Category: Software & Systems Design

Architecture Methods and Frameworks Overview

Tony C. Shan
Bank of America, USA

Winnie W. Hua
CTS Inc., USA

INTRODUCTION

The e-business models in today's globalized business world demand ever-increasing flexibility, responsiveness, and agility of information technology (IT) solutions. It is compulsory for the IT group to provide higher levels of services at a lower cost for the business to compete and succeed. The reality to IT is that there is no choice other than to build more complex, flexible, scalable, extensible, innovative, and forward-thinking technical solutions, to satisfy the growing business needs.

In large organizations like worldwide financial institutions, virtually thousands, if not millions, of IT applications and systems have been constructed or purchased to provide electronic services for external customers and internal employees in the past years, utilizing heterogeneous technologies and architectures to meet diverse functional requirements from different lines of business. In the banking industry, as an example, the business process generally contains different business sectors in consumer, commercial, small business, wealth management, and capital management. In particular, services are delivered to different channels such as automated teller machines (ATMs), Web browsers, interactive voice response, agent assistance, e-mails, mobile devices, and so on. To effectively manage the architecture assets and rationalize the architecture designs in such a diverse environment, a multi-disciplinary engineering approach is of crucial importance to abstract concerns, divide responsibilities, mitigate risks, encapsulate the complexity, reverse-engineer existing applications, identify reengineering opportunities, and conduct objective technology assessments, which leads to in-depth technical recommendations and rationalization action plans.

BACKGROUND

The computing environment has gone through a number of generations of evolution in the last few decades, ranging from monolithic, client/server, multi-tier, object-oriented, component-based, service-oriented, event-driven, to social computing models. The overall solution architecture has become increasingly complicated and thus hardly manageable through a traditional waterfall process. Previous studies (DoD, 1997; IEAD, 2004; Kruchten, 2003; OMG, 2007; Putman, 2001; The Open Group, 2007; Zachman, 1987) in the past have strived to address the issue of architecture design complexity, which has grown exponentially as the computing space is transformed to a service-oriented architecture paradigm.

The architecture methods and frameworks are the general or proven approaches to designing/developing architecture of information systems. They have progressively undergone an evolutionary growth in the last 20 years. The prominent architecture methods and frameworks developed and proposed so far are listed as follows:

- **Zachman Framework**
- **E2AF:** Extended Enterprise Architecture Framework
- **TOGAF:** The Open Group Architecture Framework
- **RUP:** Rational Unified Process, evolved to Enterprise Unified Process and OpenUP
- **MDA:** Model-Driven Architecture
- **Microsoft Solutions Framework (MSF)**, and **Microsoft Systems Architecture (MSA)**
- **C4ISR:** Command, Control, Computers, Communications (C4), Intelligence, Surveillance, and Reconnaissance (ISR).
- **FEA:** Federal Enterprise Architecture Framework
- **TEAF:** Treasury Enterprise Architecture Framework
- **PERA:** Purdue Enterprise Reference Architecture
- **RM-ODP:** Reference Model for Open Distributed Processing
- **ATAM:** Architecture Tradeoff Analysis Method
- **SAAM:** Software Architecture Analysis Method
- **IDEF:** Integrated Definition Methods
- **MODAF:** Ministry of Defense Architectural Framework

The design principles for the development of architecture methods and frameworks are specified in the following section, with detailed articulations on each method/framework provided in the subsequent section. A comparison matrix is presented afterwards, whereas future trends and recommendations are elaborated next, followed by the conclusions.

DESIGN PHILOSOPHY

The following design principles have been generally applied in developing architecture methods and frameworks:

- Information processing activities comply with applicable laws, orders, and regulations.
- Business objectives are well defined before initiating information technology solutions.
- Total business value is the primary objective when making information technology decisions.
- Architectural selections maximize the interoperability and reusability.
- Architecture methods take advantage of standardization to fulfill common customer requirements and to provide common functions.
- Information technology groups collaborate to share information, data, services, components, and infrastructure required by the business units.
- Business and information technology requirements adopt matured commercial off-the-shelf (COTS) technology where appropriate rather than customized or in-house solutions.
- Information, services, applications, systems, and infrastructure are vital assets that must be managed, controlled, and secured in a holistic manner.
- Enterprise architecture (EA) is consistent with the guidance and strategic goals at the divisional levels.

METHODS AND FRAMEWORKS

The major architecture methods and frameworks are discussed in greater detail in this section.

Zachman Framework

The Zachman Framework (Zachman, 1987) is a logical structure used to categorize and organize the descriptive representations of an enterprise IT environment, which are significant to the organization management and the development of the enterprise's information systems. It takes the form of the two-dimensional matrix, and has achieved a level of penetration in the domain of business and information systems architecture and modeling. It is mainly used as a planning or problem-solving tool. However, it tends to implicitly align with the data-driven approach and process-decomposition methods, and it operates above and across the individual project level.

E2AF

Extended Enterprise Architecture Framework (E2AF) (IEAD, 2004) takes a very similar approach to the Zachman Framework. Its scope contains business, information, system, and infrastructure in a 2-D matrix. E2AF is more technology-oriented. Both Zachman Framework and E2AF approaches are heavyweight methodologies, which necessitate a fairly steep learning curve to get started in an organization.

TOGAF

Another heavyweight approach, The Open Group Architectural Framework (TOGAF) (The Open Group, 2007), is a detailed framework with a set of supporting tools for developing an enterprise architecture to meet the business and information technology needs of an organization. The three core parts of TOGAF are *architecture development method*, *enterprise architecture continuum*, and *TOGAF resource base*. The scope of TOGAF covers *business process architecture*, *applications architecture*, *data architecture*, and *technology architecture*. The focal point of TOGAF is at the enterprise architecture level, rather than the individual application architecture level.

RUP

Rational unified process (RUP) (Kruchten, 2003) attempted to overcome the shortcomings in the heavyweight methods by applying the unified modeling language (UML) in a use-case driven, object-oriented, and component-based approach. The concept of 4+1 views interprets the overall system structure from multiple perspectives. RUP is characterized by process orientation and is generally a waterfall approach. RUP barely addresses the phases of software maintenance and operations, and lacks a broad coverage on physical topology and development/testing tools. It mainly operates at an individual project level. RUP has recently been expanded to enterprise unified process (EUP), and was partially open sourced—OpenUP. RUP is now part of the IBM Rational Method Composer (RMC) product, enabling the process customization.

MDA

Model-driven architecture (MDA) (OMG, 2007) takes a different approach. MDA aims to separate business logic or application logic from the underlying platform technology.