International Journal of
Advances in Scientific Research and Engineering (ijasre)

E-ISSN : 2454-8006

DOI: 10.31695/IJASRE.2019.33145

Volume 5, Issue 4
April - 2019

Enterprise Architecture Models:
Description of Integrated Components for Validation

Joe Essien

School of Computing and Technology,

University of West London, St Mary's Road, Ealing

London, U.K

ABSTRACT

Enterprise Architecture (EA) has been defined as the organization of a system embodied in its components, relationships to each other, environment, the principle guiding its design and evolution [1]. Thus an important characteristic of EA is to provide a holistic view of the enterprise visualizing the relevant aspects of the business for specific stakeholders. However, one of the many concerns of this interest has been how to deal with the complex challenges of implementing the models with the ability to validate its integrated components to ensure conformity with individual stakeholder's motivation. To achieve this, methodologies that describe components in relation to their behavioral attributes, impact on other elements in the domain and their dependencies have been postulated. Albeit, studies show that these taxonomies do not adequately address this requirement [2]. This article analyzes the EA concepts of ArchiMate, focusing on the business and application layers with the objective to extend motivation with tests specifications using the model-driven approach thus offer descriptive semantics for validation. The paper contributes to a better understanding on how EA models can be validated thus improve alignment with the business vision and strategy. Student Internship Program case study is used to exemplify this hypothesis.

Keywords: *Enterprise Architecture, Business Processes, Motivation, Validation, Modeling.*

1. INTRODUCTION

Enterprise Architecture has been defined as consisting of coherent principles, methods, and models used in the design and realisation of organisational structure, business processes, information systems and infrastructure [3]. Good enterprise architecture provides the insight needed to balance requirements and facilitates that translation from corporate strategy to daily operations [2]. Through the alignment of business functions and IT systems, a form of harmonization between the current state of a business (as-is) and a desired state of a business (to-be) is achieved [4]. Venkatraman, [4] in their later review identified eight other perspectives in which EA alignment can be achieved. In addition to these eight perspectives, four other fusion perspectives are described, formed from the combination of two of the individual perspectives [5]. In all these efforts and many others with respect to definitions of EA, perspective, harmonization and alignments, the issues of validation are completely ignored or at best remain rudimentary. The positions do not consider the behavioral attributes of the model's components as a process that should undergo test itself. In many organizations, EA patterns exist that encapsulate business concerns such as maintenance, upgrades, procurement, integration, acquisition and mergers, compliance in a regulatory environment and strategic planning [6], but literal analysis of these patterns shows many disparate architectures, understood by each stakeholder from different perspectives. The connections and dependencies that exist among these different views can be extremely complex in some cases [7]. To tackle these phenomena as a prerequisite to determining attributes behavior, some authors have proposed a distinction between aspects of EA visualization. One option is taking the business strategy of an enterprise as the starting point, and then deriving its IT infrastructure either via an IT strategy or through the organizational infrastructure [4], also referred to as (top-down) strategy to execution. Another way conversely, is focusing on IT as an enabler and starting from the IT strategy to derive the organizational infrastructure via a business strategy or based on the IT infrastructure, referred to as (bottom-up) execution to strategy. Though these distinctions appear rational as a means to an end, most large enterprise elect that business vision and strategy drives EA and not the reverse. For this reason, this study is construed on the first option and develops conceptual frameworks that use common business logic and models to define annotations of EA components to facilitate validation. The document describes the process of designing test specifications for an enterprise architecture model, uses the Student Internship program (SIP) as a case study to show a pragmatic application of the concept and how the specification can be extended to the model itself. EA is decomposed to extract data specifications for building the validation scenario, spanning the business and application abstractions and aggregates requirements into autonomous business behavior from the perspective of stakeholders and goals.

2. PROBLEM STATEMENT

The multi-dimensional interests and unstructured principles that tagged EA from inception led to use of heterogeneous set of approaches and modelling languages [8]. Most EA practitioners recognize four facets of EA and agree that it comprises of the business, application, information, and technology perspectives [9]. With the augmentation and advancement on these various perspectives, methods, approaches and principles over a period of four decades, one of the biggest issues facing enterprise architects today is that none of these single practices is capable of satisfying all necessary aspects of the enterprise identified collectively [10]. Attempts to mix and match rather, has resulted in EAs with inconsistent semantics and weak ontology. Therefore issues regarding systematic validation techniques become inconsequential as architects do not need to stringently scrutinize models to see the huge gaps in the composition of structural layers, artifacts types and dependencies. The implication of this is that, with the introduction of advance modelling techniques to automate EA practices, need to test and validate those models suddenly emerge as major challenge. Early frameworks such as the popular ZF, GEAF, FEAF, TOGAF, SEAM, and OMG did not create extendable models that support validation [3], [9], [11]. Consequently, attempts to introduce them at a later stage through improved versions and sometimes outright extension such as TOGAF with ArchiMate leave inconsistencies, omissions and gaps between the various layers of abstraction. Despite this, it is agreed that no Enterprise Architecture framework can completely view the enterprise in its entirety as comprising of business objectives, business processes, roles, organizational structures, organizational behaviors, information, software applications, computer systems and the relationships between these various entities [12]. Though efforts still continue to be made towards standardization [13], [14, many frameworks are specific in scope and purpose and apply to specific domains, generally weighted towards planning and business process analysis without commensurate emphasis on validation and change management.

3. VIEWS AND VISUALIZATION

To achieve quality in enterprise architecture, there is need to bring together information from unrelated domains and adopt an approach that is understood by every stakeholder [6]. A Stakeholder can be an individual, team or external entity that has interests relative to the system. It is argued that since stakeholders are influenced by their particular concerns [13]; stakeholders require specific views of architecture that focus on their concerns exclusively. In this context, a view is specified by means of a viewpoint and describes how particular concerns of the stakeholders are constructed in relation with other elements in the EA. It has been contested that stakeholders alone would not provide substance to a view. Clark et al [15] suggested that models need to be goal focused and elements of EA should be goal-driven motivated by constraints and drivers. But goals without substantiation can be abstruse. Several authors in their publications have defined goals in EA as Business-IT alignment, Governance, Standardization, Cost reduction, Consolidation, Agility, Risk management, Regulatory compliance, business continuity [7],[8],[9]. Goals in this study are intrinsic, can be decomposed and refer to added value and objectives of the stakeholder. We adopt an analogous approach that the stakeholder’s concerns are assessed to derive goals and that requirements can satisfy goals.

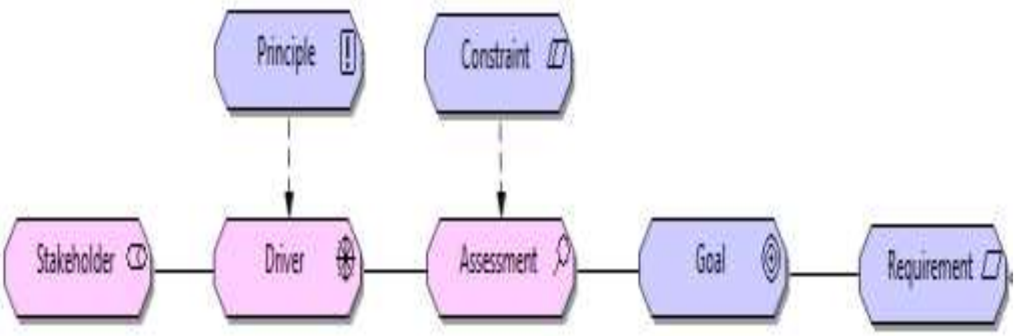

Figure 1: Relationship of Stakeholder and Goal

Goals are used with viewpoints to specify business behavior and to derive the artifact for extrapolation required for the design of the validation scenario.

4. MODELLING LANGUAGES AND VALIDATION

Modelling Language ML is a high level abstraction language, aimed at representing structures, characteristics and properties at early stage of design [12]. Over the decade, there has been proliferation of ML as means of presenting visual images of design concepts. The Unified Modeling Language (UML), one of such adopted by Object Management Group (OMG) is a standardized, general-purpose modeling language in the field of software engineering and includes a set of graphic notation techniques to create visual models of object-oriented software-intensive systems [14]. Though it combines techniques from data modeling (entity relationship diagrams), business modeling (work flows) and object modeling, it lacks the versatility that can visualize the entire enterprise as defined by IEEE, Lankhorst [19], DODAF [20]. UML is focused on definition of system structure and behavior and has no built-in testing constructs [16]. The UML Test profile currently proposed is at a much lower level of abstract based on Testing and Test Control Notation Version 3 (TTCN3) and JUNIT than required in business behavior validation at EA higher abstraction.

DOI: 10.31695/IJASRE.2019.33145

The Zachman framework (ZF) as an EA approach is a normalized six by six classification schema for organizing descriptive representations of an enterprise [17]. The rows represent distinct stakeholder perspectives of an enterprise, while the columns describe different areas of interest within those perspectives. The Zachman framework is simply a framework rather than a process, a method, a notation or a tool. Consequently the framework is rigid as rows and columns cannot be added or omitted to allow validation or testing. The Zachman framework is useful more in an enterprise as a general assessment tool for organizing a complete and holistic set of existing architecture descriptions and artifact sets; and to identify gaps in information and focus development efforts to fill the gaps. In ZF, there is no correct modelling tool for any particular cell. Any modelling tool may be used to depict the structural components of the cell e.g., UML diagrams, analytic equations, functional flow block diagrams, block diagrams of linear systems theory, transfer functions, state-space models, differential equations, object-oriented models, etc [17], 2006). Each entity in the cell may use any representation for functions, processes, events, objects, data and interfaces. This makes it very difficult for components within ZF cells to share homogeneous annotations, semantics and relationship thus application of a standardized validation method on its framework is impracticable.

The Extended Enterprise modelling Language (EEML) was developed as a comprehensive and generic framework for evaluating models, called SEQUAL [18]. SEmiotic QUALity (SEQUAL) framework is systems modelling top-down reference model for evaluating the quality of models. It distinguishes between goals and means by separating the expected result from procedures needed to achievement it, through a process based on linguistic and semiotic on real world view with participation of the stakeholders. The core of the framework includes the discussion on syntax, semantics, and pragmatics parallel to the use of terms in the semiotic [18]. Though EEML SEQUAL has few desirables, the associated limitations are enormous as it is completely domain specific. The representation of business rules is dependent on prevalent use and implementation. Maintenance and knowledge enhancement is key requirement to its usability. EEML is under further developed in the EU projects Unified Enterprise ML (UEML) to validate and disseminate a set of core language for its support as a basis for interoperability within a smart organization [14].

Another popular means of validating the EA is the use of maturity matrix commonly referred to as Dynamic Architecture Maturity Matrix (DYA AMM). DYA AMM is used as an instrument to assess the level of Enterprise Architecture Maturity (EAM) in organizations and often has many (up to 20) key areas that represent a different dimension within EAM. The DYA AMM assessment method makes it possible to assess organizations on an overall maturity level as well as a specific level. The information for assessments is gathered through a survey questions that relate to one of the identified key areas [5]. However, this method which is purely qualitative has limitations as lack of a comprehensive approach to data gathering can affect judgement. The researcher's presence during data gathering, which is often unavoidable, can affect the subjects' responses. Issues of ethics, anonymity and confidentiality can present problems when presenting findings. Among other concerns, the questionnaires are sometimes not well understood by the respondents and the CIO bias undoubtedly may influence the outcome.

The Open Group Architecture Framework (TOGAF) [13] is one of the most popular frameworks in EA. To provide a uniform representation for diagrams that describe enterprise architectures, the ArchiMate EAML is developed to support TOGAF ADM and to offer an integrated architectural approach that describes and visualizes the different architecture domains, their underlying relations and dependencies. However, assessment methodology is not integrated with ArchiMate Core or its extension.
Maturity assessment discussed earlier is also deployed in TOGAF ArchiMate to identify the level of compliance between business vision and business capabilities. Our rationale for adopting TOGAF and ArchiMate is on this basis to fill the gap; coupled with the fact that ArchiMate actually sets the platform for achieving this by offering formal descriptions of components that support reasoning about the structural and behavioral properties of the organization. It provides graphical language for the representation of EA models and enables the introduction of annotations and semantics for validation.

5. MOTIVATION DRIVEN VALIDATION APPROACH

The Motivation Driven Validation Approach (MDVA) validates scenarios of an instance model from a logical model incrementally, across the different views and layers of EA by testing component attributes against goals in the motivation Extension. This method though deploys the same principles of Behavior Driven Development (BDD), differs as it focuses on behavioral specification of the EA artifacts rather than objectives. Three steps are iterated;

- Specification of model validation rules;
- Validation of the rule on the model instance;
- Validation of result with motivational goal;

Validation of artifact is based on the desired behavior with attributes set for the related motivational goal. The MDVA uses structure and behavioral patterns to ensure traceability thus ensuring that the right design decisions are taken at the modeling stages. Not only does the MDVA improve the quality and design of the framework, it also simplifies the modeling process. The validation scenarios for MDVA describe the behavior and attributes of the component to be validated in order to realize set motivation goal. MDVA ensures better conformance to user goals and provides the means for model traceability required for artifact validation. As accepted by many authors, motivational conceptions can be used to model the basis that inspires the design or change of enterprise architecture [11].

DOI: 10.31695/IJASRE.2019.33145

6. MDVA METHODOLOGY

The MDVA consist of both the behavioral and the structural attributes of the EA components. Physical models of business behavior are created as derivative instances with different stakeholder perspectives for validation. Unlike BDD, test basis created are not based on the business behavior itself but on the attributes of the artifacts that constitute the model instance at a high level of abstraction.

6.1 MDVA Design

The MDVA is conceptualized from the ArchiMate Motivation Extension by deploying motivational element across the business and application layers. The methodology iterates correlations of motivational elements over the taxonomy to establish extent and coverage of the business behavior defined. Through the process, gaps and overlapping functionalities are identified allowing the model to be validated.

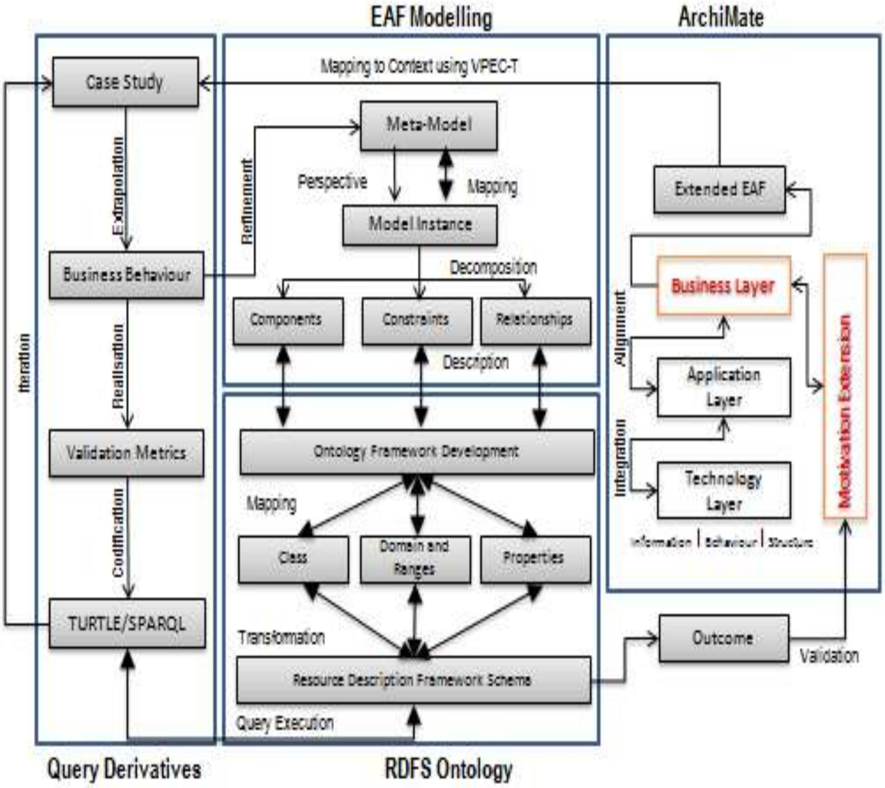

Figure 2: Abstract Syntax of the MDVA schema

Figure 2 shows the MDVA concept proposed in this paper. By iterative refinement of business behavior on the Business Layer, components are extrapolated into views to aggregate viewpoints for a particular test requirement. Constraints are then applied on the components to derive test attributes for the logical model. These components form the test basis for the logical model design. The diagram depicts realization of physical model instances from the conceptual and logical models. These are validation from different views with test scenarios specified from constraints. Goals are part of the motivation extension of ArchiMate and ensure alignment and integration with the core EA. Through an iterative process, these models are revalidated through each test attribute of the artifact, generating traceability from specific view for each stakeholder.

6.2 The Case Study

The case study, grounded on student internship programme (SIP) at an academic institution is used to illustrate the MDVA. A system is required with the aim to implement a program that offer student placement. The objective is to automate the process of matching students with employers and internships, allowing students to manage their CV, search for internship listings, request and apply for internship and store their feedback once the internship has taken place. The system allows employers to manage internship listings, track progress on internship listing and provide feedback on student internships once they have taken place. Administrators create users, search and match student CV with opportunities, forward student CV to employers, track student visits and generate reports on system usage. For this paper, a motivational model is required that can validate models created for student’s viewpoint.

DOI: 10.31695/IJASRE.2019.33145

7. ARCHIMATE MODELING

The ArchiMate language provides a means to handle modeling complexities of modern information-intensive enterprises. For our modelling concepts ArchiMate enterprise modeling language is used with the objective to extend the motivation attributes with tests specifications using the Model Driven Approach. Figure 3 shows the complete goals and aggregation refined for all the viewpoints in the SIP based on the requirements specified in the case study. Here the views of Student, Employer and Career Office are integrated to establish congruency and to ensure that there are no gaps. The student needs to be able to search for internship program and provide feedback; the employer needs to be able to provide the available internship opportunities as well as feedback; Career Office automates the match process as well as generates reports. The overall goal of the enterprise is to be able to guarantee that "Student Start Internship". Constraints are modeled into the design and Goals are realized through Requirements.

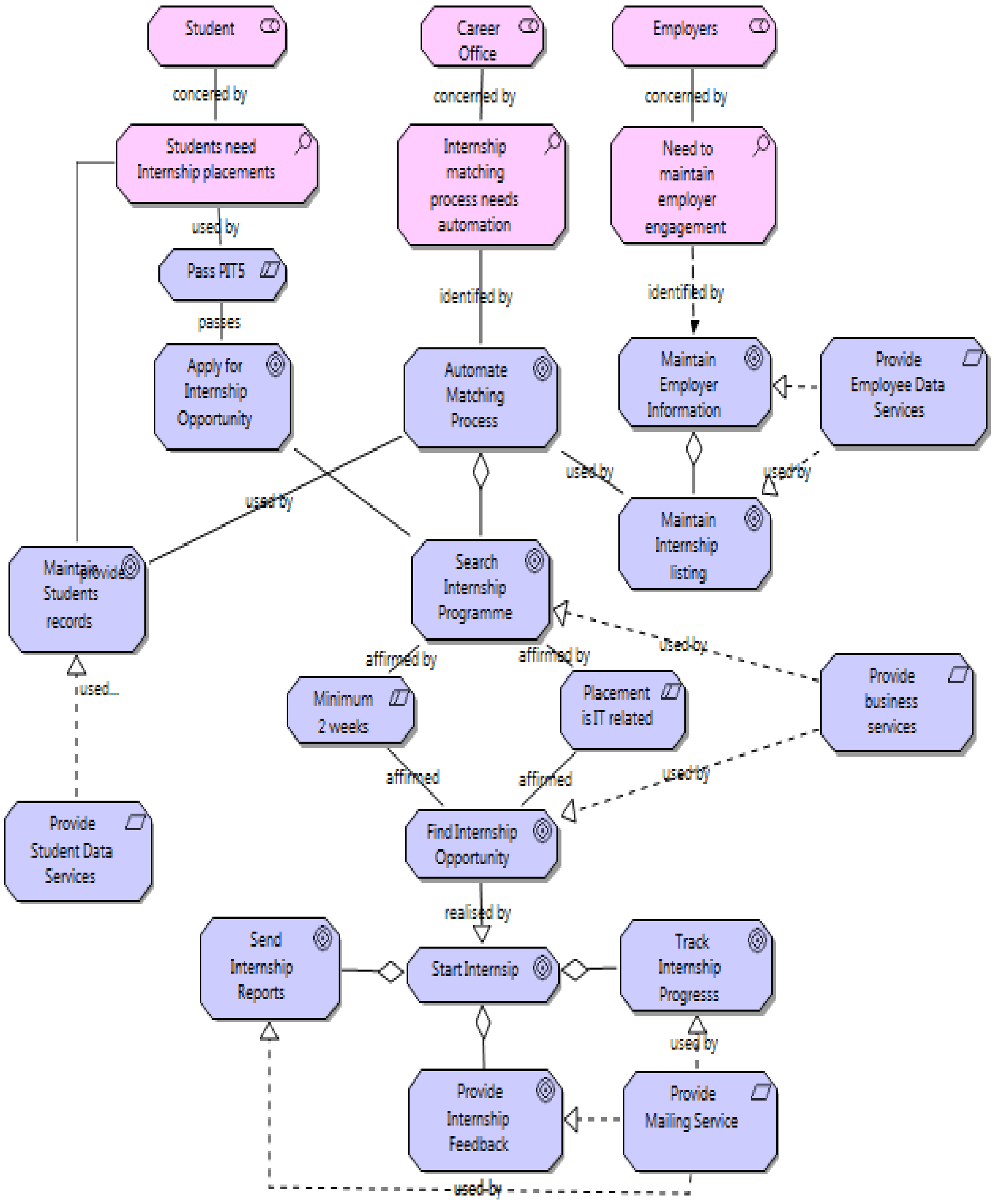
Student
Career Office
Employers
concerned by
concerned by
concerned by
Students need Internship placements
Internship matching process needs automation
Need to maintain employer engagement
used by
Pass PIT5
identified by
identified by
passes
Apply for Internship Opportunity
Automate Matching Process
Maintain Employer Information
Provide Employee Data Services
used by
used by
used by
Maintain Students records
provide
Search Internship Programme
Maintain Internship listing
used...
affirmed by
affirmed by
used by
Minimum 2 weeks
Placement is IT related
Provide business services
affirmed
affirmed
used by
Provide Student Data Services
Find Internship Opportunity
realised by
Send Internship Reports
Start Internsip
Track Internship Progresss
used by
Provide Internship Feedback
Provide Mailing Service
used by

Figure 3: Use Case Goals aggregation

In the following subsections, the preliminary process for implementation of the MDVA is carried out. Models from the motivational Requirements establish constraints and associations with Business Role, Business Function, Business Process and Business objects at the Business Layer. For the purpose of this paper, only the student's abstraction will be used with associated Goals.

7.1 Constraints

Though some of the constraints can also apply to the career office as the career office match students with placement opportunities also, we focus on student's view and present constraint associated with students only to create our validation scenario. Figure 4 illustrates the modelling of constraints that affect the goal "Start Internship" for students.

DOI: 10.31695/IJASRE.2019.33145

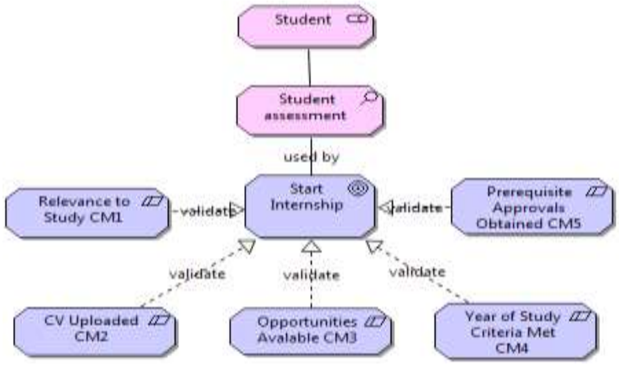

Figure 4: Realization of Constraints within Goals

The Goal “Start Internship” is realized though five Requirements which also present the validation conditions. These are Relevance to study (CM1), Uploaded CV (CM2), Internship Opportunity is available (CM3), student is currently in a year of study permitted for the internship (CM4) and prerequisite approvals obtained (CM5) Figure 4. These conditions form the basis for criteria and scenario descriptions and the tests procedure needed to validate the model.

7.2 Business Role

Business role is used in a structural organizational sense to relate with Business processes or Business functions. Business Role is modeled in the Business Layer though it can be extended with components from both the Application and Technology layers and are assigned primarily to one or more business processes or business functions. The model created in Figure 5 for the case study illustrates the assignment of Business Function to a Business Role. While the Business Function assesses Business Objects, it triggers Events which initiates relevant Business Processes. Application Service is invoked through assess relationship in collaboration with Application Components and Data Objects.

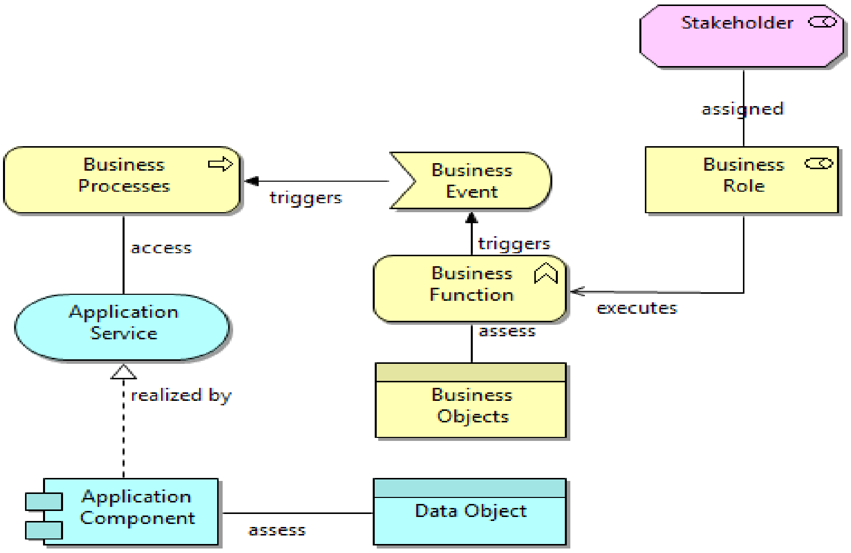

Figure 5: Assignment of Relationships across the Business Application Layers

With reference to our case study, the architecture model can be interpreted from the perspective of the student as follows; the student as the stakeholder is assigned the Business Role “Personal Information Provider” in order to meet the requirement CV upload CM2 and realize the Goal “Start Internship Programme” see Figure 4. The Business Function “Update Details” is executed using predefined Business Object “Data Templates”. On completion of the task, a Business Event “Update Records” is triggered which initiates Business Processes through the Application Service to update the Data Objects.

7.3 Business Object

Business objects are manipulated by behavior in this study. The behaviors under consideration are Business Processes, Business Function and Business Objects. Here Business Process triggers Business Function while Business Object grants access only to associated artifacts.

DOI: 10.31695/IJASRE.2019.33145

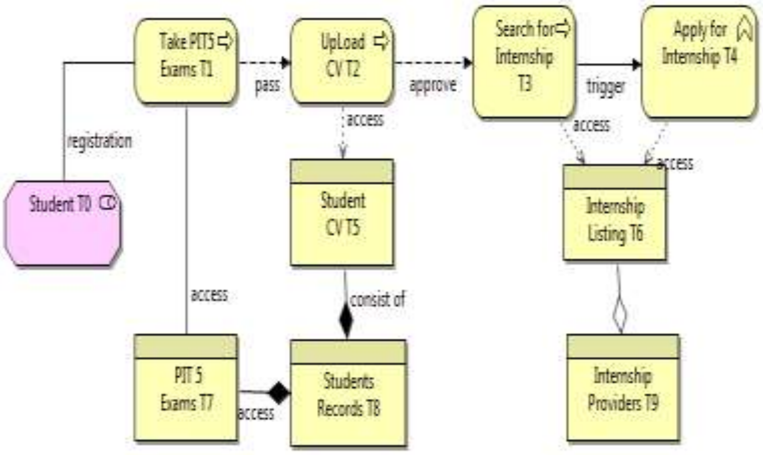

Figure 6: Validation of Business Objects with Business Processes from Student's Perspective

The model in Figure 6 shows an identification of Business Objects associated with the case study from the student's perspective. There is a process chain for PIT5 exams in T7 to Search for Internship only after validation through test conditions defined in the Business Objects T8 and T5. The "Search for Internship" (T3) triggers "Apply for Internship" (T4) but both objects collaboratively access "Internship Listing" T6.

7.4 Business Process

Business Process describes a flow of activities in the model represented in Figure 5 and 6. The Business Processes T1, T2 and T3 trigger the Business Function element T4 (Figure 6) represented as Business Process in Figure 5 and provides access to the Application service. Figure 6 shows aggregation and composite relationship attributes of the passive Business Function access relationship with Business Process. In the case study, the Business Process represents a workflow consisting of smaller processes leading to a Business Function "Apply for Internship".

8 MDVA IMPLEMENTATION APPROACH

Harnessing the techniques described in previous sections, MDVA is grounded on the establishment of concrete test basis, defined at business level scenarios and annotated with constraints from the motivation concepts to support comparison between obtained and expected results. Figure 7 shows the transformation of Figure 5 relative to Figure 6 to include constraints defined in Figure 4.

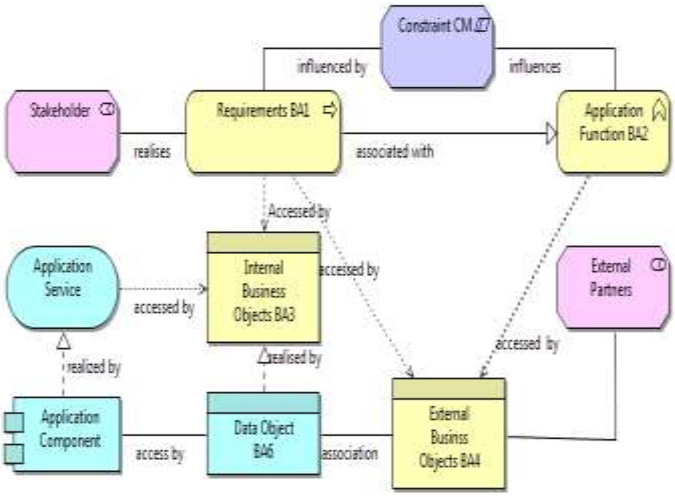

Figure 7: Model transformation of business objective integrated with motivational constraints

In implementation of MDVA, we adopt the first step which is the creation of the conceptual model from the Goals requirements as a perspective to be validated, Fig 2 and Fig 7. Then, the next step transforms the conceptual model into a logical model based on constraint integrated into the taxonomy, specifying artefacts of the model that are to be tested. During this transformation, the test basis is generated explicitly including relations mappings with a traceability of model defined. The third step defines test scenarios with constraints and creates test conditions for validation of the EA artifacts. This is shown in Table 1 where in the model in Figure 4, constraints associated with the Goal "Start Internship" are extrapolated and cross-referenced with the business object model in Figure 6 and the architecture model in Figure 7. The constraints CM1, CM2, CM3, CM4 and CM5 are validated through CM constraints paths in the model transformation defined in Figure 7 and associated with corresponding objects in the actual implementation model. The constraints are applied on the defined artifacts to identify the existence of the object as well as validate stated conditions. Some of the test conditions defined in the goal motivation constructs is exemplified on the table using business readable domain specific mnemonics.

Table 1: Application of Constraints to Model Artifacts.

Artifact map/ Constraints Description	Low level Definition of Test conditions for Artifacts Validation

CM1:on
BA1<T1>,BA3<T8 >
 	
Relevance to Study
	If Object.T1 = pass PIT5
	then result =True
	else object.T1 =false endif
	if Object.T8->notEmpty()
	then result = true
	else result =false endif

CM2:on BA1<T2>,BA3<T5>
 	
CV Uploaded
	If Object.T5 ->notEmpty()
	then result = true
	else result =false endif
	if uploadcv().T2=true
	then object.T2 ->not empty
	result =true endif

CM4: on BA3<T8> Year of Study	1<year of study<3
CM5: on BA1<T3> Requisite approvals	CM1+CM2+CM4 = True

At a higher level of abstraction, this can also be expressed using a BDD notation such as Gherkins for each of the constraints; For Example, CM1:BA1<T1>, BA3<T8 > is validated as;

Given that artifact T1 exist in BA1
And artifact T8 exist in BA3
When Constraint CM1 is parsed in T1 and T8
Then the result shall be True

A simple traceability model to demonstrate this notation usability is shown in Figure 8.

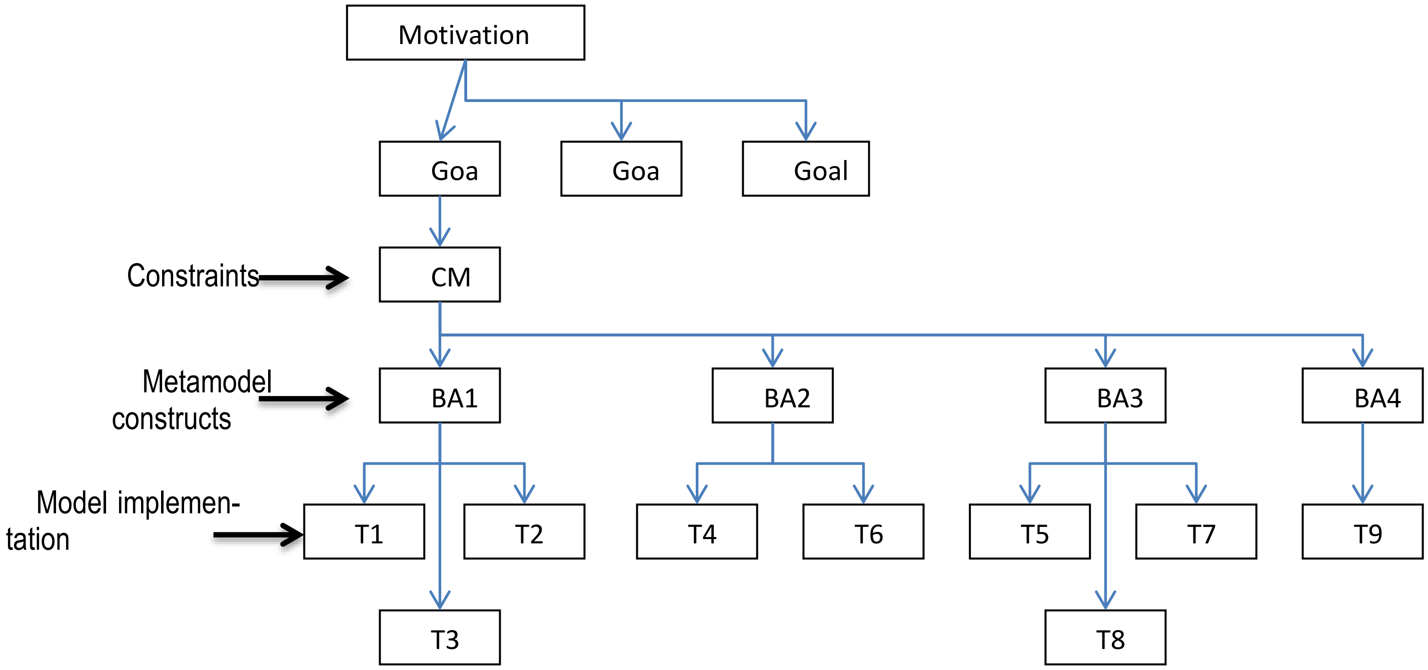

Figure 8: Traceability model for artefact validation

DOI: 10.31695/IJASRE.2019.33145

The MDVA technique addresses the traceability problem by creating relationships between transformed models and artefacts as part of the conversion process, externalizing the relationships among the test-artefact models to allow for comparison with expected outcome.

9. CONCLUSIONS

MDVA presented in this paper is an approach that decomposes business processes and develops constructs for the models to allow validation. Modelling motivational goals involve the conceptualization of different aspects of the enterprise from different viewpoints and levels of abstraction during the life cycle of the architecture. This article includes such conceptualizations derived through modelling and descriptions of models of the business behavior; specifying concepts of intentions in terms of goals, constraints and requirements. The models offer description of integrated components and illustrate the relationships between the various artifacts that constitute the taxonomy, relating business vision, mission and strategy with information systems through modeling extensions of ArchiMate.

Enterprise Architecture and its management have continued to be a topic of ongoing and increasing interest to practitioners. Standardization of concepts (considering disparities in ZF), methodology (as consolidated by TOGAF) would facilitate stabilization and leverage with new innovations to extend EA with validation models, notations and semantics. New technological trends such as cloud computing and big data pose challenge to EA integration. Creation of more EA management roles within enterprise needs to be embraced to allow evolution and provide more information for further research.

REFERENCES

1. IEEE Computer Society. IEEE *Recommended Practice for Architectural Description of Software Intensive Systems.* IEEE Standard 1471-2000.
2. Proper, Henderik, and Marc M. Lankhorst. "Enterprise architecture-towards essential sensemaking." *Enterprise Modelling and Information Systems Architectures (EMISAJ)* 9.1 (2014): 5-21.
3. Fischer, Christian, Robert Winter, and Stephan Aier. "What is an enterprise architecture principle?." *Computer and Information Science 2010*. Springer, Berlin, Heidelberg, 2010. 193-205.
4. Venkatraman, N., and J. Henderson. "Strategic alignment: Leveraging IT for transforming organisations." *IBM Systems Journal* 32.1 (2010).
5. Coleman, Preston, and Raymond Papp. "Strategic alignment: analysis of perspectives." *Proceedings of the 2006 southern association for information systems conference*. 2006.
6. Weston, J., and J. Defee. "Performance Based Enterprise Architecture Planning–A white Paper, 2004." (2013).
7. McGovern, James, Scott W. Ambler, and Michael E. Stevens. *A practical guide to enterprise architecture*. Prentice Hall Professional, 2004.
8. Sessions, Roger. "A comparison of the top four enterprise-architecture methodologies." *Houston: ObjectWatch Inc*(2007).
9. Salmans, Brian, and Leon Allan Kappelman. "The state of EA: progress, not perfection." *The SIM guide to enterprise architecture* (2010): 165-187.
10. Noran, Ovidiu. "An analysis of the Zachman framework for enterprise architecture from the GERAM perspective." *Annual Reviews in Control* 27.2 (2003): 163-183.
11. Urbaczewski, Lise, and Stevan Mrdalj. "A comparison of enterprise architecture frameworks." *Issues in Information Systems* 7.2 (2006): 18-23.
12. Chen, David, Guy Doumeingts, and François Vernadat. "Architectures for enterprise integration and interoperability: Past, present and future." *Computers in industry* 59.7 (2008): 647-659.
13. TOGAF, *The Open Group. ArchiMate Version 2.* http://www.opengroup.org/archimate, Oct, 2018.
14. OMG, http://www.omg.org/. Assessed, September 2018.
15. Clark, Tony, Balbir S. Barn, and Samia Oussena. "Leap: a precise lightweight framework for enterprise architecture." *Proceedings of the 4th India Software Engineering Conference*. ACM, 2011.
16. Baker, Paul, et al. "The UML 2.0 testing profile." *Proc. of the Conf. on Quality Engineering in Software Technology*. 2004.
17. Bahill, A. Terry, Rick Botta, and Jesse Daniels. "The Zachman framework populated with baseball models." *Journal of Enterprise Architecture* 2.4 (2006): 50-68.
18. Krogstie, John. "Using eeml for combined goal and process oriented modeling: A case study." *Proceedings of EMMSAD*. Vol. 8. 2008.
19. Lankhorst, Marc. *Enterprise architecture at work*. Vol. 352. Berlin: Springer, 2009.
20. Hause, Matthew. "The Unified Profile for DoDAF/MODAF (UPDM) enabling systems of systems on many levels." *2010 IEEE International Systems Conference*. IEEE, 2010.

DOI: 10.31695/IJASRE.2019.33145