Enterprise Architecture Review

Héctor Arturo Flórez Fernández*

Fecha de recepción: octubre 29 de 2010
Fecha de aceptación: noviembre 23 de 2010

Abstract

Enterprise architecture is considered a tool that can identify the main components of the information systems in an organization. It also helps to plan how these components can work together to reach the business goals defined by the organization's management.

Key words: Enterprise architecture, TOGAF, DoDAF Zachman Framework, Arquimate, ARMOR, BMM, KAOS, MDA.

---

* Ingeniero Electrónico, Ingeniero de Sistemas, Especialista en Alta Gerencia, Magister en Ciencias de la Información y las Comunicaciones, Magister(c) en Gestión de Organizaciones. Docente Universidad Distrital Francisco José de Caldas. Correo electrónico: hectorarturo@yahoo.com

Introduction

This document presents a review about the topics involved in the concept of enterprise architecture, pointing out its benefits and problems, and making a description of some frameworks and languages related to it.

Enterprise Architecture

An enterprise architecture (EA) could be considered a description of an enterprise's components and the relationships between them. EA identifies the main components of the information systems inside an organization looking for the different ways to get those components, like: staff, business processes, technology, business information, financial elements, etc., working together, reaching the business goals [1].

EA also describes the necessary terminology and makes a guide about the requirements, design and scalability of the information systems of the organization. An EA project must focus the attention in enterprise goals, organization structure, business processes and business information, actors of the system and its organizational behaviors, current software applications working in the organization, etc.

An EA is built to provide an integral vision of the relationship between business and technological elements, giving the context based on models in which the organization can get different situations to be used as a support to the decision making process [2].

The person who has the responsibility of developing and deploying the enterprise architecture is named the enterprise architect. This person also provides conclusions about the architecture implemented and offers some indicators for identifying opportunities or strategies to improve the architecture developed in an effort to perform the process more effectively and efficiently.

Otherwise, there are three areas that generate critical problems in the process of enterprise architecting [1]. Those areas are:

1. Modeling. It integrates all different concepts from diverse enterprise domains to show how the concerns of the stakeholders are considered within the architecture. These models give bases for business and IT analysis. The complexity presented in an EA modeling is due to a variety of concerns from different stakeholders, with different skills and several perspectives [2]. It is an essential activity to make a description about an EA. Modeling allows visualizing the entire EA, its evolution, and its impact on the existing architecture modeling can also be used to describe the control and data flow evolved in the architecture to the stakeholders and to manage the performance on analyses results. For all previous reasons, the process of selection of a framework and a model usually is a difficult and critical task because an EA must be a strategy of the organization. It implies that once the elements related in the EA project have been chosen, it is very hard to make a change.
2. Management. It is an essential activity for developing and deploying an EA. Bigger organizations are continually updating, upgrading and renovating their information systems generating multiple tasks that concern different schedules. It creates situations that should be solved for the EA through activities like coordination of schedules to guarantee interdependencies between the systems related, solving constraints presented on the

different systems and ensuring interoperability at the syntactic and semantic interactions between information systems.

3. Maintenance. It is necessary to an EA to preserve the operational consistency while the organization continues to evolve the architecture. New subsystems or enhanced deployments should not impact daily operations. It implies that the changes required in the evolution process of the architecture must be carefully executed.

TOGAF

The Open Group Architecture Framework (TOGAF) is a framework that provides an approach for the design, planning, deploying and government of enterprise architecture. Four levels or dimensions usually are modeled in this architecture. These levels are: business, technology, data and applications.

Business level defines the business strategy, governance, organization, and key business processes. Technology level describes the logical software and hardware capabilities that are required to support the deployment of business, data, and application services including information technology infrastructure, middleware, networks, communications, processing, standards, etc. Data level describes the structure of an organization's logical and physical data and data management resources. Application level provides a blueprint for the individual application systems to be deployed, their interactions, and their relationships to the core business process of the organization [3].

Also, TOGAF relies ona set of basic architectures that are used to facilitate the definition of the current and future state of the architecture.

TOGAF defines five categories of stakeholders: end-users organization, project organization, system operations and externals and stakeholders with corporate functions [2].

The Open Group Architecture Framework Architecture Development Method (TOGAF ADM) is a prescriptive, step-by-step instruction guide for an architect. It is presented in a series of phases that guide the architect or architecture team through the architecting lifecycle of system development [4]. It provides a tested and repeatable process for developing architectures. It also includes an architecture framework, developing architecture content, transitioning and governing the realization of architectures.

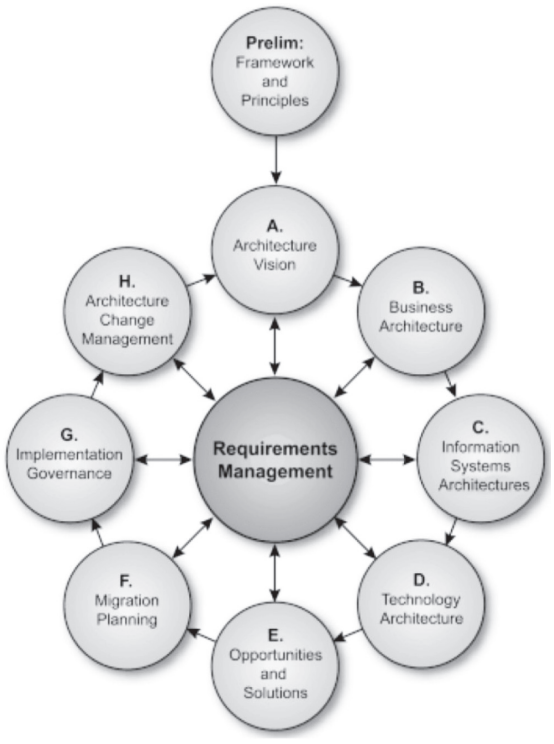
The first seven releases of TOGAF ADM (1995-2001) were focused on providing technical architecture guidance. The 2002 release of TOGAF 8.0 extended this earlier technical focus with elements of business, data, and applications architectures.

Activities included in ADM have an iterative cycle of continuous architecture definition searching business goals and opportunities. ADM contains the following phases [4]:

- Preliminary Phase. Describes the preparation and initiation activities required to arrange the business directive for new enterprise architecture.
- Phase A: Architecture Vision describes the initial phase of an architecture development cycle. It includes information about defining the scope, identifying the stakeholders, creating the Architecture Vision, and obtaining approvals.
- Phase B: Business Architecture describes the development of Business Architecture to support an agreed Architecture Vision.

- Phase C: Information Systems Architectures describe the development of Information Systems Architectures for an architecture project, including the development of Data and Application Architectures.
- Phase D: Technology Architecture describes the development of the Technology Architecture for an architecture project.
- Phase E: Opportunities and Solutions suggest initial implementation planning and the identification of delivery vehicles for the architecture defined in the previous phases.
- Phase F: Migration Planning addresses the formulation of a set of detailed sequences of transition architectures with a supporting Implementation and Migration Plan.

Fig. 1: TOGAF's Architecture Development Method (ADM) [4]



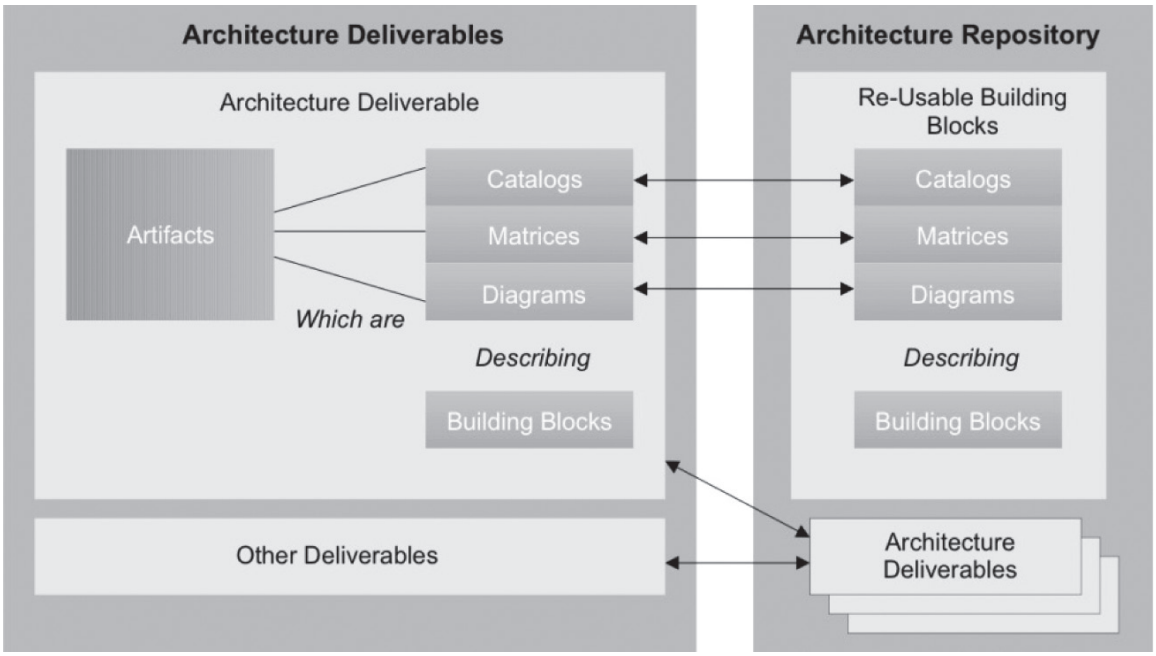
- Phase G: Implementation Governance provides an architectural oversight of the implementation.
- Phase H: Architecture Change Management establishes procedures for managing changes to the new architecture.
- Requirements Management examines the process of managing architecture requirements throughout the ADM.
-

Architects who deploy with TOGAF ADM produce a number of results such as process flows, architectural requirements, project plans, project compliance assessments, etc. The TOGAF Architecture Content Framework provides a structural model for architectural content that allows major work products to be consistently defined, structured, and presented.

The Architecture Content Framework uses three categories to describe the type of architectural work product within the context of use. The products are the following [4]:

- Deliverable is a work product that is contractually specified which the architect presents reviewed for the stakeholders. Deliverables represent the output of projects and those are typically to be archived after completion of a project, or transitioned into an Architecture Repository as a reference model or standard.
- Artifact is a more granular architectural work product that describes architecture from a specific viewpoint. It can include a network diagram, a server specification, a use-case specification, a list of architectural requirements, and a business interaction matrix. Artifacts are classified as catalogs representing lists of things, matrices that can provide relationships between things, and diagrams that represent pictures of things. An architectural deliverable may contain many.

Fig. 2: Relationships between Deliverables, Artifacts, and Building Blocks [3]



- Building block represents a component of business, information technology, or architectural capability that can be combined with other building blocks to deliver architectures and solutions. These building blocks usually may be re-usable components. Building blocks can be defined at various levels of detail, depending on what stage of architecture development has been reached.

DoDAF

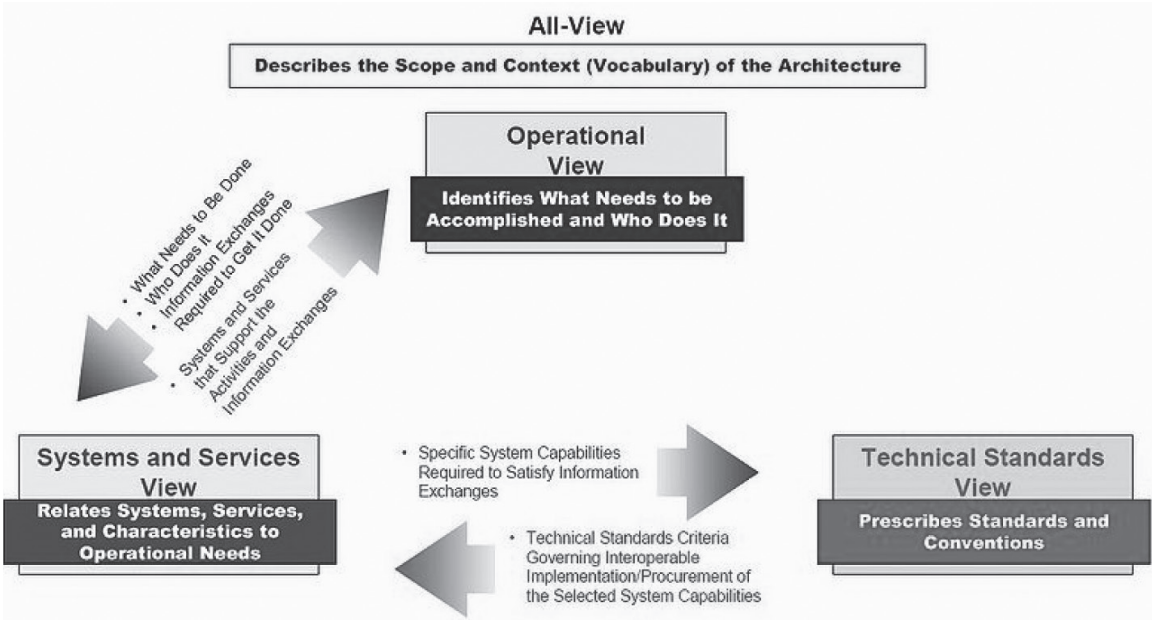
The Department of Defense Architecture Framework (DoDAF) is a reference model that has the goal to organize the enterprise architecture (EA) and systems architecture into complementary and consistent viewpoints.

DoDAF is focused in an architecture description where the architecture model consists of several sub models known as products and reflecting the architecture from multiple viewpoints [4].

DoDAF defines a set of products that have concrete functions like visualizing, understanding, and assimilating the complexities of an architecture description through graphic, tabular, or textual means [5]. Products are organized under four views:

- All View (AV). These products provide overarching descriptions of the entire architecture and define its scope and context.
- Operational View (OV). These products provide descriptions of tasks, activities, operational elements, and information exchanges required to accomplish DoD missions through textual and graphical representations of operational nodes and elements.
- Systems and Services View (SV). These products compound a set of graphical

Fig. 3: DoDAF V1.5 Views [5]



and textual elements that describe systems and services and interconnections providing support to DoD functions.

- Technical Standards View (TV). These products define technical standards, implementation conventions, business rules, and criteria that govern the architecture.

Zachman Framework

Zachman Framework is an Enterprise Architecture framework which provides a structured way of viewing and defining an enterprise. It consists of a classification matrix with two dimensions, based on the intersection of the following six communication questions: What, Where, When, Why, Who and How, with six rows according to transformations [6]. The Zachman Framework is considered taxonomy for organizing architectural artifacts.

This framework depicts different perspectives that are being represented in the engineering process. The defined perspectives are as follows [7]:

- Scope (Contextual) Perspective or Planner's View. It is defined in row number one. This perspective makes a reference about the inner and outer limits, the list of relevant descriptive representations. It is equivalent to a summary for a planner who wants an overview about the system. With this information, is possible to estimate what it would cost, and how it would relate to the general environment.
- Business Model (Conceptual) Perspective or Owner's View. It is defined in row number two. It is focused on the owner that can be the customer or user of the end product. These descriptive representations reflect what the owners are going to do with the end product, or how

Fig. 4: Zachman Framework Views [6]

	Why	How	What	Who	Where	When
Contextual	Goal List	Process List	Material List	Organizational Unit & Role List	Geographical Locations List	Event List
Conceptual	Goal Relationship	Process Model	Entity Relationship Model	Organizational Unit & Role Rel. Model	Locations Model	Event Model
Logical	Rules Diagram	Process Diagram	Data Model Diagram	Role relationship Diagram	Locations Diagram	Event Diagram
Physical	Rules Specification	Process Function Specification	Data Entity Specification	Role Specification	Location Specification	Event Specification
Detailed	Rules Details	Process Details	Data Details	Role Details	Location details	Event Details

they will use it once they have it in their possession.

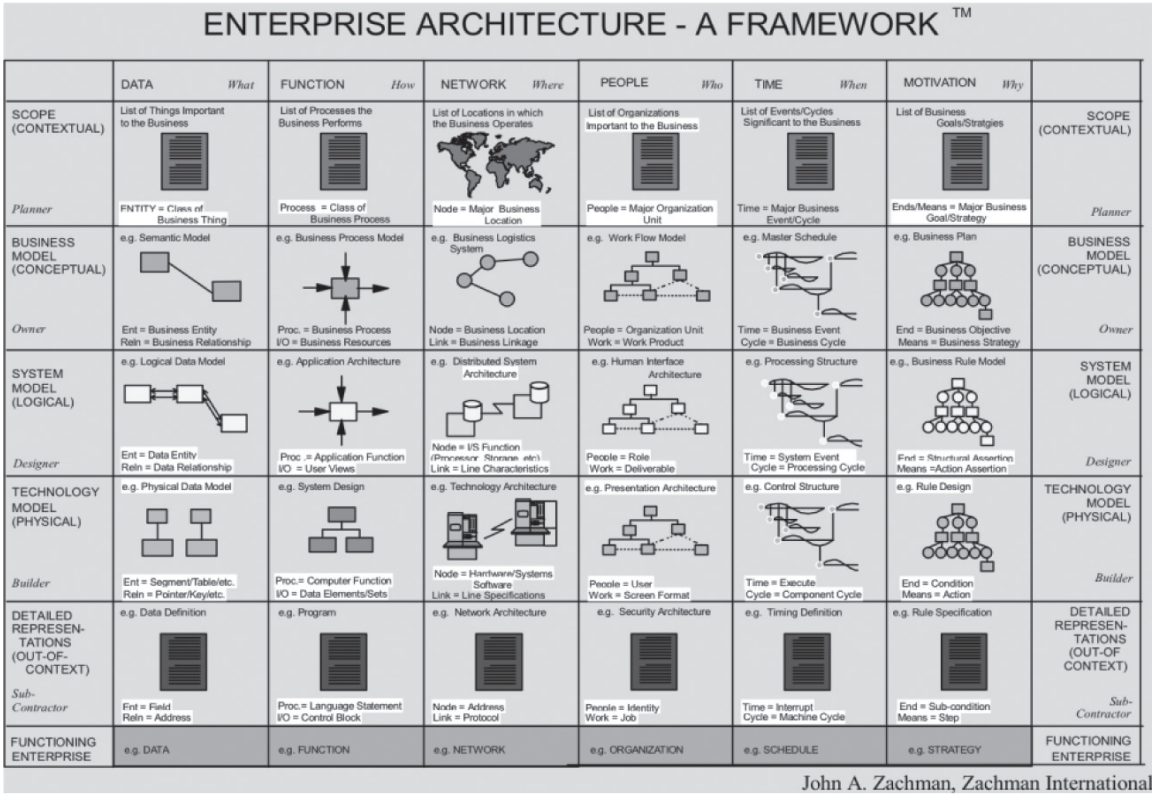
- System Model (Logical) Perspective or Designer's View. It is defined in row number three. These descriptive representations reflect the laws of nature, the system, or logical constraints for the design of the product. This is the logical view of the end product.
- Technology Model (Physical) Perspective or Builder's View. It is defined in row number four. These descriptive representations reflect the physical constraints of applying the technology in the construction of the product.
- Detailed Representation (Out of Context) Perspective or Sub-Contractor's View. It is defined in row number five. It is related to the transformation from the design of the product to the end product. For enterprises, these are the product specifications relating the technology constraints to the vendor products in which the technology constraints are materialized.

ArchiMate

ArchiMate is an enterprise architecture modeling language to support the description, analysis and visualization of architecture. It presents a set of concepts and relationships between architecture and domains, offering a simple structure for describing the contents of the domains represented by the model. ArchiMate is one of the open standards hosted by the Open Group and is based on the IEEE 1471 standard.

The language consists of active structure elements, behavioral elements, and passive structure elements [8]. The active structure elements are the business actors, application components, and devices that display actual behavior. Also, these concepts are assigned to behavioral concepts, to show who or what performs the behavior. The passive structure elements are the objects on which the behavior is performed [8]. These three aspects have been inspired by natural language, where

Fig. 5: Zachman Framework [7]



a sentence has a subject referring an active structure, a verb referring a behavior, and an object referring a passive structure.

The ArchiMate language defines three main layers as follows [8]:

- The Business Layer. It offers products and services to external customers, which are realized in the organization by business processes performed by business actors.
- The Application Layer. It supports the business layer with application services that are realized by software applications.
- The Technology Layer. It offers infrastructure services like processing, storage, and communication services, needed to run applications, realized by computer and communication hardware and software.

In the Architectural Framework is possible to identify a number of other important aspects, some of which may cross several conceptual domains like goals, security, governance, costs, performance, timing, planning, and evolution.

The aspects may be added to the models by means of additional concepts, relationships, or attributes. Also, it may be useful to add concepts or attributes related to the design process. Examples of such concepts or attributes are requirements and design decisions.

ARMOR

ARchitectural Modeling Of Requirements (ARMOR), places a tool to make a requirements modeling in the area of the Enterprise Architect.

Requirements modeling is an important activity in the process of designing and managing enterprise architectures and helps to understand structure and analyze the way business requirements are related to information technologies requirements. Then, here the concept of goal is a goal-oriented requirement modeling used to define some effect and behavior of a system. Otherwise, the goal must be related to more abstract goals referring a business goal that define why this goal is needed, and is also related to more concrete goals that define how the goal can be realized [9].

According to facilitate the acceptance and usability of ARMOR, the conceptual model and concrete syntax should be aligned with other existing languages like Business Motivation Model, i* framework and KAOS requirements modeling language [10].

BMM

The Business Rules Group (BRG) devised the Business Motivation Model (BMM) to provide a scheme and structure for developing, communicating, and managing business plans in an organized way. BMM consists in identifying factors that motivate the establishing of business plans, it identifies and defines the elements of business plans and indicates how all these factors and elements are interrelated [11].

The central notion of the BMM is motivation. An enterprise should not only define in its business plan what approach it follows for its business activities, but also why it follows this approach and what results it wants to achieve. BMM contains the following three important parts [10]:

- Ends, which describe the aspirations of the enterprise, like what the enterprise wants to accomplish.
- Means, which describe the action plans of the enterprise to achieve the ends, and the capabilities that can be exploited for this purpose.
- Influencers, which describe the assessment of the elements that may influence the operation of the enterprise, and thus influence its ends and means.

Fig. 6: Zachman Framework [7]

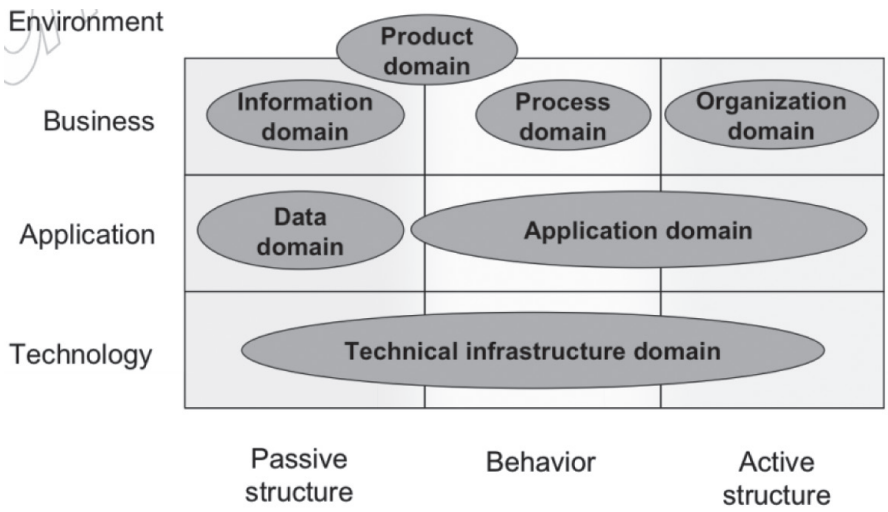
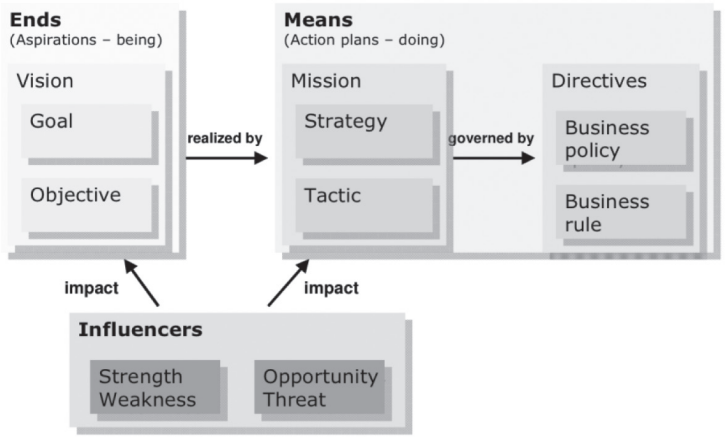


Fig. 7: BMM Overview [10]



An End is something the business seeks to accomplish. It does not include any indication of how it will be achieved. It is useful to document who defined the End and at what point in time [11].

Vision is an image of what the organization wants to be or become. It usually encompasses the entire organization. It describes the future state of the enterprise. A Vision is often composite, rather than focused toward one particular aspect of the business problem. It is made upof goals and objectives.

A Mean represents any device, capability, regime, technique, restriction, agency, instrument, or method that may be called upon, activated, or enforced to achieve Ends [11]. It is useful to document who established the Means and at what point in time.

Means contain a Mission and a Directive. Mission indicates a correspondingly long-term approach, one that is focused on achieving the Vision. Directive is related to governance or guidance.

An Influencer can be anything that has the capability to produce an effect without force or direct exercise of command. The Influencers specifically of concern to business plans are those that can impact the enterprise in its employment of Means or achievement of its Ends.

i* Framework

The i* framework is focused on concepts for modeling and analysis during the early requirements phase. It emphasizes the "whys" that underlie system requirements, rather than specifying "what" the system should do [10].

The i* framework has been developed to model and reason about organizational environments and their information systems.

Two types of models are distinguished in the i* framework: the Strategic Dependency (SD) model and the Strategic Rationale (SR) model.

A Strategic Dependency model describes the dependencies among actors in an organizational context. A dependency model makes an agreement between two actors, where one actor depends on another to obtain a goal, perform a task or deliver a resource. A dependency may involve a soft goal, which

represents a vaguely defined goal with no clear criteria for its fulfillment.

The Strategic Rationale model describes stakeholder interests and concerns, and how they can be addressed by various configurations of systems and environments. This model adds more detail to the SD model by looking actors to model internal intentional relationships. Intentional elements like goals, tasks, resources, and soft goals, appear as external dependencies and as internal elements. Means-end relations and task decompositions can link intentional elements.

The i* framework allows various types and levels of analysis, for example, to assess the ability, workability, viability, and believability of goals and tasks.

KAOS

KAOS is a methodology for requirements engineering enabling analysts to build requirements models and to derive requirements documents from KAOS models [12].

The first key idea behind KAOS is to build a model for the requirements, that is, for describing the problem to be solved and the constraints that must be fulfilled by any solution provider.

KAOS has been designed for the following purposes [12]:

- To fit problem descriptions by allowing to define and manipulate concepts relevant to the them.
- To improve the problem analysis process by providing a systematic approach for discovering and structuring requirements.
- To clarify the responsibilities of all the project stakeholders.
- To let the stakeholders communicate easily and efficiently about the requirements.

With KAOS, the analysts' discovery of the new system goals is made by interviewing current and future users and by analyzing the existing systems, reading the available technical documents. KAOS enables the analysts to structure the collected goals. Each goal in the model is typically justified by at least another goal that explains why the goal was introduced in the model. Also, each goal is refined as a collection of sub goals describing how the refined goal can be reached.

MDA

Model Driven Architecture (MDA) is an approach to system development using models to direct the course of understanding, design, construction, deployment, operation, maintenance, and modification [13].

MDA defines an approach to information technology system specification that separates the specification of system functionality from the specification of the implementation of that functionality on a specific technology platform.

In MDA, a model is a representation of a part of the function, structure and/or behavior of a system. A specification is formal when it is based on a language that has a well-defined form, meaning, equivalent to the syntax and semantic, and possible rules of analysis, inference, or proof for its constructs. The syntax may be graphical or textual. The semantics might be defined, more or less formally, in terms of things observed in the world being described, or by translating higher-level language constructs into other constructs that have a well-defined meaning [14].

In MDA, the concept abstraction is used in the sense defined in the Reference Model of Open Distributed Processing (RM-ODP). It is useful to characterize models in terms of the abstraction criteria that were used to determine what is included in the model. A model that is based on specific abstraction criteria is often referred to as a model from the viewpoint defined by those criteria, or in short, as a view of the system.

In MDA, the concept platform is used to refer to technological and engineering details that are irrelevant to the fundamental functionality of a software component.

In MDA, UML models represent the design for an application. The models used are: PIM and PSM. Both models are described with meta-models that are expressed with UML, MOF (Meta-Object Facility) or other languages. Using UML models within the procedure model could be declared as using techniques. Another technique in the MDA approach is the mapping or transformation between the phases of the procedure model. The development process in MDA consists of four steps as follows: creating a computation independent model (CIM), made up by business analysts to describe the business, creating a platform independent model (PIM), creating a platform specific model (PSM) and generating the application. These four steps build the procedure model for MDA development projects [13].

REFERENCES

S. Kaisler, F Armour, M Valivullah. Enterprise Architecting: Critical Problems.

C Peña, J Villalobos. An MDE approach to design enterprise architecture viewpoints.

TOGAF Version 9. The Open Group Architecture Framework (TOGAF).

F Dandashi, R Siegers, J Jones, T Blevins, The Open Group Architecture Framework (TOGAF) and the US Department of Defense Architecture Framework (DoDAF).

DoD Architecture Framework Version 1.5.

The Zachman Framework: The Official Concise Definition. Zachman International. 20.

J Zachman. The Zachman Framework For Enterprise Architecture: Primer for Enterprise Engineering and Manufacturing.

ArchiMate 1.0 Specification. The Open Group.

D Quartel, H Jonkers, W Engelsman. A Requirements Modelling Language for Enterprise Architecture.

W Engelsmanb, D Quartela, H Jonkers, M van Sinderenb. Extending enterprise architecture modelling with business goals and requirements.

The Business Motivation Model. Business Governance in a Volatile World. The Business Rules Group.

A KAOS Tutorial.

S Leist, G Zellner. Evaluation of Current Architecture Frameworks.

Model Driven Architecture (MDA). Document number ormsc/2001-07-01. Architecture Board ORMSC