World Applied Sciences Journal 3 (Supple 2): 43-49, 2008
ISSN 1818-4952

A Methodology to Create Data Architecture in Zachman Framework

[1]*Reza Rezaei and* [2]*Fereidoon Shams*

[1]Computer Engineering Department,
Faculty of Engineering,Islamic Azad University of Saveh Branch, Saveh, Iran
[2]Computer Engineering Department, Shahid Beheshti University, Tehran, Iran

Abstract: Lack of methodology for applying Zachman framework is a challenge that an architect who use it will face. Although there have been methodologies which were some how mapped with this framework but until now no methodology has supported one or more columns of this framework completely and in an integrated way. In this paper, we proposed an integrated process for developing Data Architecture views in Zachman framework. One of the most important advantages of this approach is its property which is based on the enterprise's strategic plan, goals and responsibilities, Business and applications Architecture.

Key words: Zachman framework . methodology . data architecture . enterprise architecture framework . modeling.

INTRODUCTION

It goes without saying that nowadays utilizing information and communication technologies in enterprises is one of the most challenging tasks. An enterprise is considered a set of elaborate physical and logical processes in which information flow plays a crucial role. The common way to comprehend procedures in an enterprise is to provide views of components within that enterprise, which is called architecture. Architecture, such as Data Architecture represents only a single view of an enterprise, but Enterprise Architecture refers to a collection of architectures which are assembled to form a comprehensive view of an enterprise. Organizing such great amounts of information requires a framework. Among various proposed frameworks, the Zachman Framework (ZF) is one of the most considerable ways of conceptualization. ZF is widely accepted as the main framework in EA. Compared to other proposed frameworks, it has evident advantages, nevertheless; there are challenges to overcome, among them is the absence of a methodology to specify modeling approach.

The challenge we referred to is also addressed in other researches. Several solutions have been proposed in order to create a methodology, however achieving no success in thoroughly covering all aspects of the framework. The proposed approach in this paper represents a methodology to create Data Architecture in ZF.

The rest of this paper is organized as follows. In Section 2, we introduce Zachman Framework. Next, the problem was defined in section 3. We discuss our proposed approach in section 4 and present a case study in section 5. Finally, in section 6, we make conclusions and suggest some comments for future works.

Zachman framework: In 1987, an IBM researcher, named John A. Zachman, proposed a framework for Information System Architecture, which is now called Zachman Framework (ZF) [1-5]. Zachman borrowed the term architecture from the building trades and discussed the various types of drawings and blueprints a building architect typically developed in order to create a house. He then suggested parallels in software development. He stressed that an organization does not have a single architecture, but has, instead, a whole range of diagrams and documents representing different aspects or viewpoints and different stages. In the years since he wrote his original article, Zachman has worked to refine and elaborate his framework. Figure 1 provides an overview of the current Zachman Framework. ZF is a two dimensional information matrix consisting of 6 rows and 6 columns.

The vertical dimension (the rows) describes the perspectives of those who use the models or descriptions contained in the cells. The top row represents the most generic perspective of an organization, while lower rows are successively more concrete. The bottom row represents a description of

Corresponding Author: Reza Rezaei, Computer Engineering Department, Faculty of Engineering,
Islamic Azad University of Saveh Branch, Saveh, Iran

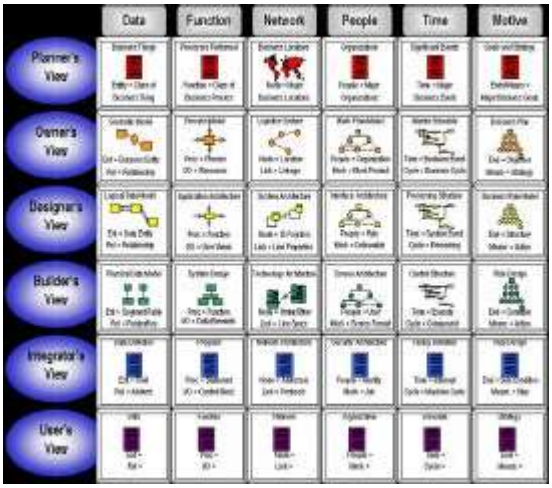

Fig. 1: Zachman framework

the actual data, code and people that make up the enterprise.

The perspectives, starting from the top of Fig. 1, are:

Scope: (Contextual) The Planner's Perspective. This describes the models, architectures and representations that provide the boundaries for the organization and describe what senior executives must consider when they think about the organization and how it interacts with the world.

Business model: (Conceptual) The Owner's Perspective. This describes the models, architectures and descriptions used by the individuals who are the owners of the business process. They focus on the usage characteristics of the products.

System model: (Logical) The Designer's Perspective. This describes the models, architectures and descriptions used by engineers, architects and those who mediate between what is desirable and what is technically possible.

Technology model: (Physical) The Builder's Perspective. This describes the models, architectures and descriptions used by technicians, engineers and contractors who design and create the actual product. The emphasis here is on constraints and what will actually be constructed.

Detailed representations: (Out-of-Context Perspective) A Sub-Contractor's Perspective. This describes the actual elements or parts that are included in, or make up, the final product (e.g. software components). Using the construction metaphor, Zachman refers to it as a sub-contractor's perspective and this makes sense to software developers when the design is implemented with modules or components acquired from others.

The functioning enterprise: The bottom row represents the actual deployed or running elements, data and people of the organization. It isn't a perspective, as such, but the "real world," in all its complexity, that underlies all of the more or less abstract perspectives above it.

The horizontal dimension of the framework (the columns) describes the types of abstractions that define each perspective. These abstractions are based on the widely used questions that people have historically asked when they sought understanding. The six questions or types of abstractions are as follows:

Data: What is it made of? This focuses on the material composition of the product. In the case of software systems, it focuses on data. Zachman has proposed a simple, illustrative model for each of the columns. In this case, the model is: Thing-Relationship-Thing.

Function: How does it work? This focuses on the functions or transformations of the product. The model is: Process-Input/Output-Process

Network: Where are the elements located relative to one another? This focuses on the geometry or connectivity of the product. The model is: Node-Line-Node

People: Who does what work? This focuses on the people and the manuals and the operating instructions or models they use to perform their tasks. The model is: People-Work-People

Time: When do things happen? This focuses on the life cycles, timing and schedules used to control activities. The model is: Event-Cycle-Event

Motivation: Why do things happen? This focuses on goals, plans and rules that prescribe policies and ends that guide the organization. The model is: End-Means-End

The problem space: ZF is widely accepted as the main framework in EA. Compared to other proposed frameworks, it has evident advantages to list: (1) using well-defined perspectives, (2) using comprehensive abstracts, (3) normality and (4) extensive usage in practice, nevertheless; there are challenges to overcome, among them is the absence of a methodology to specify modeling approach.

Although some methodology were suggested for ZF, but none of them succeed to cover all aspects of the framework. We can refer to EAP as the best methodology suggested for ZF, however, it addresses two rows of the framework and can't support the lower ones.

ZF expresses what information must be created for each cell of the framework; however, it doesn't indicate how this information must be created. We are not intended to address this as ZF weak points, since ZF is just a framework, not a methodology. Anyhow, an architect who uses ZF has to overcome this problem.

Due to importance of data in an enterprise, an approach is proposed to show how Data models can be created in ZF. This approach contains steps that are required to create a Data Architecture in the framework. Each step indicates how the required information should be gathered and how they should be used to create appropriate models for each cell in ZF.

We use strategic planning, goals and functions to extract data architecture. Hence, this approach is Data-centric and will align Data architecture to Function architecture. Proposed approach can be used as an integrated process to cover all cells from Planner to Sub-contractor's perspective. Note that we will drop the Function Enterprise row, since it is not a model but a real world. Figure 2 depicts the problem space.

Proposed approach: In this section we define the Data column cells and present our approach to extract Data architecture.

Planner-data cell ("List of Things Important to the Business")

Define: This is simply a list of things (or objects, or assets) that the Enterprise is interested in-the "universe of discourse" relative to things. It is probably adequate that this list is at a fairly high level of aggregation. It defines the scope, or boundaries, of the Rows 2-5 models of things that are significant to the Enterprise [5].

Solution: Data hierarchy can be extracted based on data analysis of enterprise strategic plan, enterprise goals and enterprise missions [2-10] Fig. 2.

Owner-data cell ("Semantic Model")

Define: This is a model of the actual Enterprise things (objects, assets) that are significant to the Enterprise. It typically would be represented as an "E/R"-type model and would be at a level of definition that it would express concepts (terms and facts) used in the significant business objectives/strategies that would later be implemented as "Business Rules" [5].

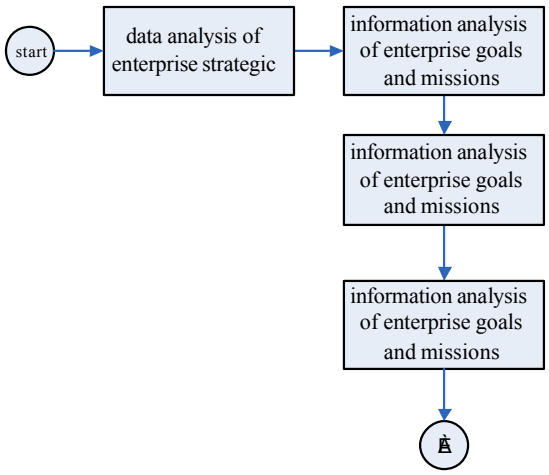

Fig. 2: Planner-data cell extract process

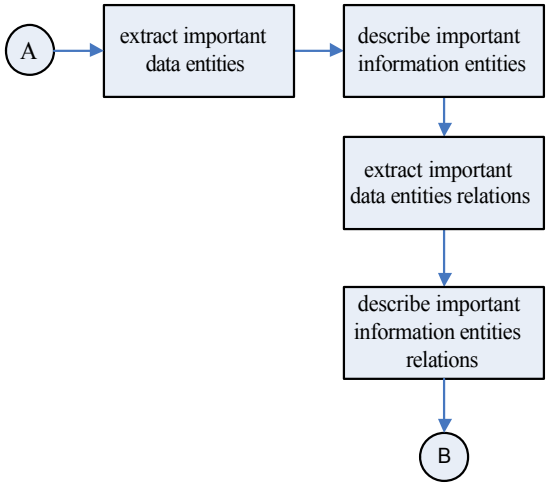

Fig. 3: Owner-data cell extract process

Solution: In this perspective important data entities and their relations extract base on planner's perspective outputs [2-10] Fig. 3.

Designer-data cell ("Logical Data Model")

Define: This is a model of the logical (implementation-technology neutral) representation of the things of the Enterprise about which it records information (in either automated or non-automated form). It would be represented as a fully attributed, keyed, normalized E/R-type model reflecting the intent of the Semantic Model [5].

Solution: first, E/R model extracts based on owner's perspective outputs. Next, this model will be normalized. After normalizing, data entities will be crossed with process entities. Finally, data entities identifications will be extracted [2-10] Fig. 4.

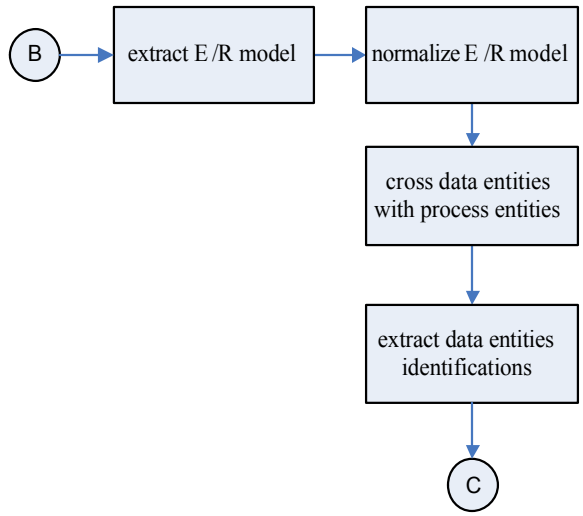

Fig. 4: Designer-data cell extract process

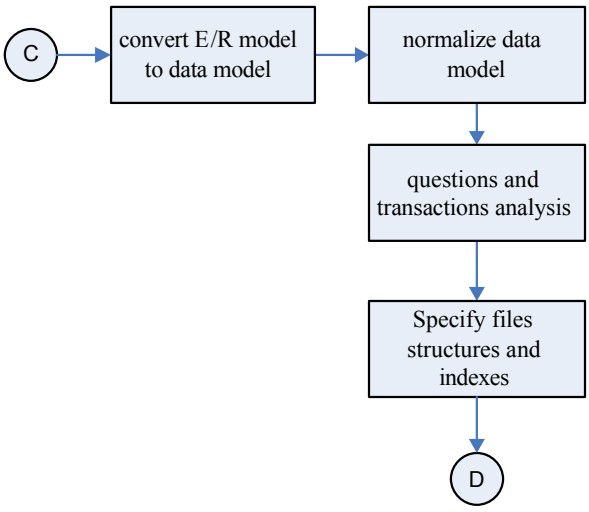

Fig. 5: Builder-data cell extract process

Builder-data cell ("Physical Data Model")
Define: This is a technology constrained, or physical representation of the things of the Enterprise. The representation style of this model would depend on the technology chosen for implementation. If relational technology is chosen, this would be a model of the table structure required to support the Logical Data Model in a relational-style model. In an Object-Oriented notation, this would be the class-hierarchy/association style models [5].

Solution: In this perspective, the E/R model converts to data model. Next, data model will be normalized. After that, questions and transactions will be analyzed and will be improved if require. Finally, files structures and indexes will be specified [2-10] Fig. 5.

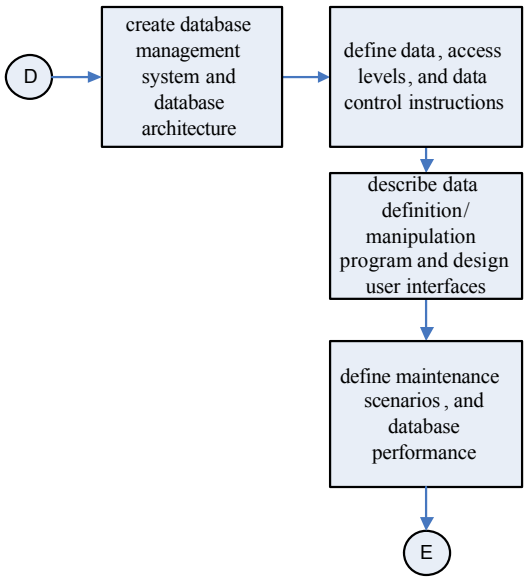

Fig. 6: Subcontractor-data cell extract process

Subcontractor-data cell ("Data Definition")
Define: This would be the definition of all the data objects specified by the Physical Data Model and would include all the data definition language required for implementation[5].

Solution: first, database management system and database architecture will be created. Next, data and their control and access levels instructions will be defined. After that, data definition and data manipulation program will be described. Finally, user interfaces, maintenance scenarios and database performance will be defined [2-10] Fig. 6.
Case study: In previous section, an integrated way for creating various views of data architecture within Zachman framework was presented. Since the suggested solutions in the field of enterprise architecture are less supportable by formal methods, it is common to examine their creditability through case studies. Conducting case studies through suggested solutions will evaluate the efficiency of the proposed method as an actual criterion.

Ports & shipping organization of Iran (Shipping and Marine affairs authority) is the organization that we aimed at study its data architecture based on the suggested method.

It must be noted that as the constructor and the subcontractor views require designing and implementing the physical model and definition of organization data, they were not addressed in this study.

Creating programmer cell: The strategic plan and organization goals are presented in Table 1.

Table 1: Strategic plan and organization goals

Type	Title
Strategy	Programming for most compliance with regulations of adjoined international conventions
Strategy	Continuous establishing and issuing national standards regarding security, search and rescue
Strategy	Improving the coordination between local, provincial and national operational potentials, preventing and dealing with security issues, security and protection of marine environment
Goal	Improving security level in marine transportation and shipping subdivision
Goal	Improving the level of health and protection of marine environment Better compliance with international conventions and membership in them
Goal	Improving the ability of rescuing natural disaster casualties

Table 2: Hierarchy of data issues

Field	Data issue
Marine	Information about vessel transportation in the canal Information about delivering services to the vessel and marine units
	Coastal stations and transitional equipments information
	Hydrography and dredging information
Shipping	Information about marine training and shipping certifications
	Information about vessels registry
Security	Information about canal security
	Information about vessel security control and inspection
	Information about marine search and rescue
	Information about ship and merchandise salvation in the sea
Environmental protection	Information about pollution prevention and avoidance
	Information about dealing with marine disasters
International relationships	Information about membership in international conventions
	Information about membership in international communities

Table 3: Important data issues

Issue	Entity
Information about vessels registry	Information about vessel flag registry
	Information about canceling vessel registry
	Information about repairing and changing vessel user
	Information about manufacturer and vessel repairing companies
	Information about issuing technical and security certification for vessels which are under the flag
	Information about vessel mortgage registry

Organization goals and roles, data issues and their hierarchical structure are extracted by analyzing strategic plan data and information which is shown in Table 2.

In the fourth step, the identified and extracted data issues are explained generally.

Creating the owner cell: In this section, important data entities of organization are extracted based on data issues created in the programmer view. For example, extracted data entities which are important for marine vessels registry data issue are shown in Table 3.

Each of important information entities are extracted after identifying and describing the relationships between important data entities. Then, each of these relationships are described. For example, the relationship between a given entity with other entities is shown in Table 4.

Creating the designer cell: In this view, an integrated relationship-entity model is extracted based on extracted entities and relationships in the owner view and initial normalization is conducted on it. In the third step, the extracted entities are adapted with processes to check and analyze entities validation and to identify

Table 4: Relationships between entities

Entity	Related entities
Information about vessel flag registry	Information about issuing technical and security certifications for vessels which are under the flag
	Control and inspection of the vessel security
	Information about radio services
	Information about issuing the vessel license
	Information about servicing to the vessel
	License for transmission equipments and assigning codes
	Information about the vessel mortgage registry
	Information about repairing and changing user vessel
	Information about canceling long term vessel registry
	Information about manufacturer and vessel repairing companies

Table 5: Adaptation of entities with processes

	Entities			
Functions	Vessel manufacturing request	Inspecting the vessel being constructed	Information about vessel temporary registry	Information about vessel administration
Recording vessel registry request	CRU			
Vessel temporary registry	R	R	CRU	
Canceling vessel temporary registry			U	
Change in vessel administration				RU

Table 6: Results of questionnaires

Products answer	Complete (%)	Acceptable (%)	Incomplete (%)	Absolutely no (%)	White (%)
Data issues	7	83	7	3	0
Data entities	12	78	7	1	2
Data entities relationships	13	77	5	2	3
Entity-relationship model	10	78	9	11	2
Adapting entities with processes	11	74	7	3	5
Data entities identification card	10	73	8	2	7

required application systems. Then, the data entities identification card is extracted. For example, a part of entities adaptation with the processes is presented in Table 5.

In order to evaluate the accuracy and correctness of the presented method for creating integrated data architecture and examining outputs and manufactured products, some questionnaires were provided for the organization architects and managers.

The results of these questionnaires that are shown in Fig. 6, will prove the accuracy and correctness of the method presented for creating all data architecture views.

CONCLUSION

Generally, the benefits of the suggested method for creating data architecture in Zachman framework can be explained as follows:

- The suggested method for creating data architecture based on Zachman framework has been designed and implemented as a known framework.
- The suggested method will present in detail the the process of creating all data architecture views.
- One of the other suggested method is that organizations architects can create their own data architecture using this method.
- In the third step, the suggested method for creating data architecture considers the process of creating data architecture designer view, data architecture equivalence and the function through adapting entities with processes.
- The suggested method for creating data architecture considers the equivalence of data architecture with the organization strategic plan by data analysis of the organization strategic plan and information analysis of the goals in the programmer level.

- The suggested method make the data architecture inclusive through creating all data architectures in Zachman framework.
- Examining data architecture in each step of its creating process is possible through using the suggested method and following the steps defined for each data architecture views.
- Since in Zachman framework the criterions are well defined and all required activities are exactly defined, lower time and cost is required to create various data architecture views and this can be considered as one of the other benefits of this method.

REFERENCES

1. Spewak Steven, H., 1993. Enterprise Architecture Planning: Developing a Blueprint for Data, Applications and Technology, John Wiley and Sons.
2. Zachman, J.A., 2003. The Zachman Framework: A Primer for Enterprise Engineering and Manufacturing.
3. Zachman, J.A., 1987. A Framework for Information Systems Architecture. IBM Systems Journal, Vol; 26 (3).
4. Sowa, John F. and A. Zachman, John, 1992. Extending and Formalizing the Framework for Information Systems Architecture. IBM Systems Journal, 31 (3): 590-616.
5. Zachman, J.A., 2003. The Framework for Enterprise Architecture. Cell Definitions. ZIFA.
6. Popkin's System Architect, 2004. Building Enterprise Architecture: The Popkin Process, Popkin Company.
7. Villiers, D.J.De, 2001. Using the Zachman Framework to Assess the Rational Unified Process. IBM Rational Edge.
8. Dennis, M. Ahern, Clourse Aaron and Turner Richard, 2003. CMMI Distilled, A practical guide to integrated process improvement, Second Edn., Addison Wesley, Part1, Chapter 1.
9. Blueprint Technologies, 2004. Best practice approach to enterprise architecture. 8618 Westwood Center Suite. 310, Vienna, VA 22182, United States. 2. NASA-ESDIS, Bldg 32, Rm S224C Mail Code 423.
10. Martin, James, 1990. Information Engineering, Prentice Hall.