Zachman Enterprise Framework Insights

1. INTRODUCTION

The Zachman Framework was developed in 1987 while businesses were beginning to automate their record-keeping. The Zachman framework is an enterprise level classification of the knowledge required to change the enterprise successfully.

Rather than the one-dimensional analysis of other enterprise architectures (TOGAF), the Zachman framework is a two-dimensional classification schema that shows the interaction between the objects (columns) involved in the business: What, How, When, Who, Where and Why (the Kipling questions), and the level of detailed knowledge required as the investigation descends through the rows (abstraction levels) of: Scope, Enterprise, System, Technology, and Components.

The Zachman Framework is not a methodology in that it does not imply any specific method or process for collecting, managing, or using the information that it describes rather, it is a two dimensional classification that creates a framework for organizing architectural artifacts (models, design documents, specifications, etc.) that takes into account both the level the artifact targets (for example, scope or technology) and what particular issue (for example, what or how) is being addressed.

The figure below is a screenshot of the opening screen for the CASE tool Visible Analyst and it shows the Zachman Framework as applied to modelling data.

Figure 1. Zachman Enterprise Framework

Updating the deliverable document names to reflect current software development practice gives a new version of the Zachman Enterprise Framework as shown in Figure 2.

Abstraction Level	What data	How function	Where network	Who people	When time	Why motivation
scope	List of Business Data	List of Business Processes	List of Business Locations	List of Business Units	List of Business Cycles	List of Business Objectives
enterprise model	Conceptual (Kimball Star) Data Model	Business Process Model	Enterprise Architecture	Work Flow Model	Master Schedule	Business Plan
system model	Logical (Semantic) Data Model	Application Architecture	Distributed Systems Architecture	Human Interface Architecture	Processing Structure	Business Rule Model
technology model	Database Specification	Functional Specification	Technology Architecture	Presentation Architecture	Control Structure	Rule Design
detailed representation	Data Definition (DDL) files	Source Code Program files	Network Components	Security Architecture	Timing Definition	Rule Specification

Figure 2. Zachman Enterprise Framework with up-to-date titles for the deliverable documents

2. FRAMEWORK RULES

When using the framework for an enterprise architecture, there are rules to ensure that the concept of the architecture is not compromised.

1. *The columns have no order*: The columns are interchangeable but existing columns cannot be renamed, removed or new columns added
2. *Each column has a simple, unique generic model*: Each column can have its own meta-model. Each column may have its own methodology, for example, the How column has the Software Development Life Cycle (SDLC) methodology. By analogy, the What column has the Database Development Life Cycle (DDLC).
3. *Each row describes a distinct, unique perspective*: The rows are not interchangeable and existing rows cannot be renamed, removed or new rows added. Each row describes the view of a particular business group and is unique to it. All rows are usually present in most hierarchical organizations.
4. *Each cell is unique*: The combination of 2,3 & 4 must produce unique cells where each cell represents a particular case. For the same reason as for not adding rows and columns, changing the cell names may change the fundamental logical structure of the Framework. No concept can be classified into more than one cell and each cell model specializes its column's generic model.
5. *The composite or integration of all cell models in one row/column constitutes a complete model from the perspective of that row/column.* The cell models are only vertical or horizontal relationships between cells. Do not create diagonal relationships between cells.

The main objection to the framework is its scale which is the entire business and the large number of individual tasks required to document the business. The framework does not provide any insights or methodologies to help.

3. FRAMEWORK INSIGHTS

The Zachman Framework can be organized to give new insights into the process of documenting the enterprise as shown in Figure 3.

	Business Operations			Business Organization		
Abstraction Level	**What data**	**How function**	**Where network**	**Who people**	**When time**	**Why motivation**
scope	List of Business Data	List of Business Processes	List of Business Locations	List of Business Units	List of Business Cycles	List of Business Objectives
	D	S				
enterprise model	Conceptual (Kimball Star) Data Model	Business Process Model	Logistics Architecture	Work Flow Model	Master Schedule	Business Plan
	D	D				
system model	Logical (Semantic) Data Model	Application Architecture	Distributed Systems Architecture	Human Interface Architecture	Processing Structure	Business Rule Model
	L	L				
technology model	Database Specification	Functional Specification	Technology Architecture	Presentation Architecture	Control Structure	Rule Design
	C	C				
detailed representation	Data Definition (DDL) files	Source Code Program files	Network Components	Security Architecture	Timing Definition	Rule Specification

Figure 3. Insights into the Zachman Enterprise Framework

3.1 Location of the Operating Enterprise

The first insight provided by the framework is that for any working business, it is already at the bottom row of the framework (see Figure 2). All the necessary documentation is available to the workers, whether that documentation is catalogued, well organized and widely available is a different question.

3.2 Separation of Enterprise Operations and Organization

The second insight provided by the framework is the organization of the columns into two groups – operations and organization. The columns What, How and Where focus on the processes of the business. The columns Who, When and Why focus on the management and strategy of the business.

3.3 Integrating the Enterprise Architecture

The third insight provided by the framework is the context of other architectures. The Zachman framework provides a context for the TOGAF architecture (yellow cells in Figure 3) and shows where the scope of the architecture originates and what needs to be described in detail. The TOGAF Architecture addresses only 4 of the 6 Kipling questions (columns) and is only concerned with a single level of abstraction (row). This implies less scope and less in-depth analysis in the documents.

The TOGAF architecture implies a vertical stack dependency which is not necessarily true. Figure 3 shows that the enterprise architecture can be horizontal as well as vertical as assumed by TOGAF. This observation calls into question the assumed TOGAF architectural organization.

3.4 Software Development Methodology

The fourth insight provided by the framework is the identification of the development methodologies used to automate the operational processes. The How column is the Software Development Life Cycle (SDLC) methodology to automate the functions of the business.

3.5 Database Development Methodology

The fifth insight provided by the framework is the identification of the development methodologies used to automate the operational processes. The What column is the Database Development Life Cycle (DDLC) methodology to automate the record keeping for the business.

4. BUSINESS OPERATIONS

When planning changes to the enterprise, it is not necessary to examine the entire architecture. Looking only at the operational part of the Zachman Framework, we see the tasks and methodology mapping required to affect operational changes as shown in Figure 4.

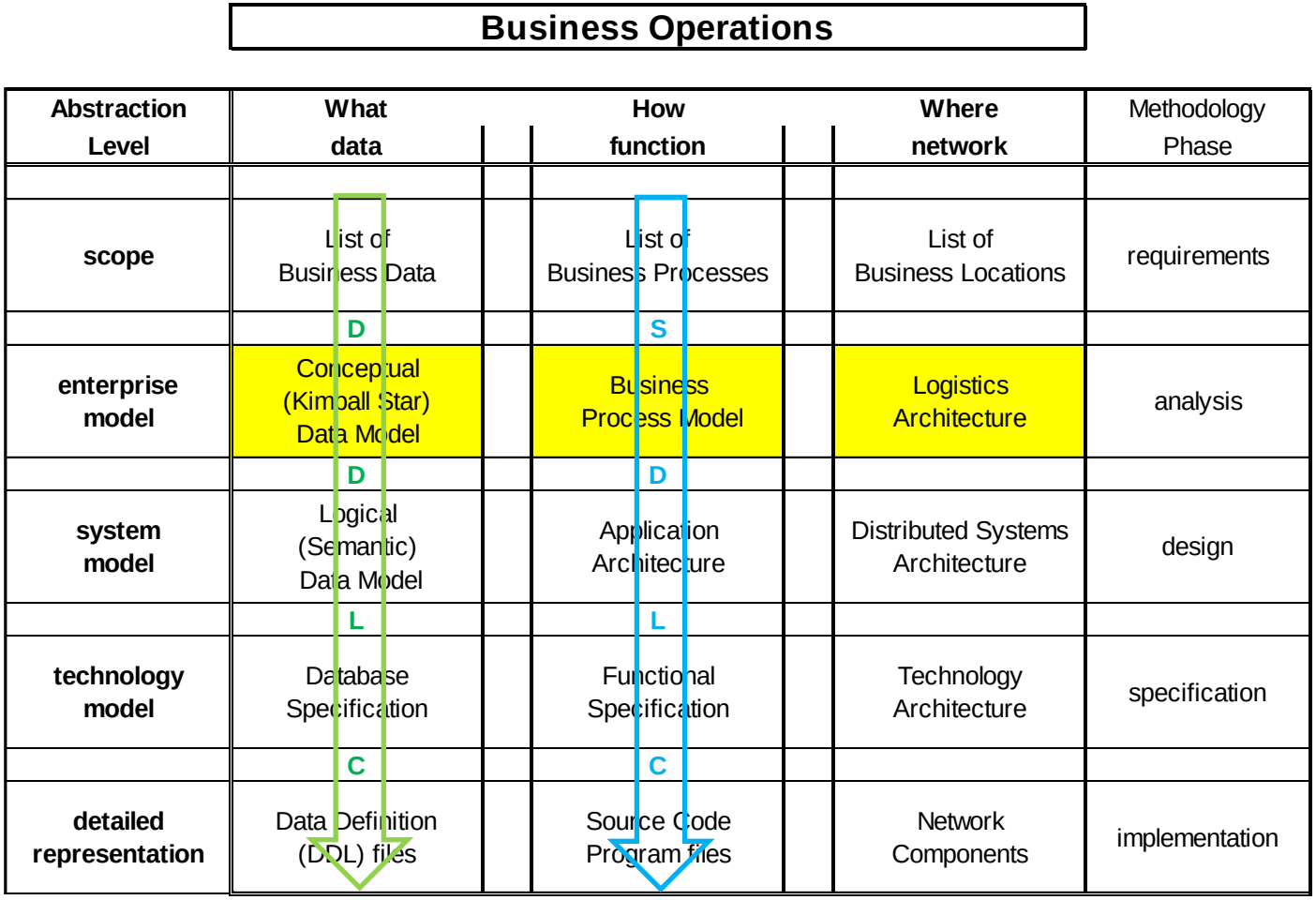

Figure 4. Zachman Business Operations

Looking at the operational portion of the Zachman Framework (figure 4), currently there is no identified methodology for the column Where. This column represents the level of the computer/network architecture. The Where column is dependent on the available computer/network facilities which are always available depending on budget.

5. APPLICATION DEVELOPMENT FRAMEWORK

Converting the framework in figure 4 into a framework for moving business operations onto computers means taking the columns What and How, and documenting how their information interacts. The interaction between the SDLC/DDLC is much more complex than an empty column indicates (figure 4).

Describing the interactions creates a much more complex architectural solution at the interface between the What and the How columns. Making these changes produces the Application Development Framework (ADF) shown in Figure 5.

Abstraction Level	What data	SDLC/DDLC interaction	How function	Methodology Phase
scope	List of Business Data	List of Data Strategies	List of Business Processes	requirements
enterprise model	Conceptual (Kimball Star) Data Model	Enterprise Bus Matrix	Business Process Model	analysis
system model	Logical (Semantic) Data Model	Business Object to Logical Tables Mapping	Application Architecture	design
technology model	Database Specification	Database Interface Specification	Functional Specification	specification
detailed representation	Data Definition (DDL) files	Interface Stored Procedure Files	Source Code Program files	implementation

Figure 5. Application Development Framework

6. REFERENCES

Zachman, J.A. "A framework for information systems architecture", IBM Systems Journal, Vol 26, No 3, 1987

Sowa, J.F. and Zachman, J.A. "Extending and formalizing the framework of information systems architecture", IBM Systems Journal, Vol 31, No 3, 1992

Zachman, John https://www.zachman.com/about-the-zachman-framework

Wikipedia https://en.wikipedia.org/wiki/Zachman_Framework

Dragon1 https://www.dragon1.com/frameworks/zachman-framework-for-enterprise-architecture

Kipling, Rudyard, *I Keep Six Honest Serving Men*

TOGAF Architecture, The Open Group Website

Software Development Life Cycle (SDLC), 6 Basic SDLC Methodologies | Robert Half

Database Development Life Cycle (DDLC), The database development life cycle | Open University UK | Edupedia Online

