

A Methodology for Implementation of Function Dimension in Zachman Framework

Meron Bekele

KFT (Kifiya Financial Technologies Plc), Addis
Ababa, Ethiopia
merrybekele@gmail.com

Tibebe Beshah

School of Information Science, Addis Ababa
University, Ethiopia
tibebe.beshah@gmail.com

Abstract

This paper aims to propose an implementation methodology for Zachman Framework for Function dimension from software development process perspective. The paper presents the limitation of Zachman Framework with regards to its implementation methodology. Zachman Framework is an enterprise architecture with a logical construct to define and control the interfaces and integration of all components of a system. The Zachman framework enables systematic capture of system specific information from the various dimensions with respect to each view.

The scope of this paper is limited to the “Function” dimension or column. An implementation methodology for Zachman framework has been proposed for each cell that lies within the scope of the study.

A case study was conducted on a company by selecting a specific project. From the selected project, a process has been identified so as to model it based on the proposed methodology. After modeling each cell, an interview has been conducted to validate the proposed methodology. The participants in the interview were stakeholders at each dimension.

The advantage of the proposed methodology is that it is helpful for software development process because it will enable defining the process at each view and also helps as a validating mechanism for the software developers to check whether they have captured each process as per the different stakeholder requirements.

Keywords: Enterprise Architecture; Zachman Framework; Dimension, Implementation Methodology; Function Column

1. Introduction

Many large size enterprises operate in a very dynamic and complex environment and these large enterprises want to ensure that they have the best enterprise architecture framework (EAF) that has the capability of managing the information technology (IT) structure and also the knowledgebase of the environment. Sometimes, the term “Enterprise Architecture (EA)” refers to the group of people responsible for modeling and then documenting the architecture. The term also denotes the process of doing this work [1]. According to the enterprise architecture community, Enterprise Architecture is a framework or “blueprint” for how organizations achieve the current and future business objectives. It examines the key business, information, application, and technology strategies and their impact on business functions. Each of these strategies is a

separate architectural discipline and enterprise architecture is the glue that binds each of these disciplines into an integrated framework [1].

2. Motivation Scenario

The prime motive for this research is the absence of an implementation methodology for architectural frameworks of Zachman enterprise architecture. The methodology will be of great help in software development as the end product is software. Moreover, the need for architectural guideline to make maximum use of the potential of information technology is worth mentioning as an additional motivational factor for this specific research.

3. Proposed Methodology

The aim of the study is to propose a methodology on how to implement Zachman Framework from system development perspective. The proposed

methodology has been developed based on the different literature reviews and analyses made. Thus, the research combines literature review and case study. As the scope of the study is only on process dimension, the proposed methodology is focused on each cell that corresponds to the process dimension. The cells that have been proposed with a methodology are Scope-Process, Business-Process, System Model-Process, and System-Process. The proposed methodology incorporates the modeling technique to be used, the modeling tool which helps model based on the technique selected, and the steps that have to be followed in going from one cell to the other. One of the rules in Zachman Framework discusses about the level of detail when we go down from one cell to the next cell below it.

When we go from one cell to another cell, a transformation is taking place through the application of a different set of constraints. Therefore, what is

making a lower row cell different from a higher row cell in the same column is not the level of detail. The cells in different rows of the same column are different because they are actually models of different things [2]. In any of the cell, we could have a high level of detail model or a medium level of detail model. Level of detail does not necessarily increase from cell to cell down a column. It is fortuitous that only the process models are hierarchical because hierarchical decompositions result in unidentifiable redundancies in the detail which makes normalization difficult, if not impossible [2]. This rule has a direct relationship with our study because it is stating that level of detail is hierarchical in process dimension. Based on this rule, in order to model any cell except the first one, we have to get an input from the cell above it and it has to be noted that any change made at a cell has an impact for the cells below it.

Table 1: Proposed Modeling techniques

<i>Views-Perspective</i>	<i>Focus</i>	<i>Modeling Technique</i>
<i>Scope-Process (Planner view)</i>	Business Function	List, Use case
<i>Business-Process (Owner view)</i>	Process Model	Business Process Modeling Notation (BPMN), Activity Diagram
<i>System Model-Process (Designer view)</i>	Application Architecture	Component Diagram , Context diagram
<i>System-Process (Builder view)</i>	System Design	Class Diagram, Subsystem

As show in Table 1, there are two modeling techniques that have been proposed for each of the cells and also a modeling tool called Altova UMode. This modeling tool is the starting point for successful software development. It visual design application models in UML and it generates Java, C#, or Visual Basic .NET code and project documentation. Or, reverse engineer existing programs into UML 2 diagrams, then fine tune the designs and complete the round trip by regenerating code. Altova UModel is the UML tool that makes visual software design practical for any project. It is the simple, cost-effective way to draw on UML [6].

4. Related Work

A method for consistent modeling of Zachman Framework cell was proposed in [3]. The authors presented three advantages of Zachman Framework (ZF), one is the fact that it uses a well defined perspective, the second one is using of

comprehensive abstract, and the third is normality and extensive usage in practice. Among the challenges sated was the absence of an integrated language to model cells in the framework.

ZF doesn't recommend any tool or model for a particular cell. There is no technique to address the cells in the Zachman Framework matrix which itself resembles an obstacle. To resolve this obstacle, the authors have proposed a Model Driven Architecture (MDA) in order to model all cells in ZF. According to the paper, applying UML seems to be the best solution up to now but the argument that was presented in the study was that UML is not mature enough to support all aspects of enterprise architecture. According to the study, MDA separates certain key of a system into Computational Independent Model (CIM), Platform Independent Model (PIM), and platform specific model (PSM). Figure 1 [3] shows the proposed solution for all cells.

Zachman Framework	Data	Function	Network	People	Time	Motivation
Planner	Out of problem space					
Owner	Business profile	Business Use Case	Organization Unit stereotype	Business profile/use case	Timing diagram	BSBR BMM
Designer	CWM	BPDM	EDOC	BPDM	Timing & State diagrams	PRR
Builder	CWM	UML 2.x	Deployment diagram	Interaction Overview diag.	Timing diagram	OCL
Contractor	CWM	CORBA, EJB, .Net, WSM	CORBA, EJB, .Net, WSM	Interaction Overview diag.	Timing diagram	OCL

Figure 1: Consistence model for Zachman Framework cells

The proposed approach has been implemented on a case study and it was evaluated by a measurement criterion that was set by the authors. The result presented states that the feedback they get were promising compared to previously proposed methods, indicating that it is well suited to diminish the modeling problems one is facing using ZF. Thompson [4] conducted a research on scaling Zachman Framework for small scale non-Enterprise applications. The software development methodology brought forth in Zachman’s System Architecture Framework can be used to design a wide range of information systems. While the Zachman Framework is very robust and typically used for developing large scale Enterprise Applications, the study demonstrates that the framework can easily be scaled to fit a small scale non-Enterprise Application.

Fazil *et al.* [5] discussed about how digital library brought a high impact phenomenon as it contributes a lot in terms of new ways in dealing with knowledge. They stated that libraries built after 1995 have a different look and feel that defines “intellectual information centers” where people and technology interact. According to the paper, most scholars believed that the way knowledge has been created and disseminated fundamentally were altered by a series of innovation begins by Michael S. Hart, with his Gutternberg’s invention as the first producer of free electronic books in 1971. It was presented that apart from being the oldest digital library developed, Gutternberg also works as a catalyst in rapid development of multiple kinds of digital libraries nowadays. The study proposed a framework to determine the content of a semantic theses digital

library based on Zachman Framework data dimension. The study used the six artifacts in Zachman Framework which are Data, Process, Place, People, Time, and Motivation and applied it to represent six layers of semantic digital library architecture. Twenty-four online theses collections of public and private institutions of higher learning in Malaysia were selected for data collection and further evaluation.

A case study approach and a collection of master’s and doctoral dissertations in Perpustakaan Tun Seri Lanang, Universiti Kebangsaan, Malaysia was chosen as the case sample. The collection scope is limited to theses from the Faculty of Technology and Information Science from year 2000 to 2009. A variety of data collection techniques were applied including a pilot study by distributing questionnaire among 20 postgraduate students of University Kebangsaan, Malaysia, interviews with librarians, focus group interviews, and on site observations to observe the specific environment of the theses digital library. As presented in the paper, feasibility study to identify the approaches, methods, and techniques in implementing semantic digital library as well as designing the models and architectures of semantic digital library were represented in a usable semantic theses digital library prototype. Then user testing, evaluation, and refinement of digital library prototype were executed until a complete literature review of semantic theses digital library projects compiled and a formal framework to determine content of a semantic theses digital library based on Zachman Framework accomplished.

5. Conclusion

This paper proposed a methodology on how to implement Zachman Framework from Function column dimension. It is reported on most of the researches that Zachman Framework is a well known and accepted enterprise framework but it lacks implementation methodology. The prime motivation for this research was the lack of implementation despite its acceptance in the industry. In order to come up with a methodology, literature review has been conducted. Based on the different literature reviews, we have proposed the methodology and it was validated by a case study. Based on the case study, Use case was selected for Planner-Process cell, BPMN was chosen for Owner-Process cell, Component diagram has been selected by majority for the Designer-Process cell, and finally Class diagram has been well accepted by programmers as a modeling technique for the builder-Process cell. The proposed methodology is valuable, especially for software developers because at every cell, each stakeholder is presented with a modeling technique that s/he is able to understand easily. On the other hand, it will help the process to be captured well and finally when developers move to development, they will have a requirement that is very clear and also that has been validated by each stakeholder. The research has advantage in puling enterprise architecture concept to the software development practice by addressing each stakeholder with the respective cell. It also adds a value by testing

Zachman Framework for software development process. One of the challenges in software development is capturing process in the manner that stakeholders planned it. Hence, the proposed methodology will help in capturing all the process at each step.

References

- [1] Castelo Branco, "A Method to Define an Enterprise Architecture using the Zachman Framework", in ACM Symposium on Applied Computing, Nicosia, Cyprus, March, 2004.
- [2] John A. Zachman, "The Zachman Framework for Enterprise Architecture: Primer for Enterprise Engineering and Manufacturing", Zachman International, 2003.
- [3] S. Shervin Ostadzadeh, Fereidoon Shams Aliee, and S. Arash Ostadzadeh, "A Method for Consistent Modeling of Zachman Framework Cells", 2007.
- [4] Carla L. Thompson, "Scaling the Zachman Framework", Regis University, School for Professional Studies, 2006.
- [5] Qurratu AiniAdibah Ahmad Fazil, Zuraidah Abdullah, Shahrul Azman Mohd, "Applying Zachman Framework to Determine the Content of Semantic Theses Digital Library", Faculty of Information Science and Technology, Universiti Kebangsaan, Malaysia, 2010.
- [6] Altova, "UML Tool for Software Modeling and Application Development", <http://www.altova.com/umodel.html>, March 2012.