Formalization of Validation Extension Metamodel for Enterprise Architecture Frameworks

Samia Oussena, Joe Essien and Peter Komisarczuk
School of Computing and Technology, University of West London, St Mary's Road, Ealing, London, U.K.

Keywords: Web Ontology, Enterprise Architecture, Metamodel, Model, Viewpoint, Resource Description Framework Schema, Modelling Language, Archimate, Business Strategy, Validation, Motivation.

Abstract: Formalization of Enterprise Architecture (EA) concepts as a whole is an area which has continued to constitute a major obstacle in understanding the principles that guide its adaptations. Ubiquitous use of terms such as models, meta-models, meta-meta-models, frameworks in the description of EA taxonomies and the relationship between the various artefacts has not been exclusive or cohesive. Consequently variant interpretations of schemas, conflicting methodologies, disparate implementation have ensued. Incongruent simulation of alignment between dynamic business architectures, heterogeneous application systems and validation techniques has been prevalent. The divergent and widespread paradigm of EA domiciliation in practice makes it even more challenging to adopt a generic formalized constructs in which models can be interpreted and verified (Martin et al., 2004). The unavailability of a unified EA modelling language able to describe a wide range of Information Technology domains compounds these challenges leading to exponentiations of EA perspectives. This paper seeks to present a formalization of concepts towards addressing validation concerns of EA through the use of ontologies and queries based on constraints specified in the model's motivation taxonomy. The paper is based on experimental research and grounded on EA taxonomies created using the ArchiMate modelling language and open source web ontology. It delves into the use of semantics triples, Resource Description Framework Schema and relational graphs to map EA taxonomy artefacts into classes and slots using end-to-end conventional formalization approach applicable within heterogeneous EA domains. The paper also expounds on a proposal that postulates implementation of the approach, enables formalized traceability of EA validation and contributes to effective validation of EA through refined taxonomy semantics, mappings and alignment of motivation.

1 INTRODUCTION

Though several publications have referred to the practice of EA and associated terminologies such as framework, model, metamodel, perspective, and viewpoint, research has shown that many concede to its ambiguity. The fact that there is no common understanding of the ideologies behind the concepts is undisputed. Comparative surveys have been carried out to identify possible dimensions of EA based on timelines of relevant literature, author's background, structural dimensions, differentiation between aspects, motivations, contributions and the handling of definitions and terminologies. Depositions from these studies indicate that increasing number of IT practioners and authors use the term EA and its associated phraseology explicitely to expound strategies that are either restrictive in order to demonstrate their domain requisites, or extended to encompass architectural understanding for all forms of EA ramifications (Braun and Winter, 2005). These types of inferences constitute irreconcileable extremities. Often, there is limited significance in relationtionship between background hypothesis and pragmatic requirement. Considering the maturity and the focus of contributions towards EA, most of the approaches postulated are still evolving especially in terms of applicability, making formalization subject to persistent variations. Frameworks and modelling are often surmised by differentiation depending on the proclivity of the practioner. With majority of presumptions being generic, it would seem pertitent that enterprise should evolve techniques for validating the models that drive their business strategy in order to ensure

Oussena S., Essien J. and Komisarczuk P..
Formalization of Validation Extension Metamodel for Enterprise Architecture Frameworks.
DOI: 10.5220/0004950304270434
In *Proceedings of the 16th International Conference on Enterprise Information Systems* (ICEIS-2014), pages 427-434
ISBN: 978-989-758-029-1

that its motivation and goals can be realized by the taxonomy. However, while business views are identified in many EA proposals, business strategy modelling from the perspective of motivation and business drivers are often overlooked (Lankhorst, 2013). Thus IT solutions cannot be traced back to business strategy in a clear and unambiguous manner. Our intended approach to formalize the Validation of Extension Metamodel (VEM) for EA Framework (EAF) aims to establish such process. Formalization of VEM for EAF is the rationalization of known validation strategy with precise semantics enabling its model-level usage to provide strategic awareness of EA and propose a conceptual relationship towards EA.

Inferred from preliminary studies for this work, formalization has not been attainable due to ambiguity and divergence that exist within modelling techniques. The extent of formalization differ depending on the purpose of the design from motivation to direct EA model, maintenance of metamodel or even the abstract meta-metamodel. As such instantiations do not establish meaningful traceability as expressed by their metamodels and frameworks. Consequently, EAF formalization is critical in order to enable transformation of semantics and principles from domain specific constructs to unambiguous descriptions of concepts. The use of ontology is a new dimension introduced to address this phenomenon.

Following this introduction, this paper is organised as follows. Section two presents pungent and concise expositions of the concepts of formalization, focusing on two main categories. The first specifies models, metamodels, framework from the perspective of EA and the second delves into ontology, resource definition framework schema, and correlations as applied to validation. Section three focuses on description of the extended validation elements. Section four rationalizes the methodology adopted using a metamodel construct of ArchiMate (TOG, 2014). Section five presents principles of transformation to ontology. Section six delves into the mapping methodology. Section seven presents the resultant ontology with validation constraints and metrics drawing inferences to query methods, graphs and traceability. Section eight concludes the paper by evaluating the outcomes, the principles of formalization and areas of further research.

2 FORMALIZATION CONCEPT

EA provides the principles, methods and models used to design and realise an enterprise's organizational structure, process, information systems and infrastructure. (Braun and Winter 2005). EA proposals such as TOGAF, Zachman, TEAF and many others, though provide principles for architectural principles for EA and guidance for interoperability is deficient of unified business strategy for formalization of metamodels for validation (Martin et al., 2004) though this requirement is widely acknowledged (Quartel et al, 2009). The formalization concepts presented in this paper serves as focal point through which precision can be appropriated towards EA metamodel validation by use of ontology and Resource Description Framework Schema (RDFS). It incorporates thoroughness into validation criteria formulation allowing EA to effectively be aligned to business strategy and motivation. Formalization of VEM allows promotion of structured and iterative semantics that can substantially query EA ontology thereby producing a more dependable EA metamodels.

EAF, widely described as an "approach which includes models and definitions for documenting architectural descriptions" (FEAF, Gartner, TEAF, SEAM) makes it difficult to formerly relate its frameworks least of all the implementation components and artefacts that support their design. As this paper discusses metamodels in general, several frameworks have been examined in terms of their structures rather than content. Inspired by TOGAF and ArchiMate (TOG, 2014), EAF in this context is considered as a collection of metamodels and models which present a means for correlation and presentation of artefacts that conceptualise and describe an EA.

2.1 Model, Metamodel and Framework

A model, referred to as a collection of related components within a domain aims at providing functionality wholly or in part to achieve specific goals is explicitly an abstraction of a metamodel. It highlights the properties of the metamodel and conforms to its boundaries and constraints. Therefore, models describe the logical business functions or capabilities, business processes, human roles and actors, the physical organization structure, data flows and data stores, business applications and platform applications, hardware and communications infrastructure of a case domain.

A metamodel consists of explicit description of constructs and constraints of a specific domain. Though metamodels have also been described as comprising of a formalized specification of domain-specific notations which adhere to strict rule set for developing EA (Gudas and Lopata, 2007), metamodel consistently represents relevant artifacts of enterprise architecture both from a business perspective and from an Information Systems perspective. Thus it can be said that while models provide the reasoning about the systems being designed, metamodels specify the language for expressing these models.

In consideration of these definitions, a logical conceptualization of a metamodel for the validation extension of EAF is extrapolated. This is presented in Figure 1, and described briefly in section three. The construct is leveraged on the business layer of EA which in much taxonomy do not emphasize validation or the derivative values. An example of this is the ArchiMate Business Layer of ARCHIMATE (TOG, 2014), widely acclaimed as the empirical standard for EA modelling.

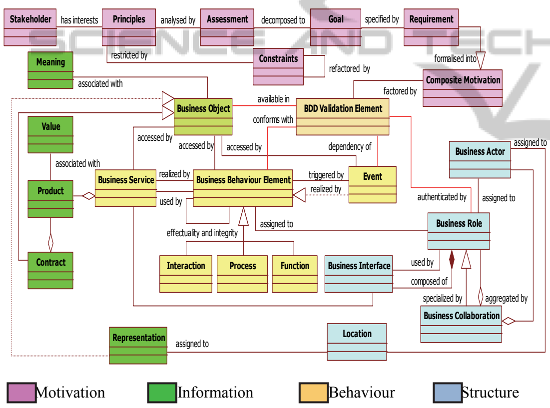


Figure 1: Validation Extension Metamodel for EAF.

The VEM presents an extension of a generic business layer of EA with embedded artefacts for validating its usability and important specifications for key performance indicators, business behaviour, perspectives and their relationships.

2.2 Contributions of VEM

Some of the contributions encapsulated within the concepts of the VEM are as follows;

- Extension of EA Modelling Language (EAML) with validation capability thus allowing transparency of decision patterns.
- Provision of a methodology for model transformation to ontology with capability for validation using unified query semantics.
- Enhancement of traceability capability for EA artefacts through RDFS makes inconsistencies in decision making more explicit.
- Extension of EAF modelling methodology with validation features means that the effect of changes can be made more manageable.

3 VALIDATION EXTENSION ELEMENTS DESCRIPTION

The validation extension elements are represented as high-level information models. The design goal on the metamodel links the business layer validation elements to business elements aggregated to composite behaviour and interaction. This sub classification allows further query relation to be distinctively applied to the business processes and business function to ascertain the artefacts integrity and effectuality respectively. Requirements which specify the Goals defined in Motivation appropriate a theme to be adopted by the evaluation iteration process. The query structure and semantics of the Validation elements allows criteria specified by constraints to be tested against Business Objects, Business Role and Business Events. The metamodel represents high-level conceptual constructs that are used to structure information evaluation, process support, information and EAF responsibilities on several model derivatives. Business viewpoints are derived from analyzing business Roles which are composite of Interface and Collaborations. However, not all layers of EA are covered in this metamodel. This is deliberate as the intention of this work is to espouse the alignment between the business strategy and motivation. The following therefore describe the unique artefacts of the metamodel.

3.1 Composite Motivation

Composite Motivation (CM) of the metamodel is composed of the intentions of the enterprise defined in the requirements, goals and constraints. The sources of these intentions are specified within Assessment. CM aggregates the theme for validation and relates with the core elements of the business layer. At a lower level of abstraction, internal drivers are assessed by SWOT analysis while at a higher level composite motivation are the external drivers namely constraints and embellishes principles, requirements and goals.

3.2 Validation Element

The core of this work is homocentric on Validation

Element (VE). VE provides the logic, semantics and links the ontology needed to validate the core business layer of the enterprise. The annotations attributed to the validation of the metamodel are essentially transformed into ontologies in order to allow the description and analysis of the relations between artefacts and composite motivation. The metamodel transformation to ontology expresses these annotations and constructs allowing validation semantics to be interjected in an automated iterative framework. The semantics also provide the basis for which the construct can be query through formalized statements and assertions. The major characteristic of validation element is that it consists of explicit description of constructs and constraints of the metamodel with ontology transformation attributes and mappings which adhere to distinctive and formalized business rule set. A business rule set in this context is a statement that defines certain aspects of the metamodel and serves as a guideline to determine the behaviour of viewpoints of the metamodel. In the case of this work, the business rule sets are translations of the following validation elements.

Availability VE (AVE) - Availability validation determines whether the artifact required for the actualization of business behavior is available in the metamodel construct.

Conformity VE (CVE) – This is validation to determine whether components meet some specified standards that has been stipulated for achieving desired business behavior.

Dependency VE (DVE) - This query deals with validating relationships with other artefacts and their ability to function as expected in normal and unusual situations within triggered events.

Authentication VE (UVE) - This refers to the assertion that the access properties of the component are substantiated with adequate privileges within roles and interfaces.

Effectual VE (EVE) - This validation assesses to what extent the intended business functions are achieved in relation to either the outcomes or impacts on other components.

Integrity VE (IVE) - This refers to the assertion that the accuracy and consistency of data stored and manipulated over the life cycle of a process in the metamodel is maintained.

3.3 Viewpoint

A viewpoint shapes the context of the metamodel with the validation element as viewed from a particular perspective. A number of standard viewpoints for modelling motivational aspects have been defined. Each of these viewpoints presents a different perspective on modelling the motivation of the EA focusing on defined abstractions of the metamodel. In this presentation, each viewpoint is an excerpt of business behaviour in relationship with a specific business role and encapsulates related requirements as extrapolated from a validation theme. The rationale for adopting this approach is to ensure that validation is focused on the intrinsic values for which the model is based.

As other elements which constitute the metamodel in Figure 1 are stereotype and are well explicated in definitions of many generic EAF such as TOGAF, SEAM, etc, no further explanations is given in this presentation.

4 APPROACH JUSTIFICATION

Several interpretations have been postulated to demonstrate that metamodels are closely related to ontologies (Gudas and Lopata, 2007) as both describe and analyze the relations between concepts. The extension put forward in this work is harnessed on this hypothesis to provide a methodology for expressing metamodel in a form that allows transformation to ontology description schema with capability for validation using unified query semantics. This extension is justified as in all cases; meaningful semantics provide the basis for which constructs can be interrogated by assertions (Gudas and Lopata, 2007). The rationale for the extension and annotation of metamodel construct with constraints is to allow distinctive transformation of model aspects to ontology with formalized specification as the entity adheres to strict rule set.

In contrast to the area of ontology languages where the Web Ontology Language (OWL) has become a de facto standard for representing and using ontologies, there is no agreement yet on the nature and the right formalism for defining mappings between ontologies. In a recent discussion on the nature of ontology mappings, some general aspects of mapping approaches have been identified (Choi et al., 2006). These aspects are adopted and extended for the mappings proposed here as they provide explicit specification of conceptualisation including descriptions of the assumptions regarding both the domain structure and the terms used to describe the domain (McShane, 2013). Hence, ontologies are central to semantic formalization as they allow harmonization of terms and relationship. As there are multiple strategies for mapping congruent information, relational schemas and metamodels to ontology, for establishment of consistency, this work adopts the direct mapping strategy as it defines a simple transformation which provides a basis for comparison and validation through RDF. The direct mapping takes as input a relational database derived from metamodel decomposition to generate direct algorithms and graphs. This allows values relating to motivation to be queried for the metamodel and its instances. Central to the approach is the extraction of business behaviour defined by the metamodel and transformation using a ubiquitous language for the domain driven design.

5 METAMODEL MAPPING TO ONTOLOGY

The process of ontology mapping in this approach is defined as follow;

- Given a model, identify testable artefacts as nodes and classes.
- Identify attributes of the node in terms of constraints.
- Identify relationship that exists between the nodes and slots. This ensures traceability.

Thus, the result of a mapping process is a set of mapping rules. Those mapping rules connect concepts in the transformation to concepts in metamodel. As a complementary method this approach provides critical insight into the contents and semantics of the metamodel artefacts but in general, it does not offer a means for validation of the underlying motivation of the metamodel.

5.1 Ontology Transformation Metaphor

A number of ontology transformation, integration methods and tools exist. Among them, are SEMAPHORE (Smartlogic, 2013), PROMPT (Noy, 2004, Choi et al., 2006), Protégé OWL (Horridge, 2009, McGuinness and Van Harmelen, 2004) are few which have working prototypes. These tools support the generating and merging of ontological elements such as class and attribute names from various sources. While SEMAPHORE automatically applies metadata and classification to improve context traceability, PROMPT provides more automation in merging ontologies. The most recent development in standard ontology languages is OWL from the World Wide Web Consortium (W3C). Protégé OWL makes it possible to describe concepts with richer set of operators and allow queries to be applied to the ontology. OWL for these reason and many others is preferred for the generation of ontology in this work.

5.2 Content Categorization

Content categorization is a link-based approach to classification. It is used in isolation or in conjunction with text-based classification to assign artefacts to one or more predefined categories based on their contents (Gyongyi et al., 2006). A number of modelling classification and knowledge management techniques have been applied to content categorization such as nearest neighbour, Support Vector Machine and Neural Networks (McShane, 2013). More recently, some preliminary studies have attempted to apply content categorization techniques into merging and mapping ontologies (Lacher and Groh 2001).

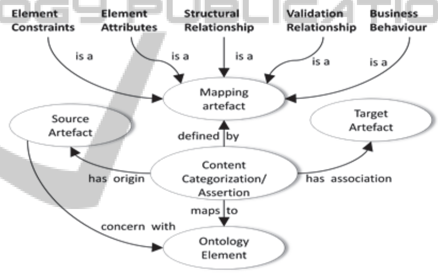


Figure 2: Conceptual framework for OWL Mapping.

Though these approaches are veteran, their analysis stipulates that generation and mergence of ontologies should follow a bottom up approach guided by application-specific instances is still widely practiced. In our approach, this theory is enhanced. While the general implementation of the mapping process identifies class artefacts from top down perspective, the mapping of the properties follow a bottom up perspective. The metamodel to ontology elements mapping are determined by similarity in characteristics per pair. In order to establish definitions of similarity and to support development of credible mapping, a framework for the mapping is defined in Figure 2. The diagram has associations that provide a way of establishing dependencies and traceability of the artefact within the schema. Definition is also attached to the content categorization in order to establish content and specify how the mapping of the ontology elements is related. The intention of the artefact pedigree is to provide an explanation of the source of its derivation.

6 MAPPINGS METHODOLOGY

This research proposes an autonomous formalisation metamodel for OWL ontology mappings. The metamodel is a consistent extension of the ArchiMate Business Layer for transformation to OWL Description Logic ontologies, RDFS and querying with SPARQL.

The UML profile can also be used to give a visual notation for specifying ontology mappings. The objective is to enable the specification of mappings in a generic sense and independent of a specific mapping language or a specific semantic relation. Examples of visual notation for this is as defined in the profile presented in Figures 3, 4, 5, and 6.

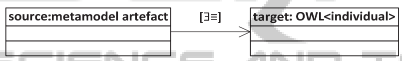


Figure 3: Direct Equivalence [∃≡] mapping between metamodel artefact and ontology element.

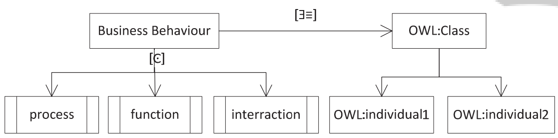


Figure 4: Equivalence mapping between business behaviour and complex class descriptions.

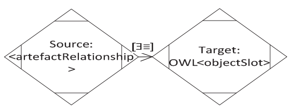


Figure 5: Equivalence mapping of Properties.

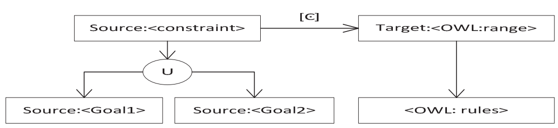


Figure 6: Composition mapping between complex motivation and ontology class.

6.1 Mapping and Creation of Classes

While there are no formalised ways of mapping generally acknowledged as standard by practitioners, to ensure that there is consistency in the methodology and to avoid overlaps of artefact mapping, a top-down class bottom-up slots approach is proposed. The structural Hierarchy is transformed to ontology using OWL Protégé (Figures 7 and 8) while the naming syntax is maintained to enforce clarity of the mapping process. There are two sibblings identified. The first is Composite_Motivation to represent a theme of motivation in a business behaviour to be queried. The second is Validation_Elements which encapsulates the core EAF artefacts to be validated.

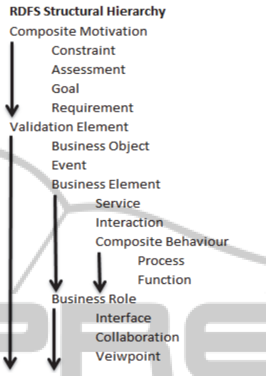


Figure 7: RDFS Hierarchy.

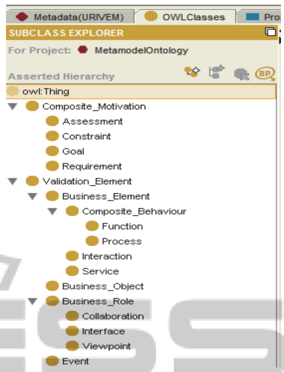
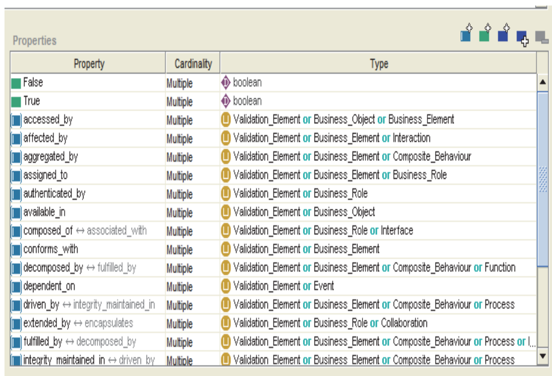


Figure 8: OWL implementation.

6.2 Characterization of Properties

Though the metamodel in Figure 1 shows bidirectional properties, characteristics of properties in EA are not strictly symmetrical. To characterise this relationship and bind the association, inverse functional characteristics can also be used. This allows the meaning of the property to be enriched as the implementation in Figure 9 shows with the ontology bindings to domains and ranges and a snapshot of many of the properties with unions of classes.

The diagram in Figure 10 portrays an extensible knowledge representation with elements of a theme for business behaviour from a viewpoint. The vocabulary generated with this ontology forms part of the triplestore that will be queried.



Properties

Property	Cardinality	Type
False	Multiple	boolean
True	Multiple	boolean
accessed_by	Multiple	Validation_Element or Business_Object or Business_Element
affected_by	Multiple	Validation_Element or Business_Element or Interaction
aggregated_by	Multiple	Validation_Element or Business_Element or Composite_Behaviour
assigned_to	Multiple	Validation_Element or Business_Element or Business_Role
authenticated_by	Multiple	Validation_Element or Business_Role
available_in	Multiple	Validation_Element or Business_Object
composed_of ↔ associated_with	Multiple	Validation_Element or Business_Role or Interface
conforms_with	Multiple	Validation_Element or Business_Element
decomposed_by ↔ fulfilled_by	Multiple	Validation_Element or Business_Element or Composite_Behaviour or Function
dependent_on	Multiple	Validation_Element or Event
driven_by ↔ integrity_maintained_in	Multiple	Validation_Element or Business_Element or Composite_Behaviour or Process
extended_by ↔ encapsulates	Multiple	Validation_Element or Business_Role or Collaboration
fulfilled_by ↔ decomposed_by	Multiple	Validation_Element or Business_Element or Composite_Behaviour or Process or I...
integrity_maintained_in ↔ driven_by	Multiple	Validation_Element or Business_Element or Composite_Behaviour or Process

Figure 9: Specification of Class Unions for Properties.

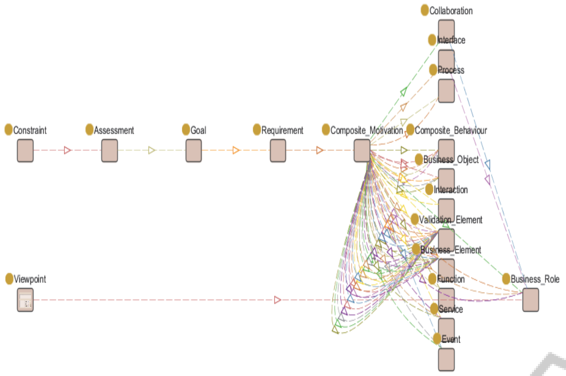


Figure 10: RDFS Graph of the metamodels from Viewpoint perspective.

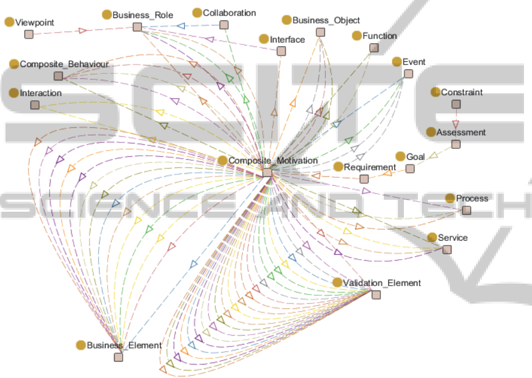


Figure 11: RDFS Graph of the metamodels showing the coverage of Composite Motivation.

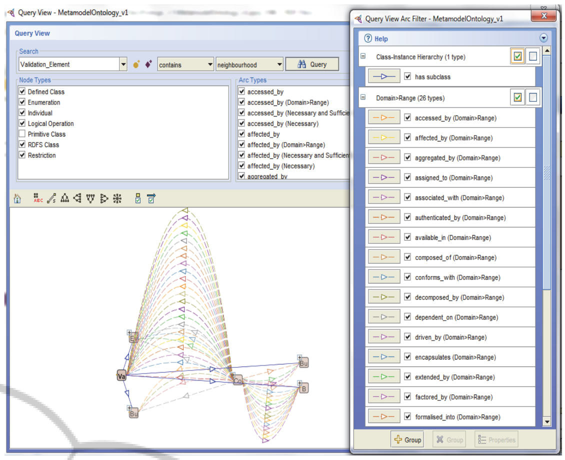


Figure 12: Querying the ontology using the Reasoner.

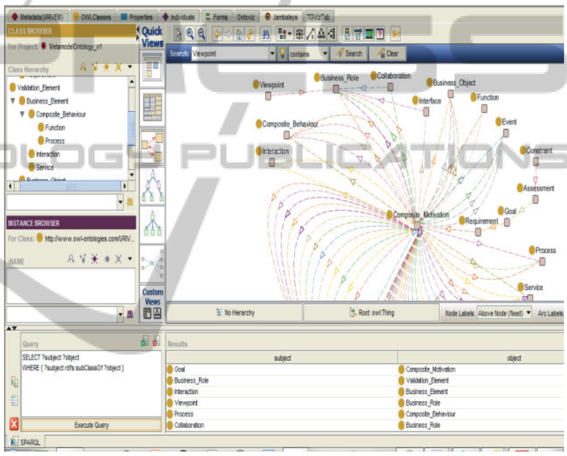


Figure 13: Querying the ontology using SPARQL.

7 QUERYING THE ONTOLOGY

While there are several literatures on querying ontologies, we try to apply a simple query construct to demonstrate whether the transformed metamodel can be validated using the triple stores generated with the RDFS. In OWL, the in-memory stores use the Reasoners to perform inferences in persistent RDFS stores, which otherwise can be difficult to perform. An example is shown in Figure 12.

The ontology can also be queried using SPARQL, recommended by W3C as standard query language for the Semantic Web. It focuses on querying RDF graphs at the triple level and RDFS, filtering out individuals and classes with specific characteristics or properties amongst many other benefits. The choice for SPARQL as a validation tool in this implementation is because it contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions and supports extensible value testing and constraining queries by source RDF graph.

Also the outcome of SPARQL queries can be results sets or RDF graphs as in Figure 13. This modest example illustrates that queries can in principle be used for constraint checking in order to validate motivation in EA models with assertions added as annotation properties to the selected class.

8 CONCLUSIONS

This paper has presented a formalization of concepts towards addressing validation concerns of EA through the use of ontologies and queries based on constraints specified in the model's motivation taxonomy. The postulations based on experimental research, is grounded on the extension of an EAML with validation capability and substantiation of its motivation using open source web ontology. It delved into the use of semantics triples, Resource Description Framework Schema (RDFS) and relational graphs to map EA metamodel and its attributes directly into classes and slots using end-to-end conventional formalization approach that can be applied within heterogeneous EA domains. The paper also expounds on proposals that postulate implementation of the approach, enabling formalized traceability of EA validation and contributes to effective validation of EA through refined taxonomy semantics, mappings and alignment of motivational goals to business behaviour and specifications. The application of the theoretical principles presented in this paper is a contribution towards an approach for providing solutions to issues surrounding EA validation in consideration of structural complexities in its metamodels. A validation metrics for testing EA artefacts has been conceptualized and encapsulated into the metamodel as well as methodology for model transformation to ontology description schema, with capability for validation using Reasoner and a unified query language. This consequently adds agility to the organization's EA modelling processes.

As validation of EAF is an area that currently draws very little diligence amongst practitioners due to complexities, this paper presents a novelty methodology through which much research can be initiated. This include amongst many others a case for integration of divergent EAFs through a common vocabulary using ontology so as to allow better congruency, traceability, validation and alignment of business objectives to Information Technology.

REFERENCES

Braun, C., & Winter, R., 2005. A comprehensive enterprise architecture metamodel and its implementation using a metamodeling platform. *Desel, Jörg; Frank, Ulrich*, 24-25.

Choi, N., Song, I. Y., & Han, H., 2006. A survey on ontology mapping. *ACM Sigmod Record*, *35*(3), 34-41.

Gudas, S., & Lopata, A., 2007. Meta-model based development of use case model for business function. *Information Technology and Control*, *36*(3), 302-309.

Gyongyi, Z., Garcia-Molina, H., & Pedersen, J., 2006. Web content categorization using link information.

Horridge, M., 2009. A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition1. 2. *The University Of Manchester*.

Lacher, M. S., & Groh, G., 2001, May. Facilitating the Exchange of Explicit Knowledge through Ontology Mappings. In *FLAIRS conference* (pp. 305-309).

Lankhorst, M., 2013. *Enterprise Architecture at Work: Modelling, Communication and Analysis* . Springer, 2013.

Martin, R. A., Robertson, E. L., & Springer, J. A., 2004, April. Architectural Principles for Enterprise Frameworks. In *CAiSE Workshops (1)* (pp. 151-162).

McGuinness, D. L., & Van Harmelen, F., 2004. OWL web ontology language overview. *W3C recommendation*, *10*(2004-03), 10.

McShane, M., & Nirenburg, S., 2013. Use of Ontology, Lexicon and Fact Repository for Reference Resolution in Ontological Semantics. In *New Trends of Research in Ontologies and Lexical Resources* (pp. 157-185). Springer Berlin Heidelberg.

Noy, N. F., 2004. Semantic integration: a survey of ontology-based approaches. *ACM Sigmod Record*, 33(4), 65-70.

Quartel, D., Engelsman, W., Jonkers, H., & Van Sinderen, M., 2009, September. A goal-oriented requirements modelling language for enterprise architecture. In *Enterprise Distributed Object Computing Conference, 2009. EDOC'09. IEEE International* (pp. 3-13). IEEE.

Smartlogic, 2013 found at :http://www.smartlogic.com/home/products/products-overview. Accesses December 2013.

The Open Group (TOG), TOGAF (Version 9.1). Available at: http://www.opengroup.org/architecture/ Accessed on: January, 2014.