International Journal of Advanced Research and Publications
ISSN: 2456-9992

Contextualization Of Enterprise Architecture Business And Application Models Using OWL Visualizer For Complexity Analysis

Joe Essien

Veritas University, Faculty of Natural and Applied Science, Abuja, Nigeria,
joessien@veritas.edu.ng

Abstract: Ontologies are an essential resource for knowledge exemplification. The ontology of a complex system can be intricate due to role relations between several concepts, diverse attributes and incongruent instances. In this paper, the visual analytics clarification, based on different coordinated viewpoints for exploring diverse ontology facets and an innovative deployment of queries to streamline traceability of abstractions is used to moderate the complexity of ontology visual representation. Though many researches have delved into the transformation of Enterprise Architecture models to ontologies, visualization of ontology abstractions have not been exhaustively exemplified especially in the context of system development. One of the major reasons why this is important is the need to effectively align business processes with application framework within organizations, especially those with numerous disparate systems. A method for ensuring the synchronization of the models and ontologies has remained a gap and an open area for more research. The transformed ontologies have not also been rigorously explored in terms of validation and traceability. In this paper, an innovative approach for aligning the business layer and the application layer of ArchiMate is presented to ensure that application can align its functionalities consistently with enterprise goals and motivation. This paper deploys the pension scheme system as a case in context to model the collaboration of business and application layers of an enterprise architecture viewpoint and transforms the viewpoint to ontology that can be queried for better visual analytics perception. The outcome of this work includes an elaborate design phase of enterprise architecture artefacts, and an executable OWL prototype that realizes architecturally significant of the case in context.

Keywords: Application Model, Business Models**,** Enterprise Architecture, Ontology, OWL, Pension, Semantic Reasoners, Traceability.

1. Introduction

Queries play an important part in developing and utilizing ontologies designed in OWL Protégé. Automated reasoners such as Pellet, FaCT++, HerMiT, ELK and so on take a collection of axioms written in OWL and offer a set of operations that can be used to query the ontology's axioms such as the inference of a hierarchy for classes described in ontology. One of the most widely known usages of the queries is to establish relationships, classifications and visual analytics which are also incidentally an archetype adopted in the implementation of traceability within enterprise architecture models. A significant prospective application of ontologies can be associated with the application of semantic reasoners and how the approach can be deployed to infer new information that is not unambiguously described in Enterprise Architecture (EA) models. Such is the case of modelled collaboration of artefacts for the business and application layers of EA and the use of the model to ontology transformation together with a synchronized query analytics to identify possible traceability based on constraints that are related to the EA ontology. To effect this paradigm, the transformed ontology is embedded with common reference schema, together with collaborative artefacts from the two layers of EA schema. The constraints inferences that are embedded encapsulate the knowledge of the domain experts for each of the two layers are modelled using artefacts and relationships from the ontology. Similar constraints are also defined using the Protégé tool and this forms the EA model representation in the ontology. These rules are used by the query analytics to infer new interpretations and provide the system with a means that reveals whether or not there is consistency and alignment between the business and application layers.

2. Integrating Ontologies with EA Models

Integration of ontologies with EA models can be achieved through ontology mapping and alignment. Ontology mapping provides the framework for representing shareable and reusable knowledge across a domain [13], as well as enabling data analytics through interoperability and integration [14]. Though ontology alignment evaluation poses an open challenge [24], dynamic ontologies is an evolving concept and their mappings in any implementation need to be maintained as a service. The challenges include how to determine solutions for better ontology mappings, as well as how to select ontologies before their application. In addition, tools and algorithms for ontology mapping are complex, including the requirements, evaluation of tool capability and ascertainment of quality of mappings. Generating ontology mappings can provide several other challenges as for instance, words in language can have abstruse meanings depending on the context. This ambiguity implies that it is inadequate to merely match class names, terms, or labels for effective ontology mapping. Therefore, it is imperative to apply context to resolve ambiguity, which includes contextual knowledge and associations among concepts [10]. Ontologies are one of the mechanisms to encode the semantics for many applications of human knowledge including machine-readable capabilities [1], [2]. Ontologies are vital for capturing meaningful relationships to allow users to search or browse relationships and to identify patterns from analysis [3], [4], [5], [6]. Ontologies are dynamic and their entities evolve over time. Common changes can include class addition; class deprecation; combination of classes; and hierarchical relationships. Therefore, ontology mappings are not static resources and need to evolve symmetrically with their source ontologies. Thus, it follows that any ontology mapping needs to be provided not only as a one-off process, but also as an ongoing service [7]. Despite that most commonly used ontologies are accessible openly, many researchers and organisations build their own ontologies; either to expand on a particular knowledge domain of an existing ontology or for areas that are not well explored. Consequently, two major use cases for an ontology mapping services have been identified: (i) mapping among public ontologies; and (ii) mapping between public and internal ontologies [8]. The former can be achieved with a repository of mappings among popular public ontologies, which has the benefit that it can manage updates, utilize existing mappings, and generate new ones. The second case can be approached by providing tooling such that the user can generate bespoke mappings from their internal ontologies to public ontologies as required. An Ontology mapping service should also adopt semantics that allow synchronization, in addition to transitivity, to enable mapping at scale across whole ontologies [9]. Ideally, it should also allow the addition of user-transitive content and validation of predicted mappings, assisted by crowd-sourcing, which has been used for ontology query analytics and appropriate for establishing traceability and alignment of model viewpoints. Specifically, for each mapping, the service should provide annotation with suitable metadata and documentation, to enable interoperability and reuse.

3. The OWL and Ontology Modelling

In recent times, there has been more endorsement of articulating ontologies using ontology languages such as the Web Ontology Language (OWL) [15]. OWL is a semantic web computational logic-based language, designed to represent rich and complex knowledge about things and the relations between them. The OWL language is characterized by formal semantics. It is structured upon the World Wide Web Consortium's (W3C) XML standard for objects called the Resource Description Framework (RDF) and its semantic graph databases, also known as RDF triple stores [13], [9]. Though RDF Schema (RDFS) offers taxonomic relations, object relations and data type properties, OWL has a richer vocabulary and is more expressive [11]. With OWL it is possible to specify cardinalities of object relations and data type properties (attributes) as well as use logical operators in definitions (e.g. use union of classes as a range of relation)[13]. Thus OWL, used together with an OWL reasoner in triple stores, enables query analytics and validation of logical inconsistencies in taxonomies and ensures traceability for classes and individual relationships [12]. Also, OWL has the provision for defining equivalence and difference between instances, classes and properties. These relationships facilitate matching of concepts even if various data sources describe these concepts disparately to a certain degree. One of the main features of ontologies is that, by having the essential relationships between concepts built into them, they enable automated reasoning about data. Such reasoning is easy to implement in semantic graph databases that use ontologies as their semantic schemata [14], [15]. The query capability of ontologies provides a more coherent and easy traceability as models are stratified between layers of an EA viewpoint taxonomy. Ontologies are suitable for synchronized reasoning as it provides additionally the means to denote any data formats, including unstructured, semi-structured or structured data, enabling smoother data integration, easier concept and text mining, and model-driven analytics12]. High level syntax is used to specify the OWL ontology structure and semantics [13]. The OWL abstract syntax depicts ontology as a categorization of annotations, axioms and facts. The annotations usually consist of both automated and human oriented meta-data [15]. Information regarding the classes, properties and individuals that constitute the ontology is encapsulated in axioms and abstract data. Each class, property and individual is either anonymous or identified by an URI reference. The abstract data states the specifics about an entity or a pair of individual groups of entity identifiers. Axioms specify the attributes of classes and properties. This style is similar to frame languages, and quite dissimilar to well known syntaxes for Descriptive Logic (DL) and Resource Description Framework (RDF) [13].

4. Modelling Business and Application Layers

In this work, the ArchiMate modelling language will be used as a tool for modelling the EA viewpoints of Business and Application layers. The ArchiMate modelling language provides a coherent and a holistic view of an enterprise in terms of its products, services, business processes, actors, business units, software applications and more [19]. Furthermore, it presents a step-wise approach that shows how its model transformation is achieved and, in doing so, shows also such transformation. The Business Layer is the part of the ArchiMate model that represents the closest level of a business to the stakeholder. The active structures of the Business layer concept consist of Business Actor, Business Role, Business Collaboration, Business Interface and Location [16], [17]. The Business Role establishes responsibilities for behavior. These responsibilities can be associated with the Actor responsible, and with functions and processes that carry out the actual behavior. The metamodel of ArchiMate gives an overview of the application layer concepts and their relationships. The main active structure concept for the application layer is the application component. This concept is used to model any structural entity in the application layer including re-usable software components that can be part of one or more applications, and also complete software applications, sub-applications, or information system [16],

[18]. One of the greatest benefits of using ArchiMate for business process modelling is that the language is universal. It is designed to be as clear and simple as possible [19]. Anyone within or connected to a business can also easily understand an ArchiMate model. The terminologies used are derived from other EA models itself, thus creating a bridge between them. This consistency helps create cohesion between stakeholders and processes in an organization. Communication is made simple and efficient with this standard language. ArchiMate concepts are not only simple, but they are also precise [18], [19]. The modeling language facilitates the design of a representation of the business operation with precision.

5. Conceptual Specification for the Proposed Pension Scheme System

Pension is an arrangement of providing people with an income when they are no longer earning a regular income from employment. Retirement plans may be set up by employers, insurance companies, the government or other institutions such as employer associations or trade unions. Retirement pensions are typically in the form of a guaranteed life annuity, thus insuring against the risk of longevity. Given the discouraging record of the traditional pension system and the limited coverage of the private-sector pension schemes, there is need for robust pension fund management system that can handle the pension process efficiently. Manual processes of pension management are costly and results in problems such as theft and fraudulent practices. Other aberrations in the process include ineffectice investment of pension funds and the return on investments, human errors in crediting of a pensioner's account, wastage of materials, time, consumables and lack of adequate information flow. The Department of Human Resource Management of a large governmental parastatal in Nigeria plans to design and implement a web-based pension fund management application to replace the semi-automated system currently in place. Major problems of the pension fund administration are inexperienced and untrained staff, non-payment or delay in the payment of pension and gratuity to the beneficiaries. Consequently, retirees usually go through tough times and rigorous processes before they were eventually paid their pensions, gratuity and other retirement benefits. Most times, the money to pay their benefits is not usually available. Essentially, the old scheme is beset with a lot of challenges and problems. Besides the aforementioned, other problems have been identified as demographic challenges and funding of outstanding pensions and gratuities, merging of service for the purpose of computing retirement benefits. Generally, these problems coupled with the administrative bottlenecks, bureaucracies, corrupt tendencies and inefficiencies, and the economic downturn have resulted in erratic and the non-payment of terminal benefits as at when due. Thus, the management of pension scheme in this institution is inundated by multiple and diverse problems resulting in accumulated arrears of pensioners. The Pension managers are also faced with the problem of determining appropriate investment portfolios, underwriting insurance company as well as the insurance brokers/consultant. There is also the problem of documentation and filing in pension offices. At the Department of Human Resource Management of the institution, a retiree is expected to mandatorily complete a set of clearance forms with various units of the institution so as to verify any indebtedness or otherwise. Some retirees fail to comply with this requirement, possibly due to the cumbersomeness of the clearance procedure, leading to delay in the processing and payment of their entitlements. A new system is desire that can manage the current poor and chaotic filing system and ensure payment of pension and gratuity to the beneficiaries on time. The new system should provide for the assessment and training of staff especially the incompetent and inexperienced pension staff. The system should adopt a workflow that improves document handling with appropriate checks and balances. The calculations and computations of gratuity and pension should be more accurate and précised as opposed to the current erroneous payments and entitlements wrongly computed. Many of the pension records should be automated to eradicate frauds and errors such as ghost pensioners, double payments and omission of names. Intelligent reports should be generated from the system to provide a strategic method of downsizing and rationalization of personnel in order to reduce the operating costs, labor costs and promote efficiency.

6 Ascertaining Values of Case Proposition

To prevent loss in translation from business needs to IT solutions, the VPEC-T analysis is used. The VPEC-T approach provides a collection of mental filters and guides to a simplified communication method [20]. It is best suited for analysis of IT specifications where communication and interaction between actors and business behavior can easily result in ambiguity [20]. VPEC-T analysis is also used when analyzing the expectations of multiple parties with different views of a single system. As each party may have different priorities and responsibilities, and may want to achieve different objectives, VPEC-T analysis is commonly adopted and used for complex enterprise IT systems or large-scale development efforts [20]. Based on the conceptual specification for the proposed pension scheme system, the following analysis is derived.

Table 1: *Table of VPEC-T Values*

Values	
As-is	To-be
Inexperienced and untrained staff	Provide for the assessment and training of staff
Administrative bottlenecks, bureaucracies, corrupt tendencies and inefficiencies	Adopt a workflow that improves document handling
Erroneous payments and entitlements wrongly computed	Calculations and computations of gratuity and pension should be accurate
Manual processes of pension management results in problems such as theft and fraud	Pension records should be automated

Table 2: *Table of VPEC-T Policies*

Policies	
As-is	To-be
Non-payment or delay in the payment of pension and gratuity to the beneficiaries resulting in accumulated arrears	Ensure payment of pension and gratuity to the beneficiaries on time
Retirees are expected to complete a set of clearance forms with various units of the institution so as to verify non-indebtedness.	Simplify documentation of Loans acquisition process and integrate Loan Records with Pension System
Manual processes of pension management results in problems such as theft and fraudulent practices	Eradicate fraud and errors such as ghost pensioners, double payments and omission of names through verification
Demographic challenges and funding of outstanding pensions and gratuities	Intelligent reports should be generated from the system to aid in a strategic decision making
Manual processes of pension management is costly	Reduce the operating costs, labor costs and promote efficiency

Events:
Several event filters are identified for the new system. These form the business triggers for actions that result in the actualization of one or more business objectives. The VPEC-T framework requires the prioritization and sequencing of these triggers and establishment of relationship between them. The following events (though not exclusive) which can trigger some actions are identified in this work.

- Retirees indebtedness status to be confirmed
- Retirees to complete periodic mandatory update
- Retirees to ensure their records are up to date
- Pensions to be paid monthly at specific date.

Content:
Content consist of information in any form that are required for the project as well as data held formally in databases and other structured interfaces. The content of the proposed system is classified into four categories namely Employment documents, Retiree records, Pension Payment Records and, Business Intelligent Reports. Content may be related to pre-requisite information, required for processes and acted upon to result in an event or information generated from an event occurrence.

Trust:
In the case of this work, Trust is not considered as a deliverable since it is unlikely to be listed as a concern unless it is impossible to resolve and appear to be a threat to the success of the project.

Sequel to this analysis, the following EA Business and Application models to-be are design and modeled for transformation to ontology.

7. Business and Application Layer Models
The ArchiMate is used to model the complex collaboration and relationships between the artefacts of the Business and Application layers in real-time for the proposed software systems. It presents a multi-disciplinary platform that facilitates and integrates visual understanding of the system from different stakeholder's perspectives. The viewpoint captures and specifies the organizational requirements using design elements of the modelling semantics expressed in terms of objects, processes, roles, services, collaborations, stakeholders, events, interactions, functions, interfaces and specialized relationships. The elements of the business interface models explicitly the logical channels and establishes relationships for roles and services for the design environment. With reference to Figure 1, all the business actors are assigned business roles with relationships that depict their specific viewpoints of the system.

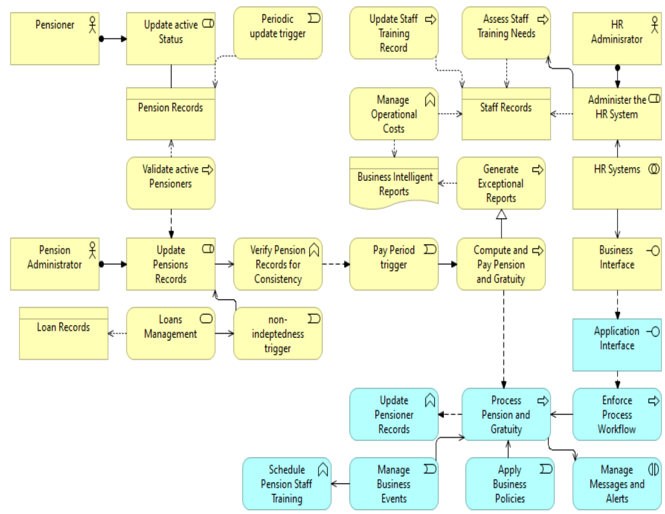

Figure 1: *Business and Application Layer Models*

Business roles with certain responsibilities or skills are also assigned to business processes or business functions. For instance, the Business Role of the stakeholder "Pension Administrator" is assigned the function of "Verify Pension Records for Consistency" and on the occurrence of the Business Event "Pay Period trigger" initiates the Business Process of "Compute and Pay Pension and Gratuity". In this model, a business process or function is interpreted as the internal behavior of a single business role. The Business Interface exposes the functionality of a Business Layer to the Application Layer. A business interface is a unit of behavior similar to a business process or function, but is executed in collaboration of two or more roles within the organization. The model depicts multiple viewpoints of the information critical to design of the new system. This includes the Pensioner, the Pension Administrator and the HR Administrator. It also encapsulates the functional hierarchy of services and the cross-functional nature of processes. One way ArchiMate model helps to balance goals is by supporting multiple views of a given situation. Each model helps in the understanding of the relationship between these different elements and viewpoints.

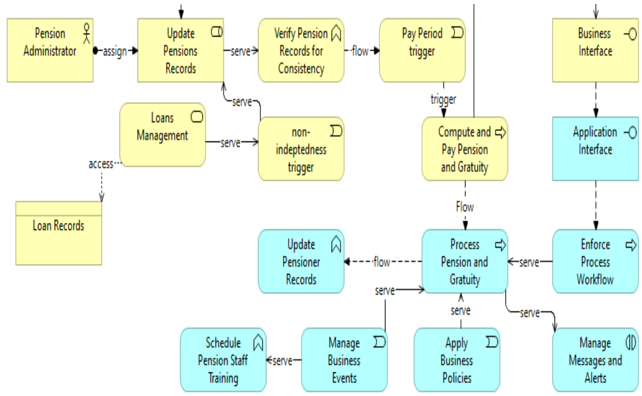

Figure 2: *Viewpoint of Pension Administrator and Application model abstraction*

The solution model presented in Figure 2 actually represents this balance by implementing the business view as an application solution. In this implementation, the annotated viewpoint of the stakeholder "Pension Administrator" is considered. Based on the viewpoint on Figure 2, logical clustering in terms of design object categorization and classification is mapped. In each case, these object tags are associated with relational information about the properties and features of the object, and tools that give insight into how to organize the object to maximize their synchronization with the ontology. The reason such a tagging is beneficial is that the properties and features of object should be implicit in terms of their relationship to the whole, main system taxonomy. Logical data clustering in this case works to uncover the design element nature, element type, and element level of abstraction, and thereby the aspects of their architecture. The relations constitute a combined structure of the model and attributes of the properties. This is precisely what the logical clustering of classification and categorization tries to relate to the ontology.

8. Transforming the Model to Ontology

The transformation of model to ontology is not conventionally injective, as some OWL abstract ontologies that do not adopt standard reserved vocabulary can still by definition map into equal RDF graphs. Be that as it may, the few instances where this can occur are with constructs that have similar meaning, such as several DisjointClasses axioms having the similar effect as one larger one. A transformation table shown in Table 1 is adopted to provide transformation rules that transpose the abstract syntax to the OWL exchange syntax. In rare cases, particularly for the DifferentIndividuals construct, there could be variant transformation rules. In such instances either rule can be chosen, resulting in a non-deterministic transposition. RDF graphs will be generated as data model that formally describes the semantics, or meaning of information relating to EA models transformed. A direct visual notation for one-to-one equivalence mapping is as defined in the profile presented in Figure 3. A direct mapping also is achieved for property using an equivalence relationship as in Figure 4.

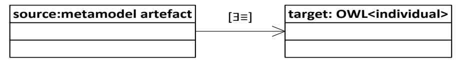

Figure 3: *Direct Equivalence mapping (Source [21])*

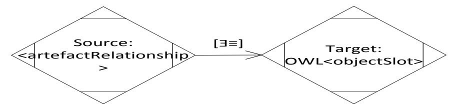

Figure 4: *Equivalence mapping between Relationship and Ontology Slot (Source [21]*

Table 3: *Mapping of Model to Ontology Hierarchy*

EA Layer	RDF TRIPLES			Mapping Profile Level		
Property Tags	Subject UID	Predicate Desc.	Object UID	L1	L2	EQR
Business Layer B1	Pensioner	assigned to	Update Status	L1		Ͼ
B2	Update Active	associated with	Pension Records		L2	∃≡
B3	Pension Records	accessed by	Validate Active		L2	∃≡
B4	Pension Records	accessed by	Periodic Update		L2	∃≡
B5	Pension Admin	assigned to	Update Records		L2	∃≡
B6	Update Records	served by	Verify Records		L2	∃≡
B7	Update Records	served by	Indebtedness		L2	∃≡
B8	Loan Records	accessed by	Loan Mngt		L2	∃≡
B9	Loan Mngt	served by	Indebtedness		L2	∃≡
B10	Verify Records	served by	Pay Event Pension		L2	∃≡
B11	Pay Event Pension	triggered by	Compute Pension & Gratuity		L2	∃≡
Application Layer A1	Compute Pension & Gratuity	actualized by	Compute Pension & Gratuity	L1		Ͼ
A2	Compute Pension & Gratuity	served by	Enforce Process workflow		L2	∃≡
A3	Compute Pension & Gratuity	served by	Manage messages & alerts		L3	∃≡
A4	Compute Pension & Gratuity	served by	Apply Business policy		L3	∃≡
A5	Compute Pension & Gratuity	served by	Manage Business Events		L3	∃≡
A6	Compute Pension & Gratuity	access by	Update Pension Records		L3	∃≡
A7	Manage Business Events	served by	Schedule Pension Staff Training Schedules			

The concept of identifying properties and features in terms of type, nature, tiers, and levels has been applied in a diverse range of fields, including anthropology, sociology, psychology, engineering, economics, positivism, functionalism, conflict theories, and mathematics, to name but a few [22]. Therefore, while concepts of logical clustering exist in nearly all the mentioned areas and also nearly all areas of information technology (IT), from application modeling, to measurements, reporting, business intelligence, this maturity does not exist to this degree within the business/process modeling transformation to ontology [23]. Although some concepts exist regarding how to tag a process according to management, or a main or supporting process, there are almost no existing concepts for process nature or process decomposition, or at least, none that put it all together for an integrated and standardized process-tagging of models to OWL ontologies adds to process modeling, process engineering.

9. Design of the Ontology Frames

Figure 5 shows the object properties of the transformed pension ontology. Property postulates that for any given individual in a class, there must be at most one out going relationship. Inverse property asserts that for any individual in a class there should be at most one incoming relationship, through the property, which can uniquely identify the subject. To avoid an inconsistent ontology, each individual has a unique relationship annotated with a property tag through the functional or inverse-functional property. The transitive property is defined as the property which shows transitive implications among individuals, such that if an individual x is similar to individual y, and y is similar to individual z then it can be implied that the individual x and z are also similar, through a transitive relation.

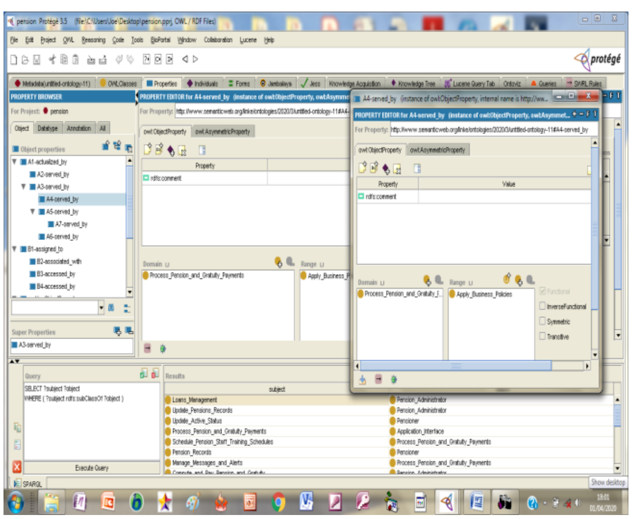

***Figure 5:** The design framework of the ontology*

10. Visualization and Complexity Analysis

Ontologies are an essential resource for knowledge exemplification. The ontology of a complex system can be intricate due to role relations between several concepts, diverse attributes, and disparate instances. In this section, the visual analytics clarification, based on different coordinated viewpoints for exploring diverse ontology facets and an innovative deployment of queries to define the degree of interest suppression is used to moderate the complexity of the ontology visual representation. Figure 6 shows a concise visual analytics of the domain and tasks representation.

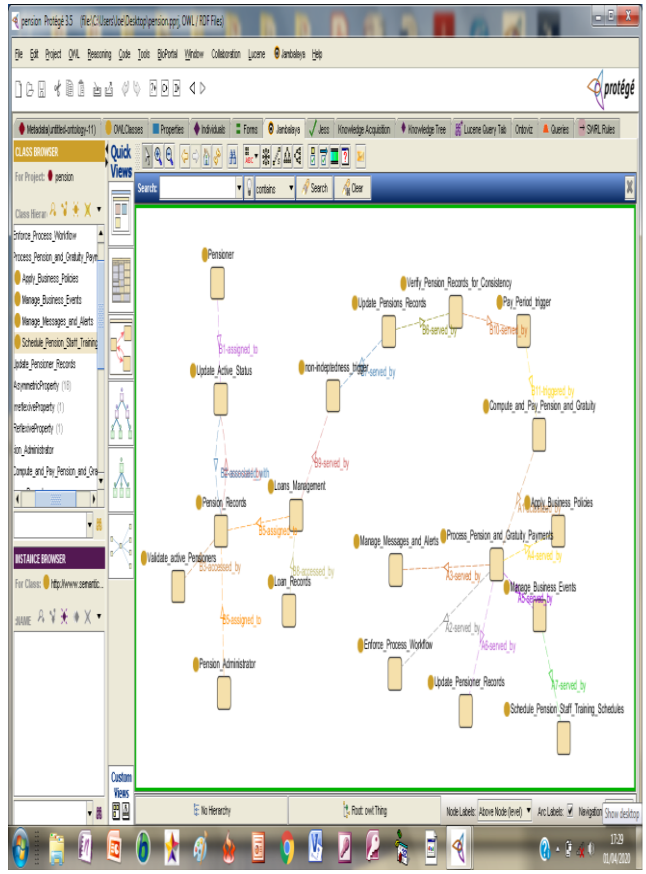

***Figure 6:** Ontology of transformed model*

Visual Analytics facilitates the understanding of the domain and tasks represented by ontologies, thus allowing the exploratory analysis to optimize the comprehension of data semantics including non-explicit relationships between data [24]. Through the query technique, concise abstraction of relevant view is achieved from the ontology, separating unnecessary information and facilitating the analysis and understanding of correlated artefacts. This is shown in Figure 7 and Figure 8.

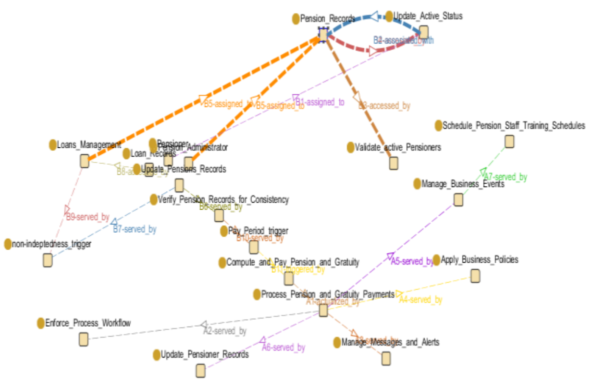

***Figure 7:** Simplifying Traceability within the Ontology*

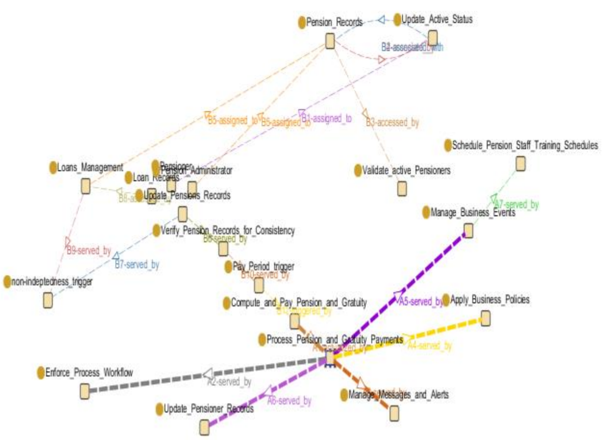

***Figure 8:** Simplifying Complexity in Process Pension*

Through this approach, limitless evaluations of the ontology can be devised as in Figure 9 with RDF graphs that can be evaluated to reinforce the importance of providing visualization and analysis dedicated to the schema and instances levels of ontologies for the discovery of explicit and non-explicit information regarding the EA model.

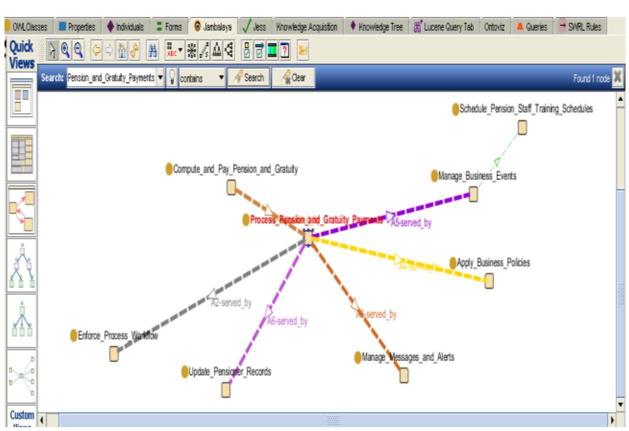

***Figure 9:** Querying the Ontology for Process RDFS*

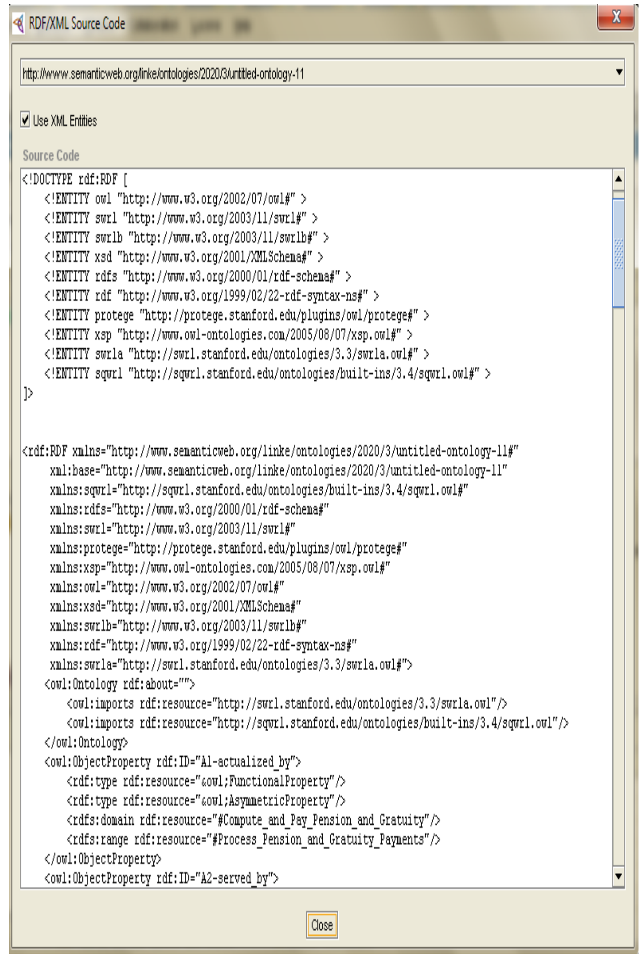

***Figure 10:** Excerpt from the RDF/XML source code*

An excerpt of the RDF/XML implementation of the ontology is shown in Figure 10.

11 Conclusion

This paper has contributed to knowledge by proffering an approach that can be applied to query and simplifying taxonomy complexity generated from Enterprise Architecture models. In particular, the viewpoint of an instance from a collaboration of Business and Application layers was exemplified in an actual business requirement specification for systems development. It has been affirmed that most processes that are complex tend to be inflexible, complicated and less susceptible to change management [24]. The major reason for simplifying complexities in EA models is to facilitate traceability and alignment of requirement specification to set goals of an organization. In this paper, an organisation requirement specification was modelled into EA taxonomies and the taxonomy transformed to ontology. The ontology was categorized extensively to allow the complexity of the initial EA taxonomy to be queried thus simplifying the intricate associations of business needs with application design.

References

[1] P.L. Whetzel, et al. NCBO Technology: powering semantically aware applications J. Biomed. Semant., 15 (Suppl. 1) (2013), p. S8

[2] R. Hoehndorf, et al. The role of ontologies in biological and biomedical research: a functional perspective Brief Bioinform., 16 (2015), pp. 1069-1080

[3] Croset, et al. Flexible data integration and curation using a graph-based approach Bioinformatics, 32 (2016), pp. 918-925

[4] D. Gomez-Cabrero, et al. Data integration in the era of omics: current and future challenges BMC Syst. Biol., 8 (Suppl. 2) (2014), p. I1

[5] V. Lapatas, et al. Data integration in biological research: an overview J. Biol. Res., 22 (2015), p. 9

[6] H. Zhang, et al. An ontology-guided semantic data integration framework to support integrative data analysis of cancer survival BMC Med. Inform. Decis. Mak., 18 (Suppl. 2) (2018), p. 4

[7] A.S. Rathore, et al. Role of knowledge management in development and lifecycle management of biopharmaceuticals Pharm. Res., 34 (2017), pp. 243-256

[8] M.P. van Iersel, et al. The BridgeDb framework: standardized access to gene,protein and metabolite identifier mapping services BMC Bioinf., 11 (2010), p. 5

[9] J.M. Mortensen, et al. Is the crowd better as an assistant or a replacement in ontology engineering? An exploration through the lens of the Gene Ontology J. Biomed. Inf., 60 (2016), pp. 199-209

[10] M.R. Kamder, et al. A systematic analysis of term reuse and term overlap across biomedical ontologies Semant. Web, 8 (2017), pp. 853-871

[11] World Wide Web Consortium. "OWL 2 web ontology language document overview." (2012).

[12] Sirin, Evren, et al. "Pellet: A practical owl-dl reasoner." Journal of Web Semantics 5.2 (2007): 51-53.

[13] McGuinness, Deborah L., and Frank Van Harmelen. "OWL web ontology language overview." W3C recommendation 10.10 (2004): 2004.

[14] Horrocks, Ian, and Peter F. Patel-Schneider. "A proposal for an OWL rules language." Proceedings of the 13th international conference on World Wide Web. 2004.

[15] Knublauch, Holger, et al. "The Protégé OWL plugin: An open development environment for semantic web applications." International semantic web conference. Springer, Berlin, Heidelberg, 2004.

[16] Josey, Andrew, et al. "An introduction to the ArchiMate® 3.0 specification." White Paper from The Open Group (2016).

[17] https://archimatetool.gitbook.io/project/archimate-business-layer/business-layer-structure-concepts Accessed 30/03/2020.

[18] https://archimatetool.gitbook.io/project/application-layer/archimate-application-layer-active-layer-concepts, Accessed 30/03/2020.

[19] de Kinderen, Sybren, Khaled Gaaloul, and Henderik A. Proper. "Bridging value modelling to ArchiMate via transaction modelling." Software & Systems Modeling 13.3 (2014): 1043-1057.

[20] Green, Nigel, and Carl Bate. Lost in translation: A handbook for information systems in the 21st century. New York: Evolved Technologist Press, 2007.

[21] Essien, Joe. Model driven validation approach for enterprise architecture and motivation extensions. Diss. University of West London, 2015.

[22] Razzaque, Mohammed Abhur, Simon Dobson, and Paddy Nixon. "Categorization and modelling of quality in context information." (2006).

[23] Born, Matthias, et al. "Business functions ontology and its application in semantic business process modelling." ACIS 2008 Proceedings (2008): 110.

[24] Silva, Isabel Cristina Siqueira, Giuseppe Santucci, and Carla Maria Dal Sasso Freitas. "Visualization and analysis of schema and instances of ontologies for improving user tasks and knowledge discovery." Journal of Computer Languages 51.