UDC:007.5
Preliminary communication

MOTIVATION ISSUES IN THE FRAMEWORK OF INFORMATION SYSTEMS ARCHITECTURE

Mladen Varga
University of Zagreb
Faculty of Economics, Zagreb
mladen.varga@efzg.hr

Abstract. *The Zachman Framework for information systems architecture is a scheme for classifying and organizing the design artefacts created in the process of designing and producing information systems. It classifies artefacts on two views or dimensions: perspectives or roles and characteristics or abstractions. Although motivation abstractions are often neglected, the motivation should be the most influential driver in designing information system. We suggest business rules approach, which breaks away business rules from information system's data and processes and places business rules in the centre of users' interests. The responsibility for defining and maintaining business rules must be taken over by business people.*

Keywords. *information systems, Zachman framework, information systems architecture, motivation abstraction, business rules.*

1. INTRODUCTION

The key to overcome the complexity and change of a complex product, such as an enterprise or an information system, is architecture. If the product is so complex that its author cannot remember all details, he/she has to write down its architecture. John Zachmann presented the Framework for Information Systems Architecture [9], and extended it to the well known Framework for Enterprise Architecture[10]. The Framework is a classification scheme that describes various views on a business and its systems, such as the information system. The paper deals with the important although neglected aspect of the architecture, with its motivation. If an enterprise imposes some business rules that guide business behaviour, it ought to be able to say why, i.e. to state *motivation* [3]. The aim of the paper is to stress the importance of the motivation issues as a driver in designing information system, and to survey the methods and techniques dealing with motivation aspects.

The paper is organized as follows. Some aspects of information systems important to the problem considered here are described in Section 2. The Zachman Framework of information systems architecture is introduced in Section 3, and abstractions in the Framework are described in Section 4. The motivation issues are described in Section 5.

2. INFORMATION SYSTEM

The usual definition says that *the 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*. This descriptive definition of information system does not consider its fundamental purpose and genesis [1]. We consider that it is not adequate for information system developers. Therefore, we suggest using the genetic definition of information system: *An information system is a subsystem of the organizational system, whose task is to link processes on the operational, management and decision-making level. Its goal is improving performance efficiency, supporting good quality management and increasing decision-making reliability* [1]. The class of systems, such as an enterprise, are called organizational systems. They are goal-oriented, dynamic, multi-level hierarchical, with information-feedback and control, active in unstable environment, learning intensive, self-organizing. Each organizational system involves people, business processes (business technology) and technical resources to operate in unstable environment in order to achieve specific goals. Effective organizational system management has to be supported by a well-designed information subsystem. The structure of the organizational system, its goals and tasks, and the way of achieving them determine the information (sub)system of the organizational system. Therefore, the information system is a complex system that has to cover all informational tasks needed to service operational, management and decision-making activities of the enterprise. Moreover, the information system is an information-based model of the enterprise, which describes the enterprise's business through data. These facts are important because our intention is to discuss motivation issues in the information systems development.

3. ZACHMAN FRAMEWORK FOR INFORMATION SYSTEMS ARCHITECTURE

The key issue in the information system development is its architecture. The Zachman framework for information systems architecture [5, 10, 11] may help in development and/or documenting an enterprise-wide information systems 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 information system. The Framework enables focused concentration on selected aspects of the information system without losing sense of the contextual or holistic perspective.

The Zachman Framework is 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 information system). Although often looked at as a framework for information systems, the Zachman Framework is successfully extended to the Framework for Enterprise Architecture [12].

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* or product abstractions, which include the *what*, *how*, *where*, *who*, *when* and *why* characteristics. They are explained as *what* it is made of (structure), *how* it works (processes), *where* the components are (flow, locations), *who* does what work (people, operations), *when* things happen (dynamics, time) and *why* various choices are made (motivation). In the other words some *things* (structure) transformed by some *processes* (transform) in some *locations* (flow) by some *people* (operations) at some *time* (dynamics) for some *reasons* (motivation). Fig. 1 shows some artefacts for product characteristics in house construction and in information systems development.

Product characteristics	Description	Question	Artefacts in house construction	Artefacts in information systems devel-
Structure (Things)	Material dscription	What	House, room	Data entity
Transform (Processes)	Functional description	How	Eat, play, sleep	Computer program, manual procedure
Flow (Locations)	Spatial description	Where	Placement of rooms	Network of locations
Operations (People)	Operational description	Who	Occupants, guests, pets	User, organization
Dynamics (Events, Time)	Timing description	When	When to eat, play, sleep	Event
Motivation	Motivation description	Why	Accommodate growing family	Business goal, business rule

Figure 1: Product characteristics of the Framework

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). Fig. 2 shows these perspectives; and models produced in each perspective in business and development terms.

The Framework for information systems 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 or abstractions, 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 information system) in its entirety. It helps to consider 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 or analytical tool in dealing with complexities and dynamics of a complex object.

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 Framework

4. ABSTRACTIONS IN THE FRAMEWORK OF IS ARCHITECTURE

Upon completion of the information system architecture the Framework cells has to be populated with appropriate artefacts within the scope of the new system. The good architecture consists of a complete set of explicitly stated models, vertically and horizontally integrated, at a very high level of detail.

Each row in Fig. 3 represents the view on IS from specific *perspective* that belongs to a unique *role*. The role of a *planner* in the contextual perspective is to set the scope (contextual model) or the strategy of an information system. 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 information system is built in concordance with all the mentioned models.

The columns in the Framework represent the different information system's characteristics or 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 *process* column describe the various aspects of operations of the information system (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 information system. The *time* column describes the effects of time on the information system. 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.

Model (perspective) *Role*	*What* **Data (entities)**	*How* **Process (activities)**	*Where* **Network (locations)**	*Who* **People**	*When* **Time**	*Why* **Motivation**
Scope (contextual) *Planner*	Class of business things	Class of business processes	Major business locations	Major organization units	Major business events	Major business goals
Enterprise model (conceptual) *Owner/analyst*	Semantic data model, conceptual data model	Business process model	Business logistics system	Workflow model	Master schedule	Business plan
System model (logical) *Designer*	Logical data model	Application architecture	Distributed systems architecture	Human interface architecture	Processing structure	Business rules
Technology model (physical) *Builder*	Physical data model	Systems design	Technology architecture	Presentation architecture	Control structure	Rule design
Component model (component) *Subcontractor*	Data definition	Programs	Network architecture	Security architecture	Timing definition	Rule specification
Functioning Enterprise (functioning) *User*	Data	Function	Network	Organization	Schedule	Goals

Figure 3: The Framework of information systems architecture

5. MOTIVATION ABSTRACTIONS OF THE FRAMEWORK

Effective organizational system management has to be supported by a well-designed information subsystem. The structure of the organizational system, its goals and tasks, and the way of achieving them determines the information system of the organizational system.

As the genetic definition of information system stated, the information system's goal is to improve business process efficiency, support good quality management and increase

decision-making reliability. Some consequences of the definition [1] are:

An information system cannot exist by itself. It is always a subsystem of some real organizational system, i.e. each organizational system has its unique and distinctive information system.

An information system is always a model of business processes of some organizational system.

A considerable knowledge of the organizational system's functions and structure are necessary in designing its information system.

Motivation is and ought to be the most influential driver in the development of a new or improving of an existing information system. As seen in Fig. 3, the motivation abstractions are about business itself, not its information systems or its technology. They ought to be able to answer the question "why", i.e. to state the *motivation* of the business.

The motivation cells of the Framework may be populated as follows. The responsibility of the planner in the contextual perspective is to set major business goals. In the conceptual perspective, the analyst is responsible to develop the business plan of the enterprise. In the logical perspective the designer's task is to realize the business plan by a set of business rules. Business rules are designed in the physical perspective and finally specified in the component perspective.

The motivation is the driver in the development of the other information system's characteristics, i.e. data, process, network, people and time columns. We may argue that the motivation column must be defined before other columns. Very first step includes defining major business goals and business plan of the enterprise, which cover first two rows of the motivation column. Next step involves defining contextual perspective in the other columns (class of business things in data column, class of business processes in process column, etc.). Defining business rules precedes defining models in conceptual perspective (semantic data model, business process model, etc.). We may conclude that defining motivation abstraction in the conceptual perspective is the source of information for defining other's columns abstractions in the contextual perspective, defining motivation abstraction in the logical perspective is the source of information for defining other's columns abstractions in the conceptual perspective, etc.

5.1. MAJOR BUSINESS GOALS

In the contextual perspective the motivation is represented mainly by stating an enterprise's vision, which is made operative by mission and strategy. No method standardizes the way of expressing vision, mission and strategy. They are usually expressed literally by sentences in natural language.

5.2. BUSINESS PLAN

Vision, mission and strategy are translated into a business plan in the conceptual perspective. Standards in this area hardly exist although many planners have used planning methodologies over the years. A good attempt to standardize the elements of a business plan is presented by the Business Rules Group in the document entitled "Organizing Business Plans; The Standard Model for Business Rule Motivation" [3]. The model provides a scheme or structure for developing and managing business plans in an organized manner. It identifies factors that motivate the establishing a business plan, defines the elements of a plan, and indicates how these factors and elements inter-relate. Some of the proposed elements of a business plan are as follows: *vision* is a statement about the future state of the enterprise; it is made operative by *mission*, and amplified by *goals*. A goal is supported by

strategies. It is mainly qualitative and may be quantified by objectives. An *objective* is achived by *tactics*.

All elements of the proposed standard model for business rule motivation are developed from a business perspective. The idea is to develop a business model of the elements of the business plan before any system design or technical development. Therefore, the business plan is the foundation for system design and development. System solutions are in that case firmly connected to their business intent.

5.3. BUSINESS RULES

Objectives of a system are achieved by tactics, which are guided by business policy and implemented through business rules. Generally, a business plan is implemented through business rules although business rules may play an important role in development of business plan. Business rules are main actors in the logical perspective. They describe knowledge about business.

From the business perspective the business rule is *a directive intended to influence or guide business behaviour, in support of business policy that has been formulated in response to an opportunity, threat, strength, or weakness of the business* [9]. Business rules represent core business concepts and policies. They represent the basic business vocabulary and rules that control or guide business behaviour. They indicate what is possible or desirable in running the business.

From the information system perspective the business rule *is a statement that defines or constrains some aspect of the business. It is intended to assert business structure, or to control or influence the behaviour of the business* [9]. A business rule pertains to the facts of the business system that are recorded as data and to the constraints on changes to the values of those facts.

A "rule model" is a kind of enterprise metadata, while all other columns belong to "information system" columns. The Business Rules Manifesto of Business Rules Group [2] prescribes rule independence, i.e. the business rules should be expressed independently of any other model type. Historically, business rules have been found in the artefacts of others columns such data, process or event columns. However, there is a tendency to treat business rules as a separate artefact, but still related to other characteristics of a given perspective.

Business rules must describe the way business itself is run, not just the information system. It is intended that the business users are the owners of business rules. Since rule statements ought to be in a plain language understandable by business users (preferably a kind of natural language), it is easier for users to accept ownership of a "rule model" than accept any other model. Business rules must be understandable to business people more than any other abstraction of the developing information system. In fact, business people have to be responsible for defining and maintaining business rules.

Enterprises that take a model-based, architected approach to software component development can use business rules to refine the models and create better designs. An enterprise that properly documents its business rules can also manage change better than one that ignores its rules [7]. Business rules can be defined, modelled and implemented as metadata for an enterprise's information system. Implementing business rules as metadata is the most rigorous and, at the same time, most flexible approach to business rule implementation. This is in contrast to other traditional process-driven or procedure-driven implementation approaches.

Business rules defined and managed separately allow design and generate applications from the business rules alone. This is essential for enabling business and information system architecture to be truly adaptive. Business rules offer several benefits, such as technical

independence, better quality requirements, ease of change. Generally, rules are more important to the business than technical equipment.

Business rules must be explicitly expressed, either graphically or as a formal language, declarative in nature, and with coherent representation model. Some characteristics of business rules are [2]:

Rules are basic to what the *business knows* about itself, i.e. to the basic business knowledge. They are motivated by identifiable and important business factors, business goals and objectives, although they are shaped by various influences.

Rules are explicit *constraints* on business behaviour. They define the boundary between acceptable and unacceptable business activity.

Rules can exist *independently* of procedures and workflows. They generally apply across processes and procedures, but they are neither processes nor procedures. They should be defined independently of "who", "where", "when" or "how" responsibility.

Rules should be specified by the *business people* who have relevant business knowledge.

Rules should be *explicit*.

Rules should be expressed *declaratively* and for the business audience, in natural language sentences. If something cannot be expressed, then it is not a rule.

Rules must be *managed*. A business rule system is never really finished because it is intentionally built for constant change. Rules, and the ability to change them effectively, are important to improve business adaptability.

It is important to stress that business rules are an underlying reality in an enterprise. They are independent of an analyst's structuring and describing them. Identifying business rules is mostly iterative and heuristic, beginning as general statements of business policy. Business rules are typically described in a general and informal fashion. They may be described in many different forms. Users define them as the statements in a natural language. The analyst's responsibility is to express business rules in a formally and rigorously defined manner as a set of atomic business rules.

Business rules can be categorized in different ways, the most known is of Ross [8]. However, there is no standard for expressing atomic business rules. The best attempt, although incomplete, of defining business rules in a formal manner is those by the Business Rules Group [4]. The Group concerned with business rules that affect the storage of persistent data, described in a technology-neutral way, and with no rules of a business that do not have an information system component. They divide business rules into three types: *structural assertions* (defined concepts or statements of facts that express some aspects of the structure of an enterprise), *action assertions* (statements of constraints or conditions that limit or control the actions of the enterprise), and *derivations* (statements of knowledge that is derived from other knowledge in the business).

A structural assertion is a statement that some *thing* is of importance to the business or exists in relationship to another thing of interest. An assertion is expressed by *terms* and *facts*. Terms, such as customer, order or account, express business concepts. Facts make assertions about these concepts and rules constrain and support these facts. This is essentially a structured business vocabulary. Examples of facts are "*A customer places an order"* and *"A customer holds an account"*; and the rule "*A customer may place an order only if the customer holds an account*".

An action assertion describes some dynamic aspect of the business. It specifies constraints on the results that actions can produce. Where the structural assertions describe possibilities, action assertions impose constraints expressed by "must" or "must not". An

example is "A car in the traffic must be registered".

A derivation is a kind of business rule where a derived fact is created by an inference from existing facts. An example is "*The total rental cost is calculated from the rental rate multiplied by the number of days*".

Each business rule may be expressed as a formal rule statement, which is an expression in a specific formal grammar. Formal grammars suitable for representing business rules are structured natural language, IDEF1X, grammars used in CASE tools, Object Role Modeling, Ross's notation, etc. An example [4] of a business rule in a rent-a-car company might be: "*A car with accumulated mileage greater than 5000 since its last service must be scheduled for servicing*". This rule may be expressed in the structured English: *If Car.mileage-current-period > 5000 then invoke Schedule-service (Car.id) End if.*

It may be helpful to use a kind of templates for expressing business rules. Here are some examples of templates for few types of business rules [6] and examples of their usage.

<Term> is defined as <textual definition>
<Term1> is referenced in the <Term2>
<Term> is computed as <formula>
IF <rule phrase> [AND <rule phrase> AND <rule phrase>…] THEN <action>

<Price Year> is defined as fiscal year. For example, price year goes from January,1 to December,31.
<Contract Price> Is Referenced In the <Agreement>
<Gross Sales> is computed as Sum (<Contract Price> x <Net Sales Quantity>)
If Fiscal Year End AND Portfolio Owner THEN Send Annual Report

Rules need to be managed in an automated repository, allowing business users and analysts to directly access and manipulate rules.

5.4. BUSINESS RULES DESIGN AND SPECIFICATION

Business rules are designed in the physical perspective. The designer's responsibility is to identify an appropriate technology for implementing the business rules in an information system.

Business rules specification belongs to the component perspective. Specifications are incorporated in a number of ways. Some structural assertions may be implemented through database schema, database triggers, object methods, etc. Action assertions and derivations are probably implemented procedurally.

Business rules are finally enforced in the functioning enterprise accompanied by its information system. In a working environment, there exists binary logic: a business rule is fulfilled or not.

6. CONCLUSION

The Zachman Framework for information system architecture is a set of design artefacts relevant for describing an information system. Present information systems methodologies do not equally populate and understand all Framework cells. A great deal is known about data and process columns and few cells in other columns, while many cells are less understood. From the business viewpoint, the most important is the motivation column describing business goals, objectives and strategies, finally specified by business rules.

Motivation abstractions represent the goals of the enterprise, which have to be supported by the information system. Unfortunately, the motivation issues are inadequately

considered in the information systems theory and practice. We are witnessing of the lack of appropriate theoretical paradigms, methods and techniques that deal with the motivation abstractions although the motivation is the most influential information system driver. Therefore, we have to stress the importance of the motivation abstractions to both information systems developers and information systems students.

The genetic definition of the information system says that an information system has to fulfil all informational tasks according to business processes efficiency, good quality management and decision-making reliability. The recommended way to achieve these goals is to specify the information system's requirements beginning with business goals, and business plans; and the way of doing the job is to define an enterprise's business rules model.

Business rules are important inputs for information system development. They describe knowledge about business, therefore they figure as information system requirements. Business rules approach separates business rules from information system's data and processes, and places them in the centre of the users' interests. Therefore, business people have to take the responsibility for defining and maintaining business rules.

REFERENCES

[1] Brumec, J. (1997): *A contribution to IS General Taxonomy*. Zbornik radova Fakulteta organizacije i informatike Varaždin, vol. 21, no. 1(22), 1-14.

[2] Business Rules Group (2003): Business *Rules Manifesto; The Principles of Rule Independence*. http://www.businessrulesgroup.org/brmanifesto/BRManifesto.pdf [2003-06-15]

[3] Business Rules Group (2000): Organizing Business Plan; The Standard Model for Business Rule Motivation. http://

[4] Business Rules Group (2000): Defining Business Rules; What Are They Really?

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

[6] Reeder, J. (2003): Templates for Capturing Business Rules. http://www.brcommunity.com/cgi-bin/x.pl/print/p-b056.html [2003-06-15]

[7] Perkins, A. (2002): *Business Rules Are Meta Data*. Business Rules Journal, Vol. 3, No. 1, http://www.BRCommunity.com/a2002/b097.html [2003-06-15]

[8] Ross, R (1997): *The Business Rule Book: Classifying, Defining and Modeling Rules, 2nd Ed.,* Business Rule Solutions.

[9] Ross, R. (2003): *The Business Rule Approach*. Computer, May 2003, 85-87.

[10] The Framework for Enterprise Architecture Cell Definitions (2003). http:// www.zifa.com (zifa03.doc) [2004-01-08]

[11] Zachman, J. A. (1987): *A Framework for Information Systems Architecture*. IBM Systems Journal, Vol. 26, No. 3.

[12] Zachman, J. A (2003): *Concepts of the Framework for Enterprise Architecture*. http://www.zifa.com [2004-01-08]

Received: 17 December 2003
Accepted: 3 July 2004