Towards a Framework for Enterprise Architecture Frameworks Comparison and Selection

Saber Abdallah
Faculty of Computers and Information, Cairo University
Saber_abdallah@hotmail.com

Galal Hassan Galal-Edeen
Faculty of Computers and Information, Cairo University & The British University in Egypt
G.Galal@fci-cu.edu.eg / Galal@bue.edu.eg

Abstract

A number of Enterprise Architecture Frameworks (EAF) do exist, which are sometimes different in approaches and at other times are based on other frameworks. This paper presents a survey of the current state of EAF. Comparisons will be held among some selected frameworks based upon different criteria. These criteria will include the current market share of the selected frameworks, goals, inputs, outcomes, strengths, weaknesses and frameworks supported tools. Conclusion is made concerning EAF selection.

Key words: Enterprise Architecture Frameworks (EAF), Information Systems Architecture Frameworks comparisons, Enterprise Architecture Frameworks selection criteria.

1. Introduction

This paper starts by outlining the basic terms of architecture, enterprise, framework and Enterprise Architecture Frameworks (EAF). Background on different frameworks and a classification of these frameworks will be presented. Different criteria for EAF will be discussed. At the end of this paper we will conclude how to choose an EAF regarding these criteria and the current state of EAF. Also recommendations for further research in this area will be presented. In this paper we will use EAF and information system frameworks interchangeably.

1.2 Basic definitions

Architecture: In this paper, we will use the IEEE Standard 1471-2000 for the term architecture. It is defined as “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.” [1], where:

− fundamental organization means essential, unifying concepts and principles.

Environment: The framework is fundamentally a conceptual model [2]. A framework is a structure within which the key components of the architecture and the relationships between these components are defined. A framework [3]:

− Helps organize thinking about the architecture
− Provides a description of the architectural artefacts
− Helps ensure that everyone is using the same set of semantics and presents them to the group of stakeholders interested in the contents of the architecture.
− Provides a way to communicate the architecture
− It level-sets stakeholders about the contents of the architecture by providing common definitions and concepts
− It shows the relationships between business and technology elements, ensuring that there is coherence between all elements and that every business element can map to a corresponding element in the technical architecture and, similarly, that technical elements can be seen as supporting key business requirements.

Enterprise: Any collection of organizations that has a common set of goals and/or a single bottom line [4]. In that sense, an enterprise can be a government agency, a whole corporation, a division of a corporation, a single department, or a chain of geographically distant organizations linked together by common ownership.

Enterprise architecture (EA): A coherent whole of principles, methods, and models that are used in the design of an enterprise's organizational structure, business processes, information systems, and infrastructure [5]. The most important characteristic of an EA is that it provides a holistic view of the enterprise.

Enterprise Architecture Framework (EAF):
The EAF is an instrument. When utilized it facilitates a holistic (cross-organizational, cross-functional, collaborative view of, delivery of and approach to the following key concepts [6]:
- Communication
- Organization around EA
- Mapping technical elements to business strategy
- Priority planning
- Asset and artifact management
- Service delivery
- Semantics
Which collectively, guide the organization in the development of the EA.
An EAF is comprised of a coupling of two major components: Methodology and Framework. Table (1) shows the main characteristics of both components.

<table>
<thead>
<tr>
<th>Methodology</th>
<th>Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>Describes standardized processes for developing the EA.</td>
<td>Standardized classification tool for the EA deliverables or artifacts. Describes the “What” of EA.</td>
</tr>
<tr>
<td>Describes standardized deliverables and processes for developing the EA</td>
<td>A place for everything and everything in its place</td>
</tr>
</tbody>
</table>

Table (1) Methodology and Framework characteristics

**Software Architecture:**

The software architecture of a system or a collection of systems consists of the important design decisions about the software structures and the interactions between those structures that comprise the systems. These design decisions support a desired set of qualities that the system should support to be successful. The design decisions provide a conceptual basis for system development, support, and maintenance [7]. It relates requirements, fixed system hardware, and infrastructure (i.e., COTS, GOTS) to software structures in order to demonstrate software effectiveness [8].

**Information System Architecture:**

An information system architecture typically encompasses an overview of the entire information system—including the software, hardware, and information architectures (the structure of the data that systems will use). In this sense, the information system architecture is a meta-architecture [2]. It relates the requirements and the external world to system/solution structures, including both hardware and software, so that the effectiveness of a system design concept can be communicated [8].

**View:**

It is a representation of one or more structural aspects of an architecture that illustrate how the architecture addresses one or more concerns held by one or more of its stakeholder [9].

**Viewpoint**

It is a collection of patterns, templates, and conventions for constructing one type of view. It defines the stakeholders whose concepts are reflected in the viewpoint and the guidelines, principles, and template models for constructing its views [9].

**Stakeholder:**

An individual, team, or organization (or classes of thereof) with interests in, or concerns relative to, a system [5]. Most stakeholders of a system are not probably interested in its architecture, but only on the impact of this on their concerns. However, an architect needs to be aware of these concerns and discuss them with the stakeholders, and thus should be able to explain the architecture to all stakeholders involved, who will often have completely different backgrounds.

2. Enterprise Architecture Frameworks

2.1 EAF history and taxonomies

EA gained popularity when John Zachman developed his framework for dealing with large information systems on 1987 [10]. Figure (1), [8] illustrates the historical relationships among several frameworks. We notice that most enterprise architecture frameworks have a common history and are built on refinements and add-ons of other frameworks. Zachman framework is the oldest and the father of the most frameworks. Different taxonomies are there for EAF. One of these is differentiations between two classes of frameworks: classic EAF, and federated EAF [11]. The federated enterprise architecture deals with creating integrated architectures. Examples of federated enterprise architectures are C4ISR architecture framework, the federal enterprise architecture framework and the treasury enterprise framework.

Another taxonomy has been put by ISO 15704 [12], it considered that there are two and only two types of architectures that deal with enterprise integration: System architectures (sometimes referred to as "type 1" architectures) that deal with the design of a system, e.g. the system part of an overall enterprise integration; Enterprise-reference projects (sometimes referred to as "type 2" architectures) that deal with the organisation of the development and implementation of a project such as an enterprise integration or other enterprise development programme.

In other words, type 1 architecture represents system or sub-system in terms of its structure and behaviours. The type 2 architecture is actually framework aiming at structuring activities/tasks
necessary to design and build a system. For example, Zachman’s architecture is type 2 architecture [13]. Outside the EA different taxonomies, we can find three main types of architecture. These three types are: EA, software architecture and system architecture, which previously defined. Figure (2), represents the Relationships among System Architecture, Software Architecture, and Enterprise Architecture.

Figure (1) Historical relationship among EAF
2.2 Enterprise Architecture survey

The most important survey report in EAF is published yearly since 2003 by Institute For Enterprise Architecture Developments (IFEAD). This report presents the results of the third electronic survey [14], its title is "Trends in Enterprise Architecture : How are Organizations Progressing?". The report is based on a 25 questions survey, addressing geographical aspects, branch aspects, EA implementations aspects as well about tools and methodologies used in Enterprise Architecture programs and the role of architects in organizations. 79 respondents filled in the EA 2005 survey. One of these issues is "What kind of Enterprise Architecture Framework does your organization use?". The following table represents the result of the survey question.

<table>
<thead>
<tr>
<th>Framework</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zachman Framework</td>
<td>0.25</td>
</tr>
<tr>
<td>Organization own</td>
<td>0.22</td>
</tr>
<tr>
<td>C4ISR, US Defense Architecture Framework</td>
<td>0.11</td>
</tr>
<tr>
<td>TOGAF, The Open Group Architecture Framework</td>
<td>0.11</td>
</tr>
<tr>
<td>Extended Enterprise Architecture Framework (E2AF)</td>
<td>0.09</td>
</tr>
<tr>
<td>FEAF, US Federal Enterprise Architecture Framework</td>
<td>0.09</td>
</tr>
<tr>
<td>Other</td>
<td>0.09</td>
</tr>
</tbody>
</table>

Table (1)

Enterprise Architecture Framework Market share

Table (1) shows the most used framework in 2005 is Zachman’s Enterprise Architecture Framework and organization’s own EAF came in the second. When dealing with such surveys, we should take into consideration the following points:

- Filling in the EA survey 2005 at the website of IFEAD was voluntary, so the results can be a little bit distorted by the fact that only people who are interested in Enterprise Architecture are taking the effort to fill in this survey.
- The rate of framework may differ from year to year, and other frameworks may be disappeared from the survey.

3. Enterprise Architecture Framework selection criteria

There are different criteria that can be used to help in selecting an EAF. These criteria can be categorized into
four sections, these section are goals, inputs, outcomes and miscellaneous as shown in the following subsections. Other criteria elements could be added to these sections, specially when taking into consideration the software frameworks architecture, which may be in some EAF included in it, such as TOGAF. These criteria will be defined only in each section under Other Criteria, but not included into comparison table.

3.1 Enterprise Architecture Goals

The following goals are common, independent of industry domain, architecture style and system size [15].
- Architecture Definition and Understanding – make use of standard terms, principles and guidelines for consistent application of the framework for the communication of architecture information to stakeholders.
- Architecture Process – employ a well-defined process to guide the construction of architecture.
- Architecture Evolution Support – employ processes and mechanisms that support systems evolution.
- Standardization – ensure development and architectural standards are maintained.
- Architecture Knowledge Base – provide consistent representation and repository of design and architecture design rationale.

Other Goal Criteria:
- Architecture Models – provide consistent standards to document architecture specifications for the planning, management, communication and execution of activities related to system development.
- Architecture Analysis – provide a set of viewpoints to guide the collection and analysis of information for making architecture choices.

3.2 Enterprise Architecture Inputs

Inputs represent information that architecture modelling considers. Typical inputs to architecture design activities are the following [15].
- Business Drivers – business goals, direction, principles, strategies and priorities
- Technology Inputs – strategic architecture direction including technology platforms, future architecture, systems interoperability and emerging technology standards.

Other Inputs Criteria
- Business Requirements – users’ requirements, functional requirements, data requirements and other business system related requirements
- Information System Environment – budget, schedule, technical constraints, resources and expertise, organization structure, other constraints, enterprise knowledge base.
- Current Architecture – current standards and infrastructure.
- Non Functional Requirements – some of these requirements are also referred to as Quality Attributes (QA) or Quality of Services (QoS). These requirements include availability, reliability, scalability, security, performance, inter-operability, modifiability, maintainability, usability and manageability.

3.3 Enterprise Architecture Outcomes

Outcomes represent results and deliverables. Typical outputs to architecture activities are the following [15].
- Business Model – describes business models, business requirements, business process, system roles, and policy statements.
- Transitional Design – provides designs and plans to support system transition and evolution.

Other Outcomes Criteria
- System Model – models major components of the system. To arrive at a system architecture model, major tradeoffs and design decisions are made. Future system enhancements are also taken into consideration.
- Information Model – contains data model, data transformation and data interface Computation Model – contains system functional description, system process flow, system operations, software components and interactions.
- Software Configuration Model – describes how software is packaged, stored, configured, managed and shared.
- Software Processing Model – describes how software processes, software threads and run-time environment are structured.
- Implementation Model – describes physical system structure such as operating environment, hardware components and networking components of the system. Models implementation processes such as installation, deployment, configuration and management.
- Platforms – describe platform software such as operating systems, hardware and networking components, protocols and standards.
- Non-functional Requirements Design – models the structure of the system to reflect design of nonfunctional requirements.
- Design Rationale – documents reasons of design based on analysis and tradeoffs that involve multiple dimensions of inputs.

3.4 Other Criteria

- Conformance - The modeling method should support the ability to define conformance testing criteria (of the implementation to the architecture specification). Otherwise, the "architecture specification" may or may not specify the architecture of the resultant system—no one will know for sure without provable tests. Some models provide compliant test criteria to ensure the architecture model is compliant with the modeling method [16].
- Clinger-Cohen Act (CCA) Compliance - The CCA specifies how the U.S. government is to plan, manage, and acquire information technology. This legislation
focuses on carrying through the information technology aspects of Government Performance and Results
(GPRA) [17]. CCA mandates the following:

- U.S. government agencies are to establish strategic performance goals for any information technology
  that supports the agency.
- Agencies are to achieve at least a five percent decrease in the costs incurred to operate and
  maintain IT systems, and a five percent increase in agency operational efficiency as a result of IT
  investments every year.
- Each federal agency is to have a chief information officer (CIO). The CIO is to help foster better
  technology investment, accountability, and decision making within the agency. The CIO is to
  implement capital planning and investment controls for IT acquisition and management, where
  performance outcomes are measured, analyzed, and reported (per GPRA).

Visualization tool - The visualization is a graphical of architecture or some part of architecture. Here we
investigate if the framework has a graphical tool or not. Different tools are available in the market, and each one
of them has its capabilities and supported framework(s), such as Allen Systems Group (ASG) Rochade,
Casewise Corporate Modeler, Computas Metis, IDS Scheer ARIS, MEGA International, Popkin Software
System Architect, Proforma ProVision and Ptech Enterprise FrameWork. A comparison among these
tools can be found in [18].

4. Frameworks

In this section we provide a high level comparison and analysis of three EAFs. Our selection for the three EAF is
based on table (1) of enterprise architecture framework market share. We select the high ranking (equal or over
11%). The three selected EAF contain the basic two classes of EAF: two classical and one federated. The
classical are Zachman and TOGAF, and the federated is C4ISR.

Table (2) provide an overview and comparisons of EAFs. The following selection are used to express the
framework status for supporting an element in the table: “Y”: If a framework explicitly supports an element.
“N”: If a framework does not support an element or there is no mention of that element in the documentation.
“P”: Where a framework partially supports or eludes to support an element.

The extent to which each EAF supports and interpret an element may differ even when they have the same values
in the same row. The following subsections will provide details analysis for each EAF described in table (2).

<table>
<thead>
<tr>
<th>Architecture Evolution Support</th>
<th>N</th>
<th>Y</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standardization</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Architecture Knowledge Base</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Inputs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Business Drivers</td>
<td>P</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Technology Inputs</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Outcomes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Business Model</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Transitional Design</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conformance</td>
<td>N</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td>CCA/Compliance</td>
<td>P</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td>Visualization tool</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

Table (2): Comparisons of EAFs

4.1 Zachman Information System Framework

4.1.1 Introduction

John Zachman is the "Father" of the Zachman Framework for Enterprise Architecture and Information Systems
Architecture. The primary goals of Zachman framework are for enterprise architecture analysis and modelling and it is also concerned with perspectives of constructing an information system. A perspective is a row in a table representing how a stakeholder in a project team would view the system [15].

Table (3): The Zachman Framework

The framework architecture is a matrix of 30 cells, where the rows represent the different perspectives and the
columns the things viewed from each perspective. The examples suggested by Zachman for each cell are shown,
slightly abbreviated table (3) [19].

Each Cell is an outcome of an architecture activity based on an aspect of a system for a particular group of people
and a singular focus on one aspect of the architecture such as data, process or location.

ZACHMAN has six rules, these rules helps in building the EAF. The six rules are as follows [20]:

**Rule 1:** Columns have no order

**Rule 2:** Each column has a simple, basic model

**Rule 3:** Basic model of each column is unique

**Rule 4:** Each row represents a distinct view of
Rule 5: Each cell is unique
Rule 6: Combining the cells in one row forms a complete description from that view

4.1.2 Zachman Framework Discussion
Although Zachman has been referred to and used in frameworks such as FEAF, DoDAF and TOGAF, it doesn't support most of the comparison criteria. Below some of Zachman framework drawbacks.

− Zachman uses an analogy from classical building architecture and military aircraft manufacturing to help define information systems architecture (ISA). This approach is useful when the final outcome is a “system” because ISA helps in understanding the components that make up each individual application or system. Information Framework (IFW) uses the alternative analogy of a “city plan” rather than a building plan. It provides an effective way to gradually develop a complete “city” of information. This compilation includes information about individual applications and systems, as well as information output from other types of projects such as strategic planning or business process reengineering. [21].
− The relation between one cell and another is not defined. That is, the consistency of the artefacts is not addressed. There is no discussion as to the consistency from one cell to another, or from one row to another, or from one column to another [16]. A solution was provided by [22] to facilitate using of Zachman and relates in a consistency way among cells.
− Semantic behaviour, and how that behaviour affects the functioning of the components and their interactions, is not addressed [16].
− The Zachman framework is not a standard written by a professional organization, so no explicit compliance rules have been published. However, if the framework is used in its entirety and all the given relationship rules are followed, then compliance can be assumed by default [8].
− Many visualization tools support Zachman are available, such as Popkin and IDS Scheer ARIS.

4.2 The Open Group Architecture Framework (TOGAF)

4.2.1 Introduction
TOGAF developed by the Open Group in 1995. TOGAF is based on the Technical Architecture Framework for Information Management (TAFIM), which developed by the Department of Defence (DoD). Version 8.0 of TOGAF is called the 'Enterprise Edition' and is dedicated to EA. The main components of the TOGAF are described in figure (3) [5].

Figure (3): TOGAF main components

- Architecture Development Method (ADM), which is a specific process defined by TOGAF for the development and maintenance of the organization’s technical architecture [23]. The ADM consists of seven major phases: initiations and framework, baseline description, target architecture, opportunities and solutions, migration planning, implementation, and architectural maintenance. The ADM is considered to be the core of TOGAF, and consists of a stepwise cyclic approach for the development of the overall enterprise architecture [5].
- The TOGAF Enterprise Continuum, which aggregates two continua, the architectural continuum and the solutions continuum. The architectural continuum describes the generalization / specialization of architectural components, whereas the solutions continuum illustrates the actual implementation of the components [23], and TOGAF Foundation Architecture that contains the Technical Reference Model, The Open Group's Standards Information Base (SIB), and The Building Blocks Information Base (BBIB) [5].
- The TOGAF Resource Base, which is a set of resources including guidelines templates and background information to help the architect in the use of the ADM.

TOGAF has a list of recommended architecture views as part of its resource base. These architecture views can be divided into two main views, as follows:
− Business architecture views, which addresses the concerns of the users of the system, and describes the flows of business information between people and business processes (e.g., People View, Process View, Function View, Business Information View, Usability View, Performance View) [5].
− Technical architecture views, which include:
o Engineering views, which address the concerns of System and Software Engineers and include Security, Software engineering, Data, System engineering, and Communications engineering view.

o Operations views, which address the concerns of Operators, Administrators and Managers and include Security, Software, Data, Computing / Hardware, Communications view.

o Acquirers’ views which address the concerns of procurement personnel responsible for acquiring the Commercial Off-The-Shelf (COTS) software and hardware and include Building blocks cost, Standards view.

4.2.2 The Open Group Architecture Framework discussion

TOGAF has "Y" for all of the comparison criteria, except for (CCA) Compliance, it gets "P". TOGAF is like Zachman in its compliance with CCA. CCA compliancy can be it can be achieved using TOGAF when following the compliancy rules [8]. From our point of view, TOGAF has many strengths and may be the leader of Enterprise Architecture Framework. The strengths are:

− TOGAF is supported by an open strong committee, which anyone can participate on and provide a fully published product of its results.

− Proven Method—TOGAF offers a proven method that is the result of years of research and development by the world's leading enterprise architects.

− Common Vocabulary—TOGAF guides architects in using a standard taxonomy for business, information systems, and technology modelling. This shared vocabulary means that everyone in an organization can read and understand the information.

− Communication—Models of the enterprise architecture give visual representation to business concepts, and, when published on the corporate intranet, disseminate knowledge of the business to the workforce.

− Command Decisions—A business-focused enterprise architecture provides knowledge about an organization and enables managers to make better-informed decisions.

− Reduced complexity—A well-developed architecture leads to a better integrated solution portfolio, fewer interfaces, increased data sharing, improved reliability of the solutions, and easier maintenance.

− Business-IT alignment—The business focus of the architecture development process and the strong emphasis on the need for the implemented solution to be architecture-compliant together will help ensure that IT solutions are aligned to the needs of the business.

− Many visualization tools support TOGAF are available, such as MEGA and Popkin.

4.3 The Command, Control, Communications, Intelligence, Surveillance, and Reconnaissance (C4ISR)

4.3.1 Introduction

The Command, Control, Communications, Intelligence, Surveillance, and Reconnaissance (C4ISR) Architecture Framework, came from Defence Science Board, who determined in the early 1990s that one of the key means for ensuring interoperable and cost-effective military systems is to establish comprehensive architectural guidance for all of DoD. Consequently, the C4ISR Integration Task Force developed version 1.0 of the C4ISR Architecture Framework in June of 1996, and the C4ISR Architecture Working Group completed version 2.0 in December of 1997 [24]. C4ISR considers three viewpoints, namely, an operational viewpoint, a systems viewpoint, and a technical viewpoint. The three views and their relationships are shown in figure (4) [11].

− The operational view describes the tasks and activities, the operational nodes, and the information flows between nodes that are required to accomplish or support an operation. The operational view describes the nature of information exchanges in detail sufficient to determine what specific degree of information-exchange interoperability is required.

− The systems 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 (or postulated systems that could be used), and facilitates the comparison of current/postulated system implementations with the needed capabilities.

− The technical view articulates the criteria that govern the implementation of required system capabilities [24].
4.3.2 C4ISR discussion

C4ISR has "Y" for most of the comparison criteria; except for Conformance and (CCA) Compliance it gets "P. There is no conformance of a system to the architecture, there are only conformance criteria for the use of C4ISR in an architecture representation [16]. In order to comply with CCA, the architecture description must [8]:

a. Include the appropriate set of products for the intended use.
b. Use the common terms and definitions as specified in the framework.
d. Describe interoperability requirements in a standard way.

C4ISR is intentionally vendor-tool-independent. Vendor products exist to provide the framework products, but no specific vendor is required. Some of the vendor tools that could be used are Ptech FrameWork, netViz and Sterling Software's COOL™ tools.

Consistency across the architecture views is not truly addressed and Relationships among the concepts, and rules of structure, are minimal [16].

5. Conclusions

This paper has review the current state of EAF and the criteria that can be taken into consideration for the election of EAF. An examination of the literature will reveal the following very popular themes:

- There is no ‘one’ best Framework
- No one framework is fully complete. Rather, each has strengths and weaknesses. It is important to match a framework’s strengths with the particular "pain points" in your organization, so that the specific aspects of the framework that you need are well represented in the choice you make. Also, it is not our expectation that any framework will be implemented in its entirety. Rather, think of a framework as a restaurant menu, where you can pick from among the most-appetizing dishes, and in doing so, think about implementing those pieces in a sequence that makes sense.
- Frameworks have strengths and weaknesses
- Frameworks are adaptable
- Tailoring the framework you choose is not only acceptable, it is expected
- Selecting a framework should not consume a great deal of resources (get in and get out).
References

About the authors:

Saber Abdallah is a PhD student at the department of information systems, faculty of computers and informatics, Cairo University. He is conducting research into enabling and supporting debate during architecting activities.

Dr Galal H. Galal-Edeen is an associate professor of information systems at the faculty of computers and informatics, Cairo University and a professor of information systems at the British University in Egypt.