Towards a Representation of Enterprise Architecture based on Zachman Framework through OMG Standards

Miguel Ehécatl Morales-Trujillo[a,b], Boris Escalante-Ramírez[b], Pilar Ángeles[b], Hanna Oktaba[c] and Guadalupe Ibargüengoitia-González[c]

[a] University of Canterbury, Christchurch, New Zealand
miguel.morales@canterbury.ac.nz
[b] Facultad de Ingeniería, Universidad Nacional Autónoma de México, Mexico City, Mexico
{boris, pilarang}@unam.mx
[c] Facultad de Ciencias, Universidad Nacional Autónoma de México, Mexico City, Mexico
{hanna.oktaba, gig}@ciencias.unam.mx

***Abstract* — This paper presents a rearrangement of the Zachman framework carried out in order to facilitate the representation of an enterprise architecture (EA). The approach proposes a diagram-based alternative to model EA using two well-known Object Management Group standards: UML and BPMN. The proposal is directed to organizations in charge of developing and maintaining software systems under the premise that the people who develop software are highly familiarized with both standards. The proposal reduces 30 elements required by Zachman to 17, which are 7 diagrams and 10 documents, providing organizations with a viable alternative to model their structure, processes and environment in a business-oriented vision.**

Keywords— Enterprise Architecture, Zachman framework, software development, UML, BPMN, modeling language

I. Introduction

Organizations constantly look for an improvement and one of the strategies for a more efficient management is process automatization through Information Technology (IT) [1]. Automatization of processes aims at fulfillment of objectives, metrics and requirements of an organization. However, for a process to be correctly aligned, it should be well-defined in a common language and amendable to detailed analysis.

Considering this premise, it becomes an acute necessity to establish a mechanism for representing the elements that comprise an organization. This representation of the elements, their relationships and executional context constitute the Enterprise Architecture (EA) of the organization. EA is a structured and aligned collection of plans for the integrated representation of a given business and IT landscape [2]. EA can be a useful tool for aligning the IT application and organization's activities. Besides, it can facilitate business success to the effectiveness by using information of management strategic and IT resources [3].

In the process of creating an EA representation for software development organizations some obstacles are faced. In the first place, it is imperative to define a method to build the elements of an EA. Even though, as exposed in [4], the way in which an EA specification would be built is not relevant, this premise does not hold at the moment of integrating the elements. If the EA structure is not properly defined since the beginning, it will be difficult to maintain relationships between future elements, which may substantially weaken the cohesion among them.

Moreover, in [5] it is established that the current EA implementation methods have a broad scope and a lack of structure; this usually causes complication and difficulty in implementation. Besides, the fact that there is no comparison between existent EAs complicates the initiative to define EA and leaves those defining it for the first time without a point of reference or comparison.

Last but not least, there is a necessity to offer alternatives to manage an EA evolution. Nowadays there is a considerable gap between defining an EA and its management over time. Any changes within the organization directly affect the previously defined EA, which is why there is a need for a specific mechanism to be able to manage the change and to allow for the EA to evolve alongside the organization [6].

Addressing this necessity, this paper presents an alternative to represent the EA of an organization, which is based on Zachman framework [7] in terms of two widely used Object Management Group (OMG) standards: the Unified Modeling Language (UML) [8] and Business Process Model and Notation (BPMN) [9].

This paper is organized as follows: Section II presents the background of the proposal and its fundamentals. Section III details the proposed adaptation of the Zachman framework and the mechanism to represent the EA. Section IV contains preliminary results related to the applicability of the proposal. Finally, section V presents conclusions and future work.

DOI reference number: 10.18293/SEKE2018-001

II. Background

In this section an overview of EA definitions, the Zachman framework and the relevant for this paper OMG standards are presented. Related to the Zachman framework work and OMG modeling languages are described as well.

A. Enterprise Architecture

EA is defined as a process of strategic planning that integrates the business management with IT in order to improve the organization's financial and enterprise efficiency. EA is a set of descriptive representations relevant for describing an enterprise so that it can realize management requirements and be maintained over the period of its useful life [10]. EA is an approach to enterprise information systems management that relies on models of the information systems and their environment [11]. EA supports the analysis, design and engineering of business-oriented systems through multiple views [12].

In order to create an EA, frameworks like Zachman, TOGAF ADM [13], DoDAF [14] or MODAF [15] have been created. An EA framework is a model used by an organization to develop good corporate governance, creating added value for their business [16]. For the purposes of this proposal, the Zachman framework will be the base to represent an EA and is detailed in the next subsection.

B. The Zachman framework

The Zachman framework is an ontology that represents EA concepts and their relationships. The ontology is developed through an empirical approach and answers the questions *who? when? why? what? how*? and *where?*. This question-based approach, according to Zachman, allows for a full and understandable description of complex ideas, which is the case with EA.

Each question aims at finding the necessary Data (what?), the Function (how?), the People (who?), the Network (where?), the Time (when?) and the Motivation (why?) involved in the EA. According to [17] each element is defined as follows:

Data (Thing—Relationship—Thing): this element focuses on the material composition of the product.

Function (Process—Input/Output—Process): this element focuses on the functions or transformations of the product.

People (People—Work—People): this element focuses on the people, the manuals and the operating instructions or models they use to perform their tasks.

Network (Node—Line—Node): this element focuses on the geometry or connectivity of the product.

Time (Event—Cycle—Event): this element focuses on the life cycles, timing and schedules used to control activities.

Motivation (End—Means—End): this element focuses on goals, plans and rules that prescribe policies and ends that guide the organization.

In addition, the framework proposes 5 models, which, according to [17], are defined as follows:

Scope (Contextual): describes the models, architectures and representations that provide the boundaries for the organization.

Business model (Conceptual): describes the models, architectures and descriptions used by the individuals who are the owners of the business process.

System model (Logical): 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): describes the models, architectures and descriptions used by technicians, engineers and contractors who design and create the actual product.

Detailed representations (Out-of-context): describes the actual elements or parts that are included in, or make up, the final product.

The 6 questions and the 5 models comprise a two-dimensional matrix of 30 cells. Each cell describes or represents a particular element, which can be defined by means of diagrams, documents or work products, according to the organization preferences.

The EA representation that is achieved following this framework is a static view of the organization, as a consequence it is impossible to model its operational processes. The major advantage, however, is the possibility to represent the EA fundamental elements in a precise and well-defined manner.

The Zachman framework is a well-known alternative for modeling an organization's EA, however, it faces three considerable weaknesses: low cohesion of its elements, lack of a method to use it and lack of specificity in the cells description.

In addition, there are no detailed examples demonstrating the successful practical application of the Zachman framework [4], which is a strong limiting factor.

C. OMG standards and Related work

OMG develops IT standards for a broad variety of industries. Two of the most well-known and broadly used in software engineering related industries are UML and BPMN. UML is one of the most used specification in IT industry; software engineering practitioners know the specification and use it daily in their projects.

In order to provide a unified language architecture, the OMG developed the Unified Profile for DoDAF and MODAF (UPDM) [18]. The UPDM specification reuses a subset of UML and provides additional extensions to allow the representation of architecture models. UPDM is based on UML class diagrams, where each class represents common elements of DoDAF and MODAF; however, the only way to differentiate between classes is through stereotypes. As a result, the main drawback of UPDM is a lack of expressiveness.

A more expressive language is ArchiMate [19] that is an EA specific modeling language created by the Open Group. ArchiMate is aligned with TOGAF. In this case, an important obstacle that hinders its spreading is its level of complexity as well as its lack of familiarity to practitioners. Besides, ArchiMate is targeted to big companies.

Based on the assumption that one quarter of all EA representations is done through the Zachman framework, the OMG published a proposal that represents the Zachman framework cells through OMG modeling specifications [20].

The proposal reuses UPDM, BPMN and UML mainly. However, there are also Zachman's cells that are not represented at all.

III. Proposed solution

The enterprise willing to model its EA has to cope with two challenges: it must define procedures for gathering the needed information and must devise a conceptual model defining the necessary information [21].

Modeling EA requires representing multiple diagrams of an enterprise, which typically shows the multiples business entities, IT systems and the services they offer [22]. Therefore we propose to model these aspects by using UML and BPMN mainly. On the one hand, the proposed solution pursues the goal of adjusting the EA elements to the organization's context. In other words, we intend that the created EA views become meaningful to the organization and fulfill its necessities or objectives.

On the other hand, the representation mechanism should use a language that is close to the organization's members, so that the EA representation can be easily assimilated and applied, demanding as little effort as possible. It is also possible to use a more complex notation, which may allow to keep more aspects together thus reducing the number of cells; however, our main goal was to increase comprehension and simplicity in the representation.

In the following lines, we describe the proposed readjustment to the Zachman framework together with the alternative that was chosen to represent each of the elements.

A. Rearrangement

The first step taken to create this proposal was to rearrange the elements of the Zachman framework. This was done in order to reduce the number of cells proposed by Zachman. A grouping of the cells of the framework in first place would reduce the complexity and the number of work products to represent the EA.

On the other hand, taking into account software engineers' reasoning, several elements were combined. For example, Business model and System model of the Data column are represented by an Entity-Relationship diagram in the context of software development, while, within the Zachman framework, they are separated cells.

Another example is that it comes natural for software engineers to model processes in terms of three fundamental elements: activities, work products and roles. Even though the Zachman framework places these elements on the layer Scope, they are separated into Function (what is done) and People (who does that).

This separated representation renders incomplete and loses the perspective of what is being modelled. Therefore, the proposed rearrangement merges both elements into one, which is might be represented by a BPMN diagram integrating what is being done and who is in charge of it.

During the software design phase, the database is often modelled by the means of tables and class diagrams. The class diagrams represent a table through its attributes and methods. Therefore, the Physical Data Model and the System Design, attached to the questions *What?* and *How?* in the Technology model, can be combined and represented as a class diagram. This class diagram will show the database and the tables that express the persistency of the system.

We propose to integrate and rearrange several of the 30 Zachman's cells, finally obtaining 17. For the proposed model to be practical, we name each element as presented in [17]; in case the elements are merged, their names are introduced through a slash. In the following subsections the 17 elements are described in more detail.

Figure 1 shows the obtained rearrangement, the first work products to be created are lists, shown in green; the blue ones are diagrams and red ones are documents.

B. Representation using diagrams

In this subsection we describe the elements represented through diagrams.

1) Semantic model / Logical data model: represents entities (things) and their relationships that are involved in the Business model and its logic representations in the System model. It is carried out by means of an Entity-Relationship diagram.

2) Physical data model / System design: represents the databases and the domains of the Technology model. The representation is carried out by means of a class diagram.

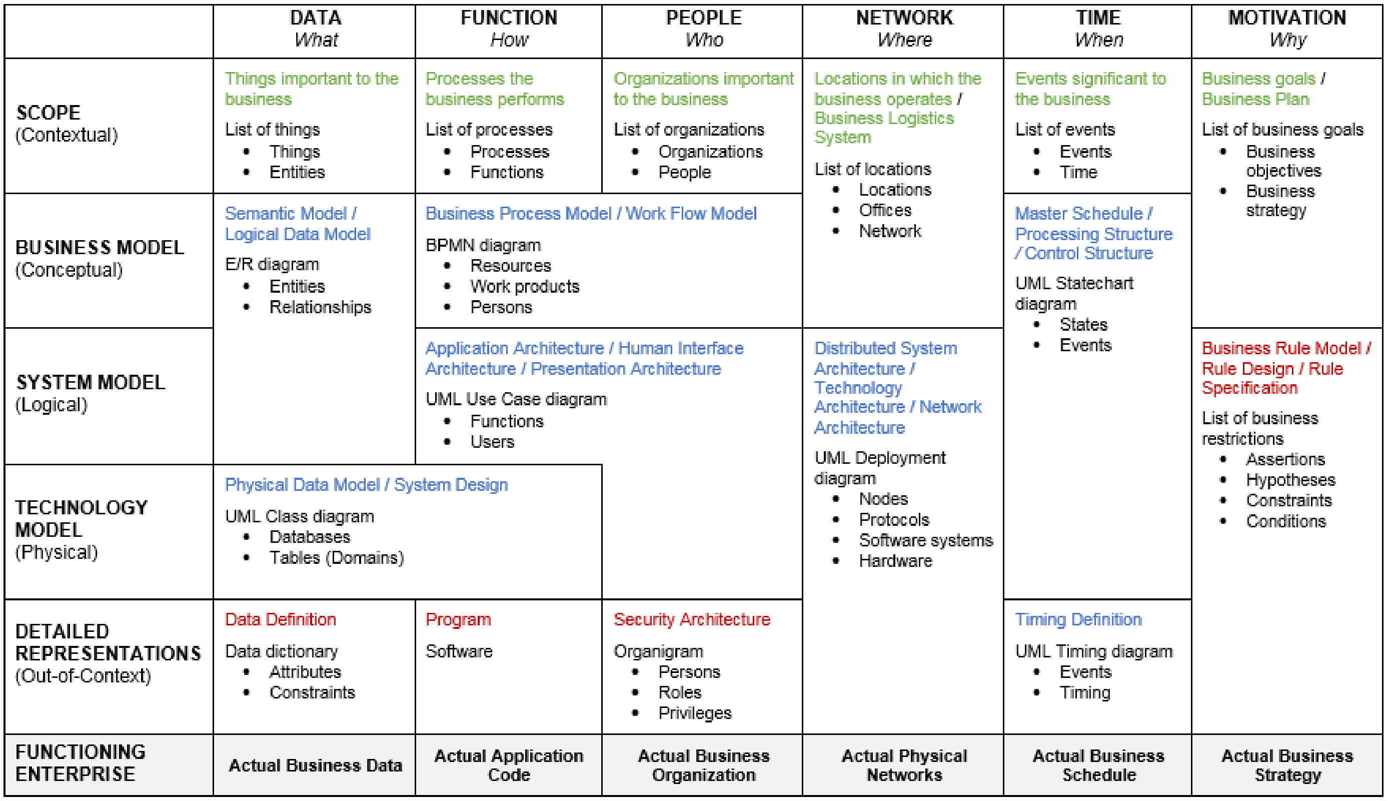
3) Business process model / Work flow model: represents the processes, resources, persons and work products involved in the business model. The representation of these elements is carried out by means of a BPMN or a UML activity diagram.

4) Application architecture / Human interface architecture / Presentation architecture: it is a combination of the System model with the Technology model and represents the functions of the system and its users. Both aspects are represented by means of a Use case diagram.

5) Distributed system architecture / Technology architecture / Network architecture: the nodes, communication protocols, hardware and software that are necessary to allow communication between different locations of the organization are represented by means of a deployment diagram. This diagram represents a physical distribution of objects alongside with how they relate and communicate with each other.

6) Master schedule / Processing structure / Control structure: the organization's time-related aspects can be represented through a statechart diagram. The states that business or organizational systems go through as well as the events that cause a change of a state are the main components of the Business, System and Technology models.

7) Timing definition: This element describes events and their times. It is possible to represent the active state of each organizational process by means of a time diagram.



	DATA *What*	FUNCTION *How*	PEOPLE *Who*	NETWORK *Where*	TIME *When*	MOTIVATION *Why*
SCOPE (Contextual)	Things important to the business List of things • Things • Entities	Processes the business performs List of processes • Processes • Functions	Organizations important to the business List of organizations • Organizations • People	Locations in which the business operates / Business Logistics System List of locations • Locations • Offices • Network	Events significant to the business List of events • Events • Time	Business goals / Business Plan List of business goals • Business objectives • Business strategy
BUSINESS MODEL (Conceptual)	Semantic Model / Logical Data Model E/R diagram • Entities • Relationships	Business Process Model / Work Flow Model BPMN diagram • Resources • Work products • Persons			Master Schedule / Processing Structure / Control Structure UML Statechart diagram • States • Events	
SYSTEM MODEL (Logical)		Application Architecture / Human Interface Architecture / Presentation Architecture UML Use Case diagram • Functions • Users		Distributed System Architecture / Technology Architecture / Network Architecture UML Deployment diagram • Nodes • Protocols • Software systems • Hardware		Business Rule Model / Rule Design / Rule Specification List of business restrictions • Assertions • Hypotheses • Constraints • Conditions
TECHNOLOGY MODEL (Physical)	Physical Data Model / System Design UML Class diagram • Databases • Tables (Domains)					
DETAILED REPRESENTATIONS (Out-of-Context)	Data Definition Data dictionary • Attributes • Constraints	Program Software	Security Architecture Organigram • Persons • Roles • Privileges		Timing Definition UML Timing diagram • Events • Timing	
FUNCTIONING ENTERPRISE	**Actual Business Data**	**Actual Application Code**	**Actual Business Organization**	**Actual Physical Networks**	**Actual Business Schedule**	**Actual Business Strategy**

Fig. 1. Rearrangement of the Zachman framework

C. Representation using lists and documents

The Scope layer in particular describes the context in terms of lists, 4 of the 6 Scope layer elements did not undergo any rearrangement nor integration with other elements:

1. List of things important to the business.
2. List of processes the business performs.
3. List of organizations important to the business.
4. List of events significant to the business.

The remaining two were merged with the cell from an inferior row.

1) List of locations / Business logistics system: this document enlists the organizational offices and their descriptions.

2) List of business goals / Business plan: this document enlists the business objectives as well as the strategies for achieving them.

3) Data definition: contains a detailed representation of data carried out by means of a data dictionary containing fields, their descriptions and restrictions applied to them.

4) Program: the programs that the organization uses to carry out its functions are represented by means of a tools and software systems list.

5) Security architecture: the detailed representation of the people involved should contain the persons or identities, and their roles and privileges. This representation is carried out by means of an organigram.

6) Business rule model / Rule design / Rule specification: this document gathers the business rules and restrictions. By means of assertions, hypothesis and restrictions the context for achieving the organizational objectives is described.

IV. Results

The proposal of representing an EA by means of work products, in this case diagrams and documents, originates in the idea from [4]. Ylimäki states that "since the Zachman framework is not a methodology, a method is needed to fill in the framework cells". However, Ylimäki's study offers 52 work products to cover the EA definition.

Our proposal, on the other hand, reduces the number of the Zachman's cells and offers 17 work products to cover them, which we consider an advantage. Besides, although they are created specifically for EA, they are based on the languages that belong and are closely familiar to software development organizations and their work teams. In addition, the UML-based approach allows representing a wider range of enterprise concerns [24], taking advantage of its flexibility and popularity.

The preliminary results are classified in advantages and drawbacks. The following are the advantages that are identified in relation to the proposed solution: (i) software development organizations widely use BPMN and UML. In consequence, people in charge of creating, maintaining or using an EA can understand its language. Besides, diagrams increase the EA expressiveness; (ii) work products and diagrams used in this proposal are fully familiar to members of work teams, which means that minimal effort is required to comprehend and create them; and (iii) the number of the necessary work products is reduced, making an EA definition simpler and lighter. Once the proposal is validated through case studies, further reductions of work products or more optimal rearrangements could be proposed.

However, several disadvantages are identified: (i) once the EA is created, additional effort is required for its maintenance and evolution. Since there is no explicit framework to guide these processes, the organization has to manage them by its own; (ii) as mentioned before, the Zachman framework faces three considerable weaknesses: low cohesion of its elements, lack of a method to use it and lack of specificity in the cells description. Even though the identified disadvantages are significant, they are all inherited from the original Zachman framework.

V. Conclusions

This proposal is an initial approach towards how software development organizations can build their EA. EA is a complex concept that aims at modeling the structure and behavior of an organization and enables its stakeholders to make decisions. There is no one-size-fits-all template for EA. However, the Zachman framework is widely acknowledged to encompass all the concepts necessary to describe an organization [20].

The Zachman framework provides an alternative for modeling an organization's EA; however, as its own creator said: "so, if you ask who is successfully implementing the whole framework, the answer is nobody that we know of yet" (http://archive.visualstudiomagazine.com/ea/magazine/spring/online/druby3/default_pf.aspx, accessed 01/05/2018). Up to now there is no solid evidence to reject this statement. The intention of our proposal is to simplify the framework, taking advantage of the fact that it is well-known in the industry although rarely used.

This proposal integrates the Zachman framework and a work product based approach. The work products are mainly created by means of UML and BPMN, and represent the fundamental components of an EA. In addition, populating the Zachman framework with OMG modeling specifications is widely supported by software tools [20]. We believe that this joined approach will allow more software development organizations to get involved into the subject of EA and, what is more important, be able to define their own by using reachable and well-known to their work teams tools.

As future work we consider the following: (i) to create real-life examples applying the proposed rearrangement; (ii) to create a guide for work teams to be able to define their own EA; and (iii) establish a framework for managing the future evolution of already defined EAs.

Acknowledgment

This work has been developed under the "Programa de Becas Postdoctorales en la UNAM" of the Dirección General de Asuntos del Personal Académico (DGAPA) of the Universidad Nacional Autónoma de México (UNAM).

References

[1] S. Townson. Why does Enterprise Architecture Matter? The Open Group (2008)

[2] M. Zhang, H. Chen and A. Luo. A Systematic Review of Business-IT Alignment Research with Enterprise Architecture. IEEE Access, Vol. 6, DOI: 10.1109/ACCESS.2018.2819185 (2018)

[3] D. Rusli and Y. Bandung. Designing an Enterprise Architecture based on TOGAF ADM and MIPI. In: Proc. Of the Intl. Conf. on Information Technology Systems and Innovation, pp. 38–43, DOI: 10.1109/ICITSI.2017.8267915 (2018)

[4] T. Ylimäki and V. Halttunen. Method Engineering in Practice: A Case of Applying the Zachman Framework in the Context of Small Enterprise Architecture Oriented Projects. In: Proc. of the Information, Knowledge, Systems Management, Vol. 5, No. 3, pp. 189–209 (2006)

[5] S. Leist and G. Zellner. Evaluation of current architecture frameworks. In: Proc. of the ACM Symposium on Applied Computing, pp. 1546–1553 (2006)

[6] B. Rouhani, M. N. Mahrin, F. Nikpay, R. B. Ahmad and P. Nikfard. A systematic literature review on Enterprise Architecture Implementation Methodologies. Information and Software Technology, No. 62, pp. 1–20, DOI: 10.1016/j.infsof.2015.01.012 (2015)

[7] J. Zachman. A framework for information systems architecture. IBM Systems Journal, Vol. 26, No. 3 (1987)

[8] OMG. Unified Modeling Language (UML). Technical report, Object Management Group, Needham, MA, USA (2017)

[9] OMG. Business Process Model and Notation (BPMN) 2.0. Technical report, Object Management Group, Needham, MA, USA (2011)

[10] J. Zachman. Enterprise architecture: The issue of the century" Database Programming and Design, vol. 10, no. 3, pp. 44–53 (1997)

[11] A. Källgren, J. Ullberg and P. Johnson. A Method for Constructing a Company Specific Enterprise Architecture Model Framework. In: Proc. of the 10th Int. Conf. on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing, pp. 346–351, DOI: 10.1109/SNPD.2009.103 (2009)

[12] M. Bakhshandeh, G. Antunes, R. Mayer, J. Borbinha and A. Caetano. A Modular Ontology for the Enterprise Architecture Domain. In: Proc. of the Int. Enterprise Distributed Object Computing Conference Workshops, pp. 5–12, DOI: 10.1109/EDOCW.2013.8 (2013)

[13] Open Group. TOGAF 9.1 Architecture Development Cycle (ADM). Reference Card (2011)

[14] United States Department of Defense. Department of Defense Architecture Framework (2015)

[15] British Ministry of Defence. The MOD Architecture Framework (2012)

[16] I. Alonso, J. Verdún and E. Tovar. The IT Implicated Within the Enterprise Architecture Model: Analysis of Architecture Models and Focus IT Architecture Domain. In: Proc. of the IEEE Int. Conf. on Service-Oriented Computing and Applications (SOCA), DOI: 10.1109/SOCA.2010.5707174 (2010)

[17] D. Frankel, P. Harmon, J. Mukerji, J. Odell, M. Owen, P. Rivitt, M. Rosenm and R. Soley. The Zachman Framework and the OMG's Model Driven Architecture. Business Process Trends (2003)

[18] OMG. Unified Profile for DoDAF and MODAF (UPDM). Technical report, Object Management Group, Needham, MA, USA (2013)

[19] Open Group. ArchiMate 3.0.1. Specification (2017)

[20] OMG. OMG's Enterprise Architecture Specifications (white paper). Object Management Group, Needham, MA, USA (2015)

[21] S. Buckl, F. Matthes, C. Schweda. Conceptual Models for Cross-cutting Aspects in Enterprise Architecture Modeling. In: Proc. of the 14th IEEE Int. Enterprise Distributed Object Computing Conference Workshops, pp. 245–252, DOI: 10.1109/EDOCW.2010.18 (2010)

[22] H. Dam, L.-S. Lê and A. Ghose. Supporting change propagation in the evolution of enterprise architectures. In: Proc. of the 14th IEEE Int. Enterprise Distributed Object Computing Conference, pp. 24–33, DOI: 10.1109/EDOC.2010.23 (2010)

[23] J. Zachman and J. Sowa. Extending and Formalizing the Framework for Information Systems Architecture. IBM Systems Journal, Vol. 31, No.3 (1992)

[24] A. Gerber, A. van der Merwe and K. Bayes. An Investigation into UML Case Tool Support for the Zachman Framework. In: Proc. Of the Enterprise Systems Conference, DOI: 10.1109/ES.2013.6690080 (2013)