecture Drivers** are external stimuli that cause the Federal Enterprise Architecture to change.

- **Strategic Direction** ensures that changes are consistent with the overall Federal direction.
- **Current Architecture** is the current state of the enterprise.
- **Target Architecture** is the target state for the enterprise within the context of the strategic direction.
- **Transitional Processes** are those processes that apply the changes from the current architecture to the target architecture, in compliance with architecture standards (such as various decision making or governance procedures, budgeting, engineering change control, etc).
- **Architectural Segments** are subsets or a smaller enterprises within the total Federal Enterprise.
- **Architectural Models** provide the documentation and the basis for managing and implementing changes in the Federal Enterprise.
- **Standards** (some of which may be made mandatory), voluntary guidelines, and best practices, all of which focus on promoting interoperability.

The FEAF consists of four levels, the first level providing a high level description of the above components, and the next three levels describing these components in increasing details [10].

More details will be gained via breaking the components into two layers : *Business* and *Design* at level II, then dividing *Design* layer into three sub-layers : *Data*, *Application* and *Technology*.

The idea of breaking components in details in a way such that mentioned above mainly comes from EAP methodology introduced by *Spewak* (*1992*).

The fourth level also provides a logical structure for classifying and organizing the artifacts. This logical structure is actually a tailored version of the Zachman Framework [10] consisting its first three columns and five rows.

Treasury Enterprise Architecture Framework (TEAF) [2]: TEAF is an EA framework for the department of treasury and its agencies. It is aligned with the FEAF [11].

At the heart of the TEAF is the TEAF Matrix which provides a customized version of Zachman table [11]. This matrix is a 4 by 4 matrix shown in figure 5 :

	Functional View	Information View	Organizational View	Infrastructure View
Planner Perspective				
Owner Perspective				
Designer Perspective				
Builder Perspective				

Figure 4 : The TEAF Matrix

Both FEAF and TEAF adapt NIST[1] model. It has been advised to be a basic model for information architecture consisting of five layers. However both of them customize this model for theire specific purposes. Layers in NIST (Firstly introduced as sub-architectures in DOE[2] methodology) are [12]:

- **Business:** Organizations, Customers, Business Functions, Standards, Policies and Procedures.
- **Information:** Information Flow, Interal, External, Content, Format and Presentation.
- **Applications:** Automated Data Processing, Manual Systems, Procedures and Logical Data Structure
- **Data:** Physical DB Design, DB and File Structures and Data Dictionaries
- **Technology:** Computers, Facilities, Communications Network and Security Infrastructure.

Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR) Framework [13]: In December 1997, the DoD[3] published its C4ISR Architecture Framework. This framework applies to all branches of the armed services and includes the numerous major and subordinate commands, field organizations, and task forces within each service [9].

The framework has four main parts: definitions of three standard views; products; references; and high-level

[1] National Institute for Standards and Technologies
[2] Department Of Energy
[3] Department Of Defence

guidance in how to use the framework to describe an anrchitecture [15].

"View" plays a role as "Layer" in the NIST model or as "Perspective" in the Zachman framework. These three architectural views are :

- **Operational Architecture View** describes the tasks and activities, the *operational nodes* and the information flows between nodes that are required to accomplish or support an operation.
- **Systems Architecture View** translates the required degree of interoperability into a set of system capabilities needed, identifies current systems that are used in support of the operational requirements and facilitates comparison of current/postulated system implementations with the needed capabilities.
- **Technical Architecture View** articulates the criteria that govern the implementation of required system capabilities [9, 15].

These three views have explicit linkages between themselves in order to maintain consistency and integration of the architecture [9, 15].

C4ISR has this advantage that provides some *Universal Reference Resources*, which serve as sources for guidelines, and attributes that must be consulted while building architecture products [13].

Also its other advantage is a *Six-Step Architecture Process*. This is described briefly in the framework document as the fundamental steps to build architecture in accordance with the Framework [13].

3. Framework Selection

Selecting a framework must be a step in building enterprise architecture. This is specially true for those enterprises which have no architecture before. Some enterprises may have frameworks selected before. It is important to note that although having already a framework, architects must evaluate the current framework in order to understand its ability to response to the drivers forcing the enterprise to be changed.

In [9] some framework selection criteria are mentioned for federal agencies. These are:

- **Area of Policy**
  - Regularity and legislative direction.
  - Agency policy.
  - Compatibility needed with another Agency or joint policy
- **Enterprise**
  - Context for the enterprise: e.g., subordinate to a larger enterprise, closely related to another enterprise.
  - Experience with a particular framework.
  - Mandates and drivers: e.g., emphasis on business versus infrastructure or operational versus technical issues
- **EA Products**
  - Priorities, intended uses and desired level of detail: e.g., large-scale modernization versus stable IT environment.
  - Resource and schedule constraints on modeling efforts.
  - Availability of existing architecture products.

Among these criteria, the context for the enterprise is in our interest here. Also we focus on the context for architecture and show that both have some counterparts in the context for framework.

We provide four classification schemes here, first two of which are about *the Context for the Enterprise* and two last ones are about *the Context for the Architecture*.

Organizational vs. Operational: Of course each organization, have many operations to do. When we talk about operations here we don't mean any activities an organization performs. Operations are distinct from other activites in two following ways:

1. Typically an operation is not a simple office or customer related service. It is as critical as there will be no more need to the organization if it doesn't perform the opeartion.
2. An operation usually invokes many more resources than other activites. The resources might include human, machines, time, fund and etc.

Examples of operations are: Military Operations, Fire Fighting Operations, Crisis Management, Software Production Process, a Soccer Match and so on.

Regarding this definition there are two broad categories for enterprises: *Organizational Enterprise* vs. *Operational Enterprise*. Also we may name them *Structural Enterprise* vs. *Behavioral*

Enterprise respectively. Note that these two types of enterprises are often mixtured in a unique organizational unit.

In order to provide better understanding of this point, refer again to our samples of an operation and consider folowing descriptions:

- A military operation (Operational Enterprise) is done by a defense related deapartment (Organizational Enterprise).
- A software production process is done by a software engineering company.
- A fire fighting operation is done by the fire guard.

In the same way we would see that frameworks themselves could be seen as having more operational or organizational emphasis. It means that some frameworks form an architecture which shows the enterprise as an organization whereas others establish an architecture that depicts the enterprise as an operation.

Having enterprises categorized in this way, we could name frameworks as the same way. So we have *Structural (Organizationl) Framework* and *Behavioral(Operational) Framework*.

The best sample of an operational framework is C4ISR. It focuses on the operational nodes and elements at the top level of its architecture and so is well organized to build architecture for an operation.

On the other hand, we have FEAF & TEAF, both accepting NIST structure as their reference architecture pattern. As we have shown before it begins with the *Organization* as its top level. These are structural frameworks.

To finalize examples, now we offer following proposals in order to build architecture:

- To build architecture for the organizations such as Department of Defense, select FEAF, and to build architecture for its military operations, select C4ISR.
- To build architecture for a software engineering company, select TEAF, and to build architecture for its projects development process, select C4ISR.

As a result we could say that if the enterprise you want to build architecture for, is an operational one, you may choose an operational framework such as C4ISR. But what to do, in the case of organizational ones?

Nested vs. Flat: Every organization has many subunits in its structure. Some of these subunits might be seen as enterprises themselves. For example in a government there are some ministries (or departments) that are enterprises themselves such as Department of Defense or Ministry of Education. We name such enterprises as *Nested Enterprise* and we mean it includes some other enterprises.

For an enterprise to be considered as a nested one, just having other enterprises included is not enough, but also these sub-enterprises must be devised to be engaged with architecture. If sub-enterprises have not been considered in the architecture or have been seen as a simple black box with some input/output lines, we could not suppose the main enterprise as a nested one.

If any of two conditions mentioned above are not present then the enterprise may be called a *Flat Enterprise*.

Similar to the enterprises, frameworks may be *nested* or *flat*. Two of three frameworks suggested by CIO for federal organizations are good examples for these two types of frameworks: FEAF & TEAF.

FEAF is considered to be a *Nested Framework* for its *Architectural Segments* component. As it has been shown before those are sub-organizations or sub-divisions of the major business in the federal organization. These segments are considered to be other enterprises within the federal enterprise [4].

Although we could ignore segments in FEAF for a flat enterprise, there is a more appropriate framework for flat enterprises: It's TEAF. TEAF is well-matched with FEAF, but it has no architectural segments included. So we might consider the TEAF as a *Flat Framework*.

We may use C4ISR for building architecture in a flat enterprise, however, note that a flat enterprise seems to be an organizational enterprise more than an operational one.

Level of Details: When planning for architecture projects, we might decide about architectural products, for example which of them are required? And in which level of details? So architectures (after planned) are not as the same to each other in the level of details for final products.

Furthermore frameworks are not equally focused on their products either. In some frameworks such as C4ISR, there are many descriptions on how to create products? Steps to be taken? What are included in each product? There are even attributes for each product fully described. Such a framework is suitable for implementing a final IT environment. We call any framework like this a *Low-Level Framework.*

To get more familiar with this concept let us verify our three accepted frameworks in this manner. As it is said before the C4ISR is a low-level one. Not only there are a lot of descriptions about its products documented officially, but also there are many researches and work done on tools and methodolgies for building architecture using C4ISR framework. TEAF could also be seen as a low-level framework but not as low-level as C4ISR. It defines attributes for its products but it has no step-by-step method to achieve them.

On the other hand we have FEAF as a high-level framework, that is not engaged with the details of products and how to produce them.

If your goal is to provide a real working IT environment you may prefer C4ISR to TEAF or FEAF even if the subject enterprise be an organizational one. In other words you as the architect should have the art of ranking these factors against each other.

States of the Architecture: This could affect the architecture effectively. Remember three states which were mentioned before for any architecture. However for an architecture it is not obligatory to have exactly those three states. It may only concern with the current (or target) state of the IT environment.

In such a way, it is not true that all of the frameworks support the three possible states of the architecture. For example C4ISR only deals with one state (current or target). Although it has some products to keep some notes on the future technologies or activites, but obvoiusly these are not a complete set of architectural products.

On the other side both FEAF and TEAF support the three states for any architecture as mentioned before. This is specially true for FEAF that supprots any of these three states with a certain component.

So if the project is planned to check the current state and provide a migration plan to achieve an offered future state you must select each of the FEAF or TEAF.

4. Conclusion

Having classified framework selection criteria, we have reached to some properties, an enterprise or an architecture may have. As an architect you may choose the framework which its properties match with the one for the enetrprise and the architecture in the best way.

As we said before in some situations we may have two or more properties occurred simultaneously, while they seem to be contradicting each other. As it was mentioned erarlier, the role of the architect is very important in such cases.

However a notable point to focus is that these situations may occure rarely. This is due to the most important part of our taxonomy that descripes an enterprise to be structural or behavioral.

An organization often comprises other orgaizations but operations included in an operation are smaller than to be seen as an enterprise. So an operational enterprise may not be seen as a nested one.

Furthermore, we are more interseted in details for operations than organizations. This is because operational enterprises are often *more mission critical* than organizational enterprises. Any mistake in these enterprises could result in *immediately unpredictable crashes*.

These doesn't mean structural enterprises are not mission critical. It means their failures could be recovered more easily. While for operational ones these failures could not be recovered at all. As a consequence most of the times, low-level architectures is needed for operationl enterprises not for organizational

enterprises. Table 1 summarizes all the criteria discussed in this paper:

	Enterprise				Architecture			
	Structural	Behavioral	Flat	Nested	Low-Level	High-Level	Single-State	Triple-State
FEAF	√			√		√		√
TEAF	√		√		√			√
C4ISR		√	√		√		√	

So far we suggested a taxonomy of enterprises in this paper. For an enterprise, to be fitted in this taxonomy easily, we need to define an enterprise in a new manner:

An enterprise is any collection of sub-units, inter-connected to each other through the flow of information.

This seems to be a very simple definition, but it could be applied on both types of the enterprises discussed earlier as well. We may expect an enterprise to be necessairly a giant object but if we accept architectures apply on enterprises then we may notice to *Zachman*, *Inmon* and *Geiger* who believe that an architecture is used to describe an *object* [7]. Also as it was said by *Zachman* and *Sowa* (*1992*) the logic and rules of the framework *can be used for structuring the description of any complex object*.

Future Work

Many problems are exist in this area, need to be solved yet. Following this work, we will consider on other framework selection criteria, trying to represent a full taxonomy of all the criteria.

Also we think about more contextual aspects of enterprises, architectures and frameworks as the selection criteria. Some of these aspects are:

- Conceptual vs. Technical
- Homogeneous vs. Heterogenous
- Dynamic vs. Static
- Single Dimension vs. Two Dimension

References

1. System Architect v8.0 2001 Manuals. *Building an Enterprise Architecture: The Popkin Process, Version 1.0*. Popkin Company. 2001.
2. Chief Information Officers Council. *Treasury Enterprise Architecture Framework, Version 1*. Department of the Treasury. July 2000.
3. John A. Zachman. *Enterprise Architecture Artifacts vs Application Development Artifacts*. Zachman International.
4. Chief Information Officers Council. *Federal Enterprise Architecture Framework, Version 1.1*. September 1999.
5. Chief Informations Officers Council. *Architecture Alignament and Assesment Guide*. October 2000.
6. John A. Zachman. *The Framework for Enterprise Architecture: Background, Description and Utility*. Zachman International.
7. Larry Whitman, Kartik Ramachandran, Vikram Ketkar. *A Taxonomy of a Living Model of the Enterprise*. Proceedings of the 2001 winter simulation conference
8. J. A. Zachman. *A Framework for Information Systems Architecture*. IBM Systems Journal. 1987.
9. Chief Information Officers Council. *A Practical Guide to Federal Enterprise Architecture, Version 1.0*. February 2001.
10. http://government.popkin.com/frameworks/feaf.htm
11. http://government.popkin.com/frameworks/teaf.htm
12. US Department of Energy. *Information Architecture Definition and Documenation Methodology*. February 6, 1997.
13. Department of Defense, C4ISR Architecture Working Group, *DoD C4ISR Architecture Framework, Version 2.0*. 18 December 1997.
14. http://government.popkin.com/frameworks/c4isr.htm
15. P. Kathie Sowell. *The C4ISR Architecture Framework : History, Status and Plans for Evolution.*
16. Spewak, Steven H., 1992, *Enterprise Architecture Planning, Developing a Blueprint for Data, Applications and Technology*, John Wiley & Sons, Inc.