Zachman Framework in Teaching Information Systems

Mladen Varga
University of Zagreb, Graduate School of Business & Economy
mladen.varga@efzg.hr

Abstract. *The Zachman Framework for information systems architecture is a classification scheme for development and documenting an enterprise-wide information system. It is useful in teaching various topics on information systems, such as: information systems development as an engineering process, basic information systems components and design artefacts, system development life cycle and information systems architecture, roles involved in development of information systems, control points of information systems development, focus of information systems development methods, and improving information systems maintenance. The Framework also helps improving information systems teaching programme.*

Keywords. Zachman framework, information systems architecture, information systems, enterprise architecture, teaching information systems

1. Introduction

Abstract concepts and high complexity of information systems (IS) are reasons that many students consider information systems topics as difficult and boring. Therefore, teaching information systems is a challenging task that permanently requires finding new ways of teaching.

The paper discusses the Zachman framework for IS architecture and its usability in teaching IS. In Section 2 and 3 IS and its role in an enterprise are described; and the Zachman Framework for information systems architecture is introduced. In Section 4 some possibilities of using Framework in teaching IS at the undergraduate university level are discussed.

2. Zachman Framework for information systems architecture

Information system is a system, whether automated or manual, that comprises the entire infrastructure, organization, people, machines, and/or methods organized to collect, process, transmit, and disseminate data that represent user information. Information system is a complex system that covers all informational tasks of the enterprise. Moreover, information system is an information-based model of the enterprise, which describes the enterprise's business through data.

The Zachman framework for IS architecture [2, 3, 5] is used for development and/or documenting an enterprise-wide IS architecture. The purpose of the framework is to provide a basic structure that supports the organization, access, integration, interpretation, development, management, changing of a set of architectural representations, called *artefacts*, of the enterprise's IS. The Framework enables focused concentration on selected aspects of the IS without losing sense of the contextual or holistic perspective.

The Zachman Framework is actually a generic classification scheme for design artefacts of any *complex product*, such as building, airplane, information system or enterprise. A complex object considered in the Framework functions as a stand-alone and self-contained unit. There is no difference whether the product is physical (building) or conceptual (enterprise or IS). Although often looked at as a framework for IS, the Zachman Framework is successfully extended to the Framework for Enterprise Architecture. In general, a self-contained enterprise would be defined by a set of business functions and assets; all integrated in support of a common mission or set of objectives [4].

Zachman derived the Framework from analogous structures in the traditional engineering disciplines such as architecture, which classify and organize the *design artefacts* created in the process of designing and producing complex products (e.g. buildings). The engineering disciplines have accumulated considerable knowledge of their product development and management. This knowledge has enabled great increases in product sophistication and product change management over time.

The first view on design artefacts is through *product characteristics*, which include the *what*, *how*, *where*, *who*, *when* and *why* characteristics. They are explained as *what* it is made of (material), *how* it works (process), *where* the components are (location), *who* does what work (people, organization), *when* things happen (time, event) and *why* various choices are made (motivation). Fig. 1 shows some artefacts for product characteristics in house construction and in information systems development.

The other view on design artefacts is through *perspectives* or *roles* in the product development process. Perspectives or roles include the *contextual* perspective (*planner* role), the *conceptual* perspective (*owner* role), the *logical* perspective (*designer* role), the *physical* perspective (*builder* role), and the *component* perspective (*sub-contractor* role). The Fig. 2 shows these perspectives; and models produced in each perspective in business and development terms.

Product characteristics	Question	Artefacts in house construction	Artefacts in information systems development
Material	What	House, room	Data entity
Process	How	Eat, play, sleep	Computer program, manual procedure
Location	Where	Placement of rooms	Network of locations
People	Who	Occupants, guests, pets	User, organization
Time	When	When to eat, play, sleep	Event
Motivation	Why	Accommodate growing family	Business goal, business rule

Figure 1. Product characteristics of the Zachman Framework

Perspective (role)	Enterprise model (development model)	Description
Contextual (planner)	Scope (contextual model)	Definition of the product's direction and purpose
Conceptual (owner)	Business model (conceptual model)	Definition (in business terms) of the product
Logical (designer)	System model (logical model)	Definition (in designer's term) of the product
Physical (builder)	Technology model (physical model)	Definition (in technology term) of the product
Component (sub-contractor)	Component model (physical component model)	Specification of the product's components

Figure 2. Perspectives (roles) of the Zachman Framework

The Zachman framework for IS architecture, graphically depicted in Fig. 3, shows the artefacts that constitute the intersection between the perspectives or roles in the design process, shown in rows, and the product characteristics, shown in columns. In the case of some complex product, such as an enterprise, some cells of the Framework matrix are more hypothetical and more empirical than the others. However, all cells exist at least hypothetically.

The Framework is easy to understand. It is comprehensive while it addresses the product (e.g. enterprise or IS) in its entirety. It helps to think of complex concepts in non-technical words. It enables to work with abstractions to isolate simple elements without losing sense of the complexity of a product as a whole. Finally, it is independent of methods and tools. It is intended to be a thinking tool in dealing with complexities and dynamics of a complex object.

3. Abstractions in the Zachman Framework of IS architecture

Let us examine the abstractions in the Framework for IS architecture. Upon completion of the architecture and the development processes, the organization will have artefacts for each framework cell within the scope of the new system, that is, a complete set of explicitly stated models, vertically and horizontally integrated, at a very high level of detail.

	What **Data (entities)**	*How* **Function (activities)**	*Where* **Network (locations)**	*Who* **People**	*When* **Time**	*Why* **Motivation**
Scope (contextual perspective) *Planner*	Class of business things *(list of things important to the business)*	Class of business processes *(list of processes the business performs)*	Major business locations *(list of locations in which the business operates)*	Major organization units *(list of orgs important to the business)*	Major business events *(list of events significant to the business)*	Major business goals *(list of business goals & strategies)*
Enterprise model (conceptual perspective) *Owner/analyst*	Semantic data model, conceptual data model *(E/R diagram)*	Business process model *(functional hierarchy, use case)*	Business logistics system	Workflow model *(organization chart)*	Master schedule *(state transition, PERT chart)*	Business plan
System model (logical perspective) *Designer*	Logical data model *(normalized data model)*	Application architecture *(use case)*	Distributed systems architecture *(component diagram)*	Human interface architecture *(use case)*	Processing structure *(data flow, entity life, Petri net)*	Business rules
Technology model (physical perspective) *Builder*	Physical data model *(physical data model)*	Systems design *(struct. diagr., action diagr., object method, pseudo code)*	Technology architecture *(deployment diagram)*	Presentation architecture *(graphic & menu screens)*	Control structure *(sequence diagram, state diagram)*	Rule design
Component model (component perspective) *Sub-contractor*	Data definition *(database schema)*	Programs *(program code, manual procedure)*	Network architecture *(node specification)*	Security architecture *(accessing specification)*	Timing definition *(interrupt specification)*	Rule specification
Functioning Enterprise (functioning perspective) *User*	Data	Function	Network	Organization	Schedule	Strategy

Figure 3. The Zachman Framework for IS architecture

Each row in the Fig. 3 represents the view on IS from specific *perspective* that belongs to unique *role*. The role of a *planner* in the contextual perspective is to set the scope (contextual model) or the strategy of an IS. The role of a business *owner*, business people or a system analyst in the conceptual perspective is to define the business in a formal way (enterprise model, business model, conceptual model). The role of a *designer* in the logical perspective is to design the IS model (logical model). The role of a *builder* in the physical perspective is to design the technology model (physical model). A *sub-contractor* is responsible for components of the system (component model). Finally, the IS is built in concordance with all the mentioned models.

The columns in the Framework represent different IS abstractions. Each row in the *data* column addresses understanding of and dealing with enterprise's data (what are things of interest). The rows in the *function* column describe the various aspects of operations of the IS (how are things processed). The *network* column is concerned with locations (where the operations are done). The *people* column describes who is involved in the IS. The *time* column describes the effects of time on the IS. The artefacts of this column are difficult to address in isolation from others, especially from function artefacts. Usually, it describes when the function is executed. The *motivation* column is concerned with the conversion of business goals and strategies into specific business rules.

4. Usability of Zachman Framework in Teaching Information systems

The Zachman Framework is a useful tool that may help answer many IS issues. A list of them, although not exhaustive, is discussed bellow.

4.1 IS development as an engineering process

An engineering process uses standard methods, techniques and tools to develop and produce a product in a proper time. During the process, various workers participate in different and specific roles or perspectives. Just like the traditional engineering discipline identify the descriptive abstractions that are used for physical products (see Fig. 1, house), it is possible in a similar way to identify the equivalent process and descriptive abstractions for non-physical products, such as enterprise or IS. The various abstractions on the IS, i.e. models of the IS, enable increases in IS sophistication and ability to manage IS change over time in the same manner as in traditional engineering processes (e.g. architecture or manufacturing). Therefore, the answer to the question whether IS development is an engineering discipline is definitely positive.

Nevertheless, instead of representing the development process as a set of steps, the Framework is organized around different perspectives taken by the various workers (planner, owner, designer, builder, and sub-contractor). All engineering projects have to pass by all perspective explicitly or implicitly.

4.2 Basic IS components and design artefacts

The Framework is useful in recognizing the basic IS components, that are specified by set of IS artefacts. The first components are *data* and *functions* (processes). In functioning IS, data are settled in the database; functions are computer programs or manual procedures. *Locations* (network) are important in distributed environment. *Time* deals with the schedule of business events. Who is involved in the execution of IS functions is described by the organization of *people* component. *Motivation* is a soft component pushing the development of an IS. Let us look at artefacts, which represent particular components.

Motivation is the answer to the question why. It deals with business goals and strategies. In the contextual perspective, the enterprise identifies its goals and strategies in business language. In the conceptual perspective, they are translated into specific business rules and constraints that apply to an enterprise's operation (e.g. business plan). In the logical perspective, business rules are expressed in terms of information that is and is not permitted to exist. In physical perspective, business rules are converted into program design elements (e.g. conditions). In the next perspective, they may become specific programs, and in last row of the Framework matrix, the business rules are enforced.

The *function* component describes the translation of the mission of an enterprise into detailed definitions of its operations. A list of groups of processes an enterprise conducts is described in the contextual perspective. The conceptual perspective describes these processes by a business process model. The logical perspective usually defines the ways of data transforming processes, e.g. the conversion of input data into output data. The physical perspective deals with the definition of computer programs and their interactions, for example in the form of pseudo-code. The next perspective converts these into program code.

The *data* component addresses the problem of enterprise's data. In the contextual perspective, there is a list of things important to operate an enterprise. The conceptual perspective defines the business model of things as seen by the participants in the business. The logical perspective defines an information-based model of the business. The physical model defines the physical elements of data used in a database management system. In the next perspective, the data tables or the appropriate elements of the database are implemented through database schema.

The *network* component is concerned with the distribution of an enterprise's business. A list of places where the business occurs is described in the contextual perspective. In the conceptual perspective, the business logistics model describes the ways various locations interact with each other. The logical perspective produces the logical model of data and process distribution. In the physical perspective, the distribution is translated into computer facilities required in each location. In the next perspective, the requirements of the facilities are described by various specifications, such as protocols.

The *people* component specifies who is involved in the business. A list of organisational units involved in the business and their missions is described in the contextual perspective. The conceptual perspective defines the business model of the workers or the units, linked to the functions they execute. The logical perspective defines the architecture of the interface between

each worker and the technology. The physical perspective defines the presentation architecture. In the next perspective, the physical design is converted into various specifications in programs, such as access permissions.

The *time* component defines the time effects of the business. A list of business events or business cycles is described in the contextual perspective. The conceptual perspective describes the event model of the business. It defines when functions are to happen and under what circumstances. The logical perspective defines the precise model of events, which causes specific data transformation. In the physical perspective, the events become triggers and messages. In the next perspective, they are incorporated into programs.

One of the most important contributions of the Framework is the recognition that data and functions are not the only components; and that is important to take into consideration motivation issues, to recognize people and organizational issues, to deal with multiple locations, and to take into consideration timing issues such as events or schedules.

4.3 System development life cycle and information systems architecture

System development life cycle (SDLC) is an appropriate organization of the steps required to develop the system. The main steps are called phases or stages. Although the Framework does not consider the development steps, each perspective corresponds to one particular development phase. The contextual perspective corresponds to the planning phase, the conceptual perspective to the analysis phase, the logical perspective to the logical design phase, the physical perspective to the physical design phase. It is important to stress that after finishing a phase and stepping into another phase the perspective is changed, and possibly done by another worker or role.

4.4 Control points of IS development

The control points or milestones in the development process, as in all engineering product development processes, are points of changing the perspective. The point of changing the contextual perspective into the conceptual perspective or finishing the planning phase, the point of changing the conceptual into the logical perspective or finishing the analysis phase, the point of changing the logical into the physical perspective or finishing the design phase are natural control points in the IS development process. In these points, the formal report of the status of development process may be produced and the result reported to project sponsors.

4.5 Focus of IS development methods

The IS development methods differentiate by IS characteristics they are focused on and the perspectives they cover. They also differ in what their primary interests are. Traditional structured methods, such as system analysis and design, are focused on the functions of IS, information engineering methods on data, object methods on data and functions together; and some methods like Yourdon modern system analysis are focused on events. Some methods cover only one perspective, such as structured analysis that covers only the conceptual perspective. With the help of the Framework, the focus of IS development methods may be easily recognized and explained.

4.6 Improving IS maintenance

One of the biggest challenges facing the modern enterprise and its IS is change. The enterprise's IS has to trace all enterprise's changes. The IS change starts with the engineering descriptions (artefacts) of the IS, depicted in the Framework's cells. The concept of architecture is central to production of quality artefacts, IS and timely results, as well as to management of changes in IS.

5. Impact of Zachman Framework on improving IS teaching programme

The Zachman Framework offers many possibilities that are useful in teaching IS. Each of the 36 cells in the Framework in the Fig. 3 equates to one model (i.e. artefact), representing one aspect of the IS. Some often used graphical artefacts are depicted in italics. The Framework may be considered vertically, concentrating on product characteristics. Some courses covering one column of the Framework are shown bellow:

- **Data** characteristics: courses Databases, Data warehouses, Database administration, Data Mining
- **Function** characteristics: courses Systems function analysis, Software engineering, Algorithms, Programming

- **Network** characteristics: course Networking
- **People** characteristics: courses including organisational and human resources issues
- **Time** characteristics: course Event management
- **Motivation** characteristics: courses Motivation, Strategy, Organizational culture, Business rules

The lack of standard artefacts in some Framework's cells is evident, as in motivation column cells for business rules modelling.

The Framework may be considered horizontally, concentrating on the workers' perspectives or roles. Some courses covering one row are:

- **Contextual** perspective: courses Systems planning, IS strategy
- **Conceptual** perspective: course Systems analysis
- **Logical** and **physical** perspective: all courses which consider design (IS or some characteristics, such as data, function, network, etc.)
- **Component** perspective: courses Implementing IS, Programming, User documentation

At the Graduate school of Business and Economy at the University of Zagreb we found the Framework helpful for investigating the IS teaching programme at the undergraduate level. Within the *Business information systems* concentration the courses thought are: *Executive information systems* (covering Framework's cells in contextual and functioning perspective), *Databases* (data characteristics), *Software engineering* (function characteristics), *Business process simulation* (all characteristics in conceptual and logical perspective for business process simulation), *Information systems management* (motivation characteristics in contextual and conceptual perspective), *Expert systems* (data and function characteristics in conceptual and logical perspective for business process functional area), *Networking* (network characteristics), *Office automation* (people characteristics for collaboration and communication functional area). For precise description of each course we take into consideration three information system functional areas: *business process* area, *decision support* area; and *collaboration and communication* area.

The analysis of the programme shows that some regions of the Framework, such as the motivation column, time and people columns, are less covered. Some questions have to be answered. Have we to incorporate new topics or new courses to cover the holes in the programme? Need we examine completely the teaching programme?

6. Conclusion

The Zachman Framework for IS architecture is a set of design artefacts, or descriptive representations, that are relevant for describing an IS. The Framework helps to produce the IS according to requirements (quality issue) and to maintain it over the period of its useful life (change issue) [4].

The Framework is a classification schema (i.e. a tool) for defining and organizing reusable models that collectively describe an enterprise and its IS. It is a good thinking tool useful in IS development and in teaching IS development.

Everyone (i.e. every organization) has to define exactly how each artefact or model should be structured and what information it should contain. Everyone has to define the metamodel for each cell, or at least the cells deemed worthy of defining. Therefore, the Framework is not a definite solution. Later, IS development can begin after metamodels have been defined to the appropriate degree of detail.

The problem is the lack of standard artefact description for some Framework's cells. However, this is the evidence of inadequate methodological foundations of parts of the Framework and constant need for their improvement.

7. References

[1] Inmon W. H, Zachman J. A., Geiger J. G. Data Stores, Data Warehousing, and the Zachman Framework: Managing Enterprise Knowledge. McGraw-Hill; 1997.

[2] The Framework for Enterprise Architecture Cell Definitions. http://www.zifa.com (zifa03.doc) [2003-01-08]

[3] Zachman, J. A. A Framework for Information Systems Architecture. IBM Systems Journal, Vol. 26, No. 3, 1987.

[4] Zachman, J. A. Enterprise Architecture: The Issue of the Century. http://www.zifa.com (zifa06.doc) [2003-01-08]

[5] Zachman, J. A. Concepts of the Framework for Enterprise Architecture. http://www.zifa.com [2003-01-08]