SOFTWARE ENGINEERING APPROACH FOR DOMAIN ONTOLOGY DEVELOPMENT: A CASE STUDY OF ISLAMIC BANKING PRODUCT

Ahmad Shaharudin Abdul Latiff[1], Haryani Haron[2] and Muthukkaruppan Annamalai[3]
Universiti Malaysia Kelantan
Kota Bharu, Kelantan
[1]shaharudin@umk.edu.my

Universiti Teknologi MARA
Shah Alam, Selangor
[2]haryani@tmsk.uitm.edu.my, [3]mk@tmsk.uitm.edu.my

ABSTRACT

In order to be effective, information systems that cater for the need of a particular domain ought to deploy suitable ontology of the industry. The ontology will be useful for their general operations, decision making and also to facilitate the financial engineering. The software engineering approach develop ontologies that are more reliable, long lived, and continually adapted. It also enables the ontology to adapt to the context of its purpose. The Software Engineering Approach for Domain Ontology Development adapts the software engineering methodology to attend to domain problems. It conducts root cause analysis to identify the characteristics and ontology development criteria for specified domains. In the context of this paper which is designing the ontology for Islamic banking product; the peculiarities of the Islamic banking businesses demand several crucial characteristics to be taken into account in the development of the ontology. The characteristics relate to the business model, diverse roles, distinct concepts, representation rules, reuse of other ontologies and varieties of disciplines related to the industry. The software engineering approach guides the conduct of root cause analysis to understand the issues faced by the domain to be put up as requirement specification of the ontology.

KEYWORD: *Ontology, Ontology Development, Software Engineering, Islamic Banking*

1.0 Introduction

Domain ontologies are related to particular domains, and consist of concepts used to describe the domain knowledge and the relationships between those concepts (Hadzic, Wongthongtham, Dillon & Chang 2009; Yu 2011). Development of domain ontologies is not an easy and straightforward task; the team of knowledge engineers and domain experts assigned to develop to develop the ontologies must ensure that the concepts in the domain of discourse, their properties and relations are accurate. As complexities of the world increases, there is need to study the particular aspects that are peculiar to the specified domains. Hence, the approach undertaken for the development of the ontology should be emphasised.

There are many methods of developing ontologies; they could be built using the traditional Ontology Development Life Cycle (ODLC) such as TOVE (Gruninger & Fox 1995), ENTERPRISE (Uschold & King 1995) METHONTOLOGY (Fernandez-Lopez et al. 1997), and DILIGENT (Pinto et al. 2004) or by adopting a software engineering approach. Software engineering was not initially meant to cater for software artefacts such as the ontology. Yet, to develop domain ontologies that are reliable, long lived, and continually adapted and improved; the software engineering approach would be recommended (Annamalai & Rosli 2012). There are several methods of software engineering that has been adapted for the development of ontologies.

During this period of economic volatility, it is important for domain ontologies to meet the specific needs of the businesses of the domain and attend to the prevailing issues and problems faced by them. The domain ontologies should be specifically built to attend to prevailing issues faced by the specified domains. Apparently there were grievances against the Islamic banks that relates to the customers' understanding of the Islamic banking products and processes (Latiff et al. 2015), and the use of appropriate ontology by information systems is deemed as an avenue for improving knowledge dissemination of the domain. The Sofware Engineering Approach for Domain Ontology Development was adopted to develop the ontology for the Islamic banking domain.

This paper looks at the Software Engineering Approach for Domain Ontology Development. In section 2, several methods of software engineering approach for ontology development are discussed to explain the need for a software engineering approach for domain ontology development. Section 3 provides the methodology which describes the phases of the Software Engineering Approach for Domain Ontology Development. Section 4 discusses the ontology produced by the approach and the final section summarises the outcome of ontology development method.

2.0 Related Works

Software engineering put emphasis on systematic processes and standards, integrative tools, and established management and organisational methods in software development (Annamalai & Rosli 2012). It is important to take note that software is not confined to computer programs, but includes all associated documentation and configuration data that is required for the programs to operate correctly. Ontology is not excluded as it defines the representational terms of the data (Gruber 1993b). Being software artefacts, ontologies are best to be developed using the software engineering approach like any other software components in order for the ontologies to be reliable, long lived, and continually adapted (Annamalai & Rosli 2012). Nevertheless, there is need to adapt the software engineering approach for ontology development to take into account the peculiarities of the characteristics and criteria of the domain involved.

There have been several approaches that have been undertaken by ontology developers to adapt software engineering approaches for ontology development; among them are as follows:

a) Unified Process for ONtology (UPON)

The UPON is an approach for large-scale ontology development that adopts the Unified Process (UP) and the Unified Modeling Language (UML) in its development (De Nicola et al. 2009). It is not meant for generic domain ontologies, but is use-case driven and aims at developing well-defined application areas. Following the UP approach, the UPON consists of cycles, phases, iterations and workflows. Every cycles are made of inception, elaboration, construction and transition phases. The phases are subdivided into iterations and each iterations comprise of requirements, analysis, design, implementation and test workflows. The workflows for the UPON methodology are as below:

i) *The requirements workflow* - which specifies the semantic needs and user view of the knowledge to be encoded in the ontology. It involves the determination of the domain of interest and the scope of the ontology, defining it's business purpose, writing one or more storyboards, creating application lexicons, identifying competency questions and modeling use cases.

ii) *The analysis workflow* - which refines and structures the ontology requirements. This workflow acquires domain resources and builds domain lexicon, builds the reference lexicon, models the application scenario using UML diagrams, and builds the reference glossary.

iii) *The design workflow* - which gives an ontological structure to the set of glossary entries gathered in the reference glossary. It involves the tasks of modeling concepts into primary categories (business actor, business object and business process) and also complementary categories, and then model concept hierarchies and domain-specific relationships.

iv) *The implementation workflow* – which encodes the ontology into a rigorous formal language. In selecting the formal language, it is necessary to consider it's expressive power, computational complexity and level of acceptance.

v) *The test workflow* – which tests the ontology to verify the semantic and pragmatic quality of the ontology. The semantic quality concerns the absence of contradictory concepts, which the pragmatic quality refers to the ontology content and it's usefulness to the users.

b) Enterprise Strength Ontology Engineering (EsOE)

The Enterprise Strength Ontology Engineering (EsOE) by Annamalai and Rosli (2012) adopts the value-added activities of Rational Unified Process (RUP) model, Agile model, Capability Maturity Model Integration (CMMI) staged model and IEEE 1074-1995 standard (IEEE) to produce a project-oriented software engineering approach for ontology development. The methodology is structured into three levels of realization as below:

i) *Level I Engineering process* - which focuses on the development processes. The processes involved at this level are problem exploration, infrastructure allocation, requirements management, design, development, testing, deployment, maintenance and retirement of the ontology.

ii) *Level II Project-focus process* - which concerns the management of the Engineering. The processes involved are project management, risk management, supplier agreement management, configuration management, product and process quality assurance, and documentation.

iii) *Level III Organisation-focus process* - which establishes the capabilities of the performed activities. This level comprises of the processes of training, organisational process definition, process measurement and analysis, and also organisational process improvement.

c) Simplified Agile Methodology for Ontology Development (SAMOD)

Like the software engineering domain, agile methodologies were also proposed for ontology engineering. Peroni (2016) introduced the Simplified Agile Methodology for Ontology Development (SAMOD), which is partially inspired by the Test-Driven Development process in Software Engineering. The methodology that targets the development of ontologies with limited amount of ontological entities is organised by the following iterative steps:

i) *Step 1: define a new test case*. Ontology engineers and domain experts work together to collect information and write motivating scenarios of the specified domain. With the motivating scenarios, sets of informal competency questions will be identified. Modelets will be developed according to the scenarios, informal competency questions and the glossary of terms.

ii) *Step 2: merge the current model with the modelet*. The modelets will be merged by adding all the axioms and then collapsing semantically-identical entities. The models will then be test queried according to their formal requirements.

iii) *Step 3: refactor the current model*. Finally the model is refactored and tested. In doing so, the ontology engineers should reuse existing knowledge, document the ontology and take advantages from technologies.

d) Lightweight Methodology for Rapid Ontology Engineering (UPON Lite)

The Lightweight Methodology for Rapid Ontology Engineering (UPON Lite) by De Nicola and Missikoff (2016) derived from the Unified Process for ONtology (UPON). With UPON Lite, ontologies are constructed by domain experts and ontology engineers only intervene to deliver final ontology formalisation. The domain content is elicited, organised and validated by domain experts. The methodology is organised as below:

i) *Step 1: Domain terminology*. This step involves the creation of domain-specific terminology to produce a domain lexicon. Resources such as textual documents, directories, dictionaries, taxonomies, standards and also ontologies could be use to develop the lexicon.

ii) *Step 2: Domain glossary*. Upon the production of the lexicon, this next step is to enrich it with textual description of each entry. An ontology structuring method called OPAL (Object, Process, Actor modeling Language) is adopted to group the concepts into three main categories (i.e., object, process and actor) and also three auxiliary categories (i.e., complex, atomic and reference properties). Synonyms are also identified to pinpoint the preferred term and label the rests as other terms.

iii) *Step 3: Taxonomy*. The lexicon and glossary of the previous steps were used to build the taxonomy.

iv) *Step 4: Predication*. This step concentrates on the properties to characterise the entities of the domain.

v) *Step 5: Parthood (meronymy)*. This step looks at the architectural structure of business entities or parts of composite entities to create hierarches based on "part of" relationship.

vi) *Step 6: Ontology*. From the previous five steps, the ontology engineers could finally produced the ontology.

As compared to the ontology development methods above, the Software Engineering Approach for Domain Ontology Development is targeted for specified domains and focuses on the problems of the domain that requires the development of the ontology as a solution. It is aimed to attends to the prevailing problems of the specified domains.

3.0 Methodology

The Sofware Engineering Approach for Domain Ontology Development was adapted from the Level I (Engineering process) of the Enterprise Strength Ontology Engineering (EsOE). The phases of the domain ontology development adapted from the software engineering approach are as follows:

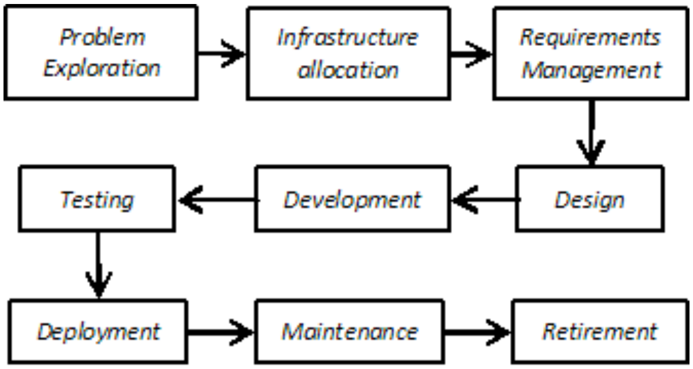

Figure 1. Ontology Development Based On Software Engineering Approach

In meeting the needs of the domain, it is important that the ontology developed be implicitly embedded with the purpose and reasons why it was designed. Hence, the activities in the problem exploration and requirements management phases are designed to identify the root causes of the problem and meet the purposes of the ontology. In order to do so, the problem exploration stage includes root cause analysis of the problem that the ontology intended to attend. The outcome of the root cause analysis shall be studied in order to produce the characteristics and development criteria of the ontology requirement specification. The ontology developed according to the defined specification would be deemed as fit to attend to the prevailing issues of the specified domain. In the case of the ontology development for the Islamic banking industry, the ontology was supposed to attend to the customers' grievances against the Islamic banks. The root cause analysis conducted produced the characteristics and development criteria of the Islamic banking ontology (Latiff et al. 2016)

3.1 Problem Exploration

The problem exploration which is equivalent to software specification or also referred to as requirements engineering is a process whereby the details of the services required from the software is being understood and defined (Sommerville 2011). The activities primarily transform the stakeholders' needs into ontology requirements; which include the process of requirements elicitation and analysis. During this process, the requirements of the ontology should be derived through observation of the problems of the domain that ignited the goals and purposes for the ontology development. The purpose and the intended use of the ontology should be made clear before the design and development of the ontology could take place (Uschold & Gruninger 1996). In order to do so, the intended users of the ontology ought to be identified beforehand and their characteristics be ascertained in order to clarify the purpose and scope of the ontology. It could be done through brainstorming, interviews, questionnaires, text analysis and inductive techniques (Pinto & Martins 2004). Besides identifying the users, the root causes of the problem should also be identified to clearly understand the services that the ontology would cater for and other pertinent matters related to it.

Although it is being unusual for root cause analysis to be conducted upfront during specification stages of software engineering as it was commonly meant to prevent software failures and attend to defects (IEEE 2006; Team 2010), it would be undeniably valuable to understand the problems and needs of the domain. As such, it would be advantageous that root cause analysis be conducted in building the specification for ontologies. Clear understanding of the root cause is crucial to comprehend the domain's problems and the result of root-cause analysis will in turn help to enhance the specification requirements of the ontology. It is only with clear identification of the users of the ontology and understanding of the problem that will make the requirements clearly elicited. A proper elicitation of requirements will facilitate analysis of them. Eventually, the analysis will enable the ontology to meet the user needs, which will in turn improve the quality of the software program and application which utilise the ontology.

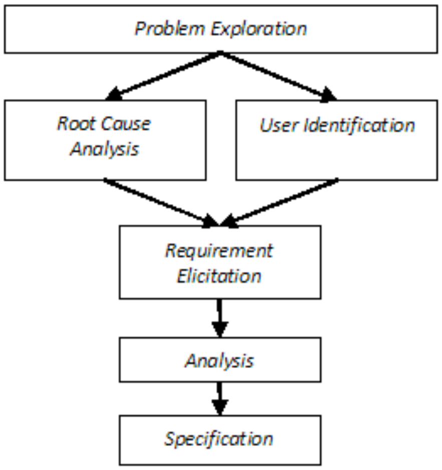

Figure 2. Problem Exploration Stage

The proposal for the development of the ontologies should also describe the motivating scenarios, together with the set of intended solutions to the scenario problems as recommended by Gruninger & Fox (1995). From the given scenario, a set of queries that need to be answered by the ontology could be raised. They become the informal competency questions that need to be expressed in the formal language of the ontology later.

3.2 Infrastructure Allocation

During this phase, the work environment for the ontology development is established. The project team is familiarised with the development environment, which include the processes and tools for knowledge engineering. For the development of the ontology for the Islamic banks, the processes and tools for knowledge engineering of the research, have been predetermined to be using open-source ontology editor, *Protégé* and the enterprise data unification platform *Stardog* respectively.

3.3 Requirements Management

The requirements management include the activities of managing the technical and non-technical requirements of the ontology, and also to identify and resolve the inconsistencies among the requirements (Annamalai & Rosli 2012). The technical requirements are properties or attributes of the ontology to be developed (Team 2010). Meanwhile, the non-technical requirements are requirements that affect the ontology development that are not properties of the product or service (Team 2010).

This stage comprises of:

a) Requirements specification, which is the activity that translates the information gathered during the analysis into a requirements document. It should be decided whether to produce either an informal, semi-formal or formal ontology specification document using intermediate representations (Fernandez-Lopez et al. 1997). The Software Requirement Specification Document for the ontology development was slightly modified from those recommended for other software components (Suárez-Figueroa, Gómez-Pérez & Villazón-Terrazas 2009) as below:

Table 1. Ontology Requirement Specification

	Islamic Banking Product Ontology Requirement Specification Document
1.	Purpose
	The primary purpose of building the ontology is to provide information to enable the banks' customers to have sufficient understanding for their decision in selecting a product. The ontology could also useful to bring explicit information on Shariah concepts and assist bank staff in conceptualising a proposed new product and testing the features of the proposed product during product developments.
2.	Scope
	The ontology focuses on the domain of Islamic banking products of financings and deposits, especially for the personal/ consumer target market. It includes business premise financings, but excludes large corporate financing, trade financing, negotiable certificate of deposits, and other corporate-targeted products. As the ontology of the Islamic banking products is intended merely to assist customers in their decision-making, it need not be a formal ontology, although there should be some minimal rules and axioms to cater for the complexities of the Islamic banking knowledge.
3.	Implementation Language
	No specified formal representative language, but the Web Ontology Language (OWL) would be used for validation purposes.
4.	Intended End-Users
	i) Customers and prospective customers of the Islamic banks, ii) Staff of the bank who are involved in new product development.
5.	Intended Uses
	i) Publishing of Product Disclosure Sheet for disseminating of information on Islamic banking products, ii) Displaying product comparison in assisting the customers to select the Islamic banking products, iii) Displaying the features and testing new Islamic banking products during new product development.
6.	Ontology Requirements
	a. Non-Functional Requirements
	Characteristics of the Ontology The **general characteristics** of the ontology for Islamic banking are as follows: i) The concepts to be dealt with in the Islamic banking industry are distinct ii) The rules in the representation of the concepts in the Islamic banking system

The **specific characteristics** of the ontology for Islamic banking are as follows:

i) The business model or Shariah concept adopted by the Islamic banking product
ii) The diverse roles played by the banks and their customers
iii) The differences of opinion of different juristic schools
iv) The borrowings of Arabic terminologies

Development Criteria to be Fulfilled by the Ontology

i) Clarity

The clarity of the contents should be made just adequate for the customers to make effective comparison among the different products of the Islamic banks.

ii) Concise

The ontology should emphasise on brevity and avoids redundant definitions and explanations that could be counter-productive and might end up confusing the customers.

iii) Coherence

a) Individual Conceptual Integrity

The individual conceptual integrity need to be verified to ensure that there is no contradiction in the interpretation of a concrete concept pertaining to the entities it represents.

b) Collective Consistency

The collective consistency should be analysed to verify that the relationships that bind the concepts reflect the dependencies between their corresponding entities.

iv) Extensibility and Expansibility

The designed ontology need to be ensured as much as possible that the hierarchies in the ontology is diversified in order to increase the power provided by multiple inheritance mechanisms.

v) Minimal Encoding Bias

The ontology should be close to a natural textual form and independent of certain symbol-level encoding.

b. Functional Requirements

The ontology should be able to answer at least the following competency questions:

1. What are the Shariah concepts adopted in the products?
2. What is the Shariah source referred to by the products?
3. What is the scholarly opinion for the product?
4. What is the translated meaning of certain verse in the source of reference that relate to the Islamic banking?
5. What is the market segment of the product?

	6. What is the minimum asset value for the product? 7. What is the maximum asset value for the product? 8. What is the maximum income eligible to apply for the product? 9. What is the minimum age for the customer to apply for the product? 10. What is the maximum age for the customer to apply for the product? 11. What is the customer's role in the contract? 12. What is the bank's role in the contract? 13. What is the type of commodity dealt with by the product? 14. What is the profit rate of the product? 15. What is the margin of financing of the product? 16. What is the maximum tenure for the contract? 17. What is the minimum tenure for the contract? 18. Does the product require any collateral? 19. Does the product require processing fee? 20. What is the maximum annual fee to be incurred on the product? 21. What is the minimum annual fee to be incurred on the product? 22. What is the takaful requirement of the product? 23. Does the product require a security? 24. What is the ta'widh (compensation) charge for overdue account? 25. What is the lock-in duration of the product? 26. Does the contract incorporate clauses of the bank's right to set-off with other accounts? 27. What is the form of return that the deposit product provides? 28. Does the product provide grace period for payment? 29. Does the product provide payment holiday? 30. What is the Arabic word for a specific term? What is the defined meaning used in the Islamic banking?

For the purpose of ontology development, the non-functional ontology requirements refer to the development criteria, qualities or general aspects that are not related to the ontology content (Suárez-Figueroa et al. 2009). On the other hand, the functional ontology requirements are the content specific requirements, which refer to the particular knowledge to be represented by the ontology (Suárez-Figueroa et al. 2009). The functional ontology requirement should include the characteristics and criteria related or required in the ontology contents, which derived from requirements analysis in the problem exploration stage. The characteristics of the ontology refers to the distinctive nature of the concepts handled by the ontology. They are divided into general characteristics which should also be present in other domains and the specific characteristics that are peculiar for the particular domain of concern. Meanwhile, the development criteria of the ontology are the generally accepted quality of the resulting ontology (Annamalai 2005; Gruber 1993a).

The characteristics of the Islamic banking products include the distinct concepts dealt by the Islamic banks, the rules in the representation of the Islamic banking concepts, the diverse Shariah concepts adopted by the Islamic banking products, the diverse roles played by the banks and their customers, the differences of opinions among the Islamic scholars, and the borrowings of Arabic terminologies by the Islamic banking concepts. Meanwhile, the criteria of the ontology involve the clarity, conciseness, coherence, extensibility and expansibility, minimal encoding bias, epistemological completeness and also the competency of the ontology.

b) Requirements validation checks the requirements for realism, consistency and completeness. The inconsistencies among the requirements were identified and resolved (Annamalai & Rosli 2012). Errors in the requirements documents were checked and be corrected (Sommerville 2011). The activity of the requirements validation need not be carried out in strict sequence after the requirements specification, but could be interleaved (Sommerville 2011).

3.4 Design

This stage involves the designing of the model for the selected domain. The description of the ontology will be built in a conceptual model in order to meet the specification defined. The conceptual model consists of concepts in the domain and relationships among those concepts are made using of conceptual models, such as from informal sketchy models, binary relations diagram or concept dictionary. With the terminology of the ontology defined, the informal competency questions could then be defined formally as an entailment or consistency problem pertaining to the axioms of the ontology (Gruninger & Fox 1995).

The design of the ontology is made with the requirement specifications in hand. It involves translating the characteristics of the ontology that has taken into account any complexities of the domain and also the criteria of the ontology. The intention for the designing activity is to structure the domain knowledge in a conceptual model that describe the problem and its solution in terms of the domain vocabulary identified during the ontology specification activity (Fernandez-Lopez et al. 1997). It involves identification of the key concepts and the relationships in the domain of interest, production of precise unambiguous text definitions for such concepts and relationships, and identification of terms to refer to such concepts and relationships as highlighted by Uschold & King 1995.

The steps of the design involve conceptualisation activity of the ontology, which are as follows:

a) Building a Glossary of Terms (GT)
The terms include concepts, instances, verbs and properties pertaining to the domain are gathered and listed. The terms could be identified from documents such as procedural manuals, guidelines, text books and articles related to the domain. The lists were expanded as the ontology development progresses. For the Islamic banking product ontology, the terms were gathered from the Ontology Requirement Specification Document, the Bank Negara Malaysia Guidelines on Product Transparency and Disclosure, and also from text books and articles on Islamic banking.

b) Grouping the terms as concepts and verbs

The terms are then grouped into concepts and verbs. Each set of concepts will be group in accordance to the sub-disciplines of the domain so that the concepts that are closely related to each other. Similarly, the verbs will also put in sets that are related to each other.

The ontologies for a particular domain could also reuse other common ontologies. Such reuse will involve fusion and integration of existing ontologies.

3.5 Development

The development stage involves the transformation the design model into components, and integrates the components into a complete ontology. The conceptual descriptions are transformed into formal models. Concepts are defined through axioms in order to restrict the interpretations of those concepts. As recommended by Fernandez-Lopez et al. (1997) and also Uschold & King (1995), definitions that have already been built into other ontologies will also reused.

The ontology is then encoded in an appropriate representation language. The coding will involve committing to the basic terms that will be used to specify the ontology (e.g. class, entity, relation), choosing a representation language, and writing the code. The Islamic Banking Product Ontology was encoded into the Web Ontology Language (OWL). OWL is a standard for coding of ontology and Semantic Web documents which were developed by the World Wide Web Consortium (W3C). It enables the coding of classes, properties, individual, and data values, and can be used with information written in RDF (Resource Description Framework).

3.6 Testing

The ontology need to be tested to ensure that its appropriateness and the definition of the terms should be semantically coherent (Fernandez-Lopez et al. 1997). Axioms could be used to define the semantics or meaning of the terms (Gruninger & Fox 1995). They shall specify the definitions of the terms in the ontology and the constraints on their interpretation, guided by the formal competency questions. The ontology will then be validated to ensure that it meets the specified requirements. It should conform to its specification and meets the expectations of its users (Sommerville 2011).

According to the Ontology Requirement Specification in Table 1, the ontology for the Islamic banks need to be tested for its non-functional and also functional requirements.

***i)* Testing of Non-Functional Requirements**

The ontology need to be ensured that it will meet their characteristics and the development criteria.

a) *Characteristics of the ontology*

The **general characteristics** of the ontology were ensured to be as follows:

- The ontology need to include the distinct concepts of the Islamic banking industry, especially the Shariah concepts and also other distinct Islamic banking concepts such as *rabbul-mal*, *mudharib*, *khiyar*, *ta'widh*, *muqasah* and many more.

- The ontology of the Islamic banking products are meant to facilitate understanding on the banking products and processes by the customers so that they could make informed decisions. They are not targeted for use in automating the banking transaction or processes, but meant for facilitating their selection of banks and product. As such, full axiomatization of the concepts and relations in the ontology may not be a necessity, except for some concepts that need to be axiomatised to technically clarify their definitions. Hence, the rules in the representation of the concepts in the Islamic banking system were included in the ontology.

The ontology also need to be ensured that it include the **specific characteristics** meant for the ontology for Islamic banking which are as follows:

- The ontology need to be ensured to include the business models or Shariah concept commonly adopted by Islamic banking products.

- The ontology was ensured to include the diverse roles played by the banks and their customers. Visualisations made using the graphic tool of the ontology editor, Protégé shown that the ontology was able to cater to the different contractual roles based on their respective Shariah concepts.

- The ontology must take into account the differences of opinion of the different juristic schools.

- The ontology should also take into account the borrowings of Arabic terminologies by the Islamic banks.

b) *Development Criteria*

The development criteria to be fulfilled by the ontology

- *Clarity*
  The clarity of the contents should be made adequate for the customers to make effective comparison among the different products of the Islamic banks. As described by Gruber (1993), definitions should be provided in natural languages in order to provide clarity.

- *Conciseness*

  The ontology need to emphasise on brevity and avoids redundant definitions and explanations that could be counter-productive and might end up confusing the customers. According to Bezerra, Costa and Okada (2009), the ontology is considered as concise when it does not store unnecessary definitions, when there is no redundancy between terms definitions, and redundancies are not inferred by other definitions.

- *Coherence*

  The ontology in its natural language documentation should be defined coherently; whether individual or collectively. For its individual conceptual integrity, the definitions of the ontology were ensured were based from a single, yet reliable definition. In the development of the Islamic banking ontology, the definitions were based on documentation by ISRA (2012) so that there is no contradiction in the interpretation of the concrete concept pertaining to the entities represented. The collective definitions should also be made consistent to reflect the dependencies between their corresponding entities.

- *Extensibility and Expansibility*

  The ontology need to be ensured that the hierarchy in the ontology was diversified in order to increase the power provided by multiple inheritance mechanisms. The concepts in the ontology, such as the class: Business below, were diversified in order to allow to be extendable and expandable in the future.

- *Minimal Encoding Bias*

  In order to ensure that the ontology be close to a natural textual form and independent of certain symbol-level encoding, it was built in textual natural language. Its codification into .owl was only made for the purpose of testing and evaluation.

- *Standardisation of Names*

  Names of concepts or relations in the ontology were standardised whenever possible and as much as it could to adopt the terms commonly used in the Islamic banking industry.

- *Ontological Completeness*

  The ontology is required to produce the information needed by customers for their decision making purposes. As such, the ontology need to be tested for completeness based on the requirements of the Product Disclosure Sheet of the Bank Negara Malaysia and other pertinent information on the products as listed in the competency questions in Table 1.

- *Competency*

  The ontology need to be tested for its competency to at least answer or provide basic information to fill up the Product Disclosure Sheets of the Bank Negara Malaysia as listed in Table 1.

ii) Testing of Non-Functional Requirements

The testing of non-functional requirements need to be conducted with the use of competency questions. This was to ensure that the ontology meets the purpose of its design. The testing could be done by querying the ontology with the competency questions using the SPARQL Protocol and RDF Query Language (SPARQL) query.

3.7 Deployment

During this stage, the ontology is integrated with the application software in order to realise its planned use.

3.8 Maintenance

The maintenance involve the management of solution of faults and failures in the ontology-based application. It include updates and corrections of the ontology when needed due to the necessities of the existing use and also reuse in related applications.

3.9 Retirement

This is the phase when an existing ontology is removed from its active environment.

4.0 Results and Discussion

Using the Software Engineering Approach for Domain Ontology Development, the Islamic banking ontology produced comprises of 404 classes, 52 object properties and 66 data properties; thus producing 2,323 axioms.

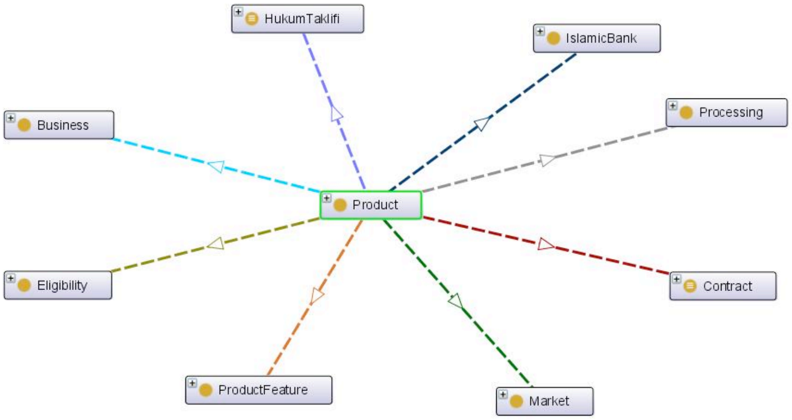

Figure 3: Ontology for Islamic Banking

Resulting from the outputs of the competency questions as listed in Table 1, the ontology is able to be queried to disclose the products' compliance with the Shariah according to the different juristic school and also to produce detailed information on the banking products. This will eventually facilitate the customers to understand the products and enable informed decision making.

The Software Engineering Approach for Domain Ontology Development was adapted from the Engineering Process (Level I) of EsOE. Yet, EsOE does not conduct root cause analysis as a method to clearly identify domain problems. As compared to UPON, SAMOD and UPON Lite where the ontology specifications were derived from use-cases or test-cases, the Software Engineering Approach for Domain Ontology Development identifies its specifications from root cause analysis. The analysis had enabled the identification of the characteristics and development criteria of the ontology which constitutes the non-functional requirements of the ontology.

5.0 Conclusion

In order to provide a reliable, long lived, and continually adapted ontology, it is best that domain ontologies be developed using the software engineering approach. However, the ontologies are dependent on the characteristics of the domain and criteria needed to attend to their problems. The Software Engineering Approach for Domain Ontology Development is able to adapt to the peculiarities of particular domains. With the conduct of the root cause analysis, ontology engineers will clearly understand the issues faced by the domain which will then be put up as requirement specification of the ontology.

References

Annamalai, M. (2005). Formative evaluation of ontologies for information agents. In Conference in Computer Science, Technology and Networking, Shah Alam.

Annamalai, M., & Rosli, M. M. (2012). Software Engineering: A Pathway to Enterprise-Strength Ontology Engineering. *International Journal of Digital Content Technology and its Applications,* 6(22), 98.

Annamalai, M., & Sterling, L. (2003, July). Guidelines for Constructing Reusable Domain Ontologies. In OAS (pp. 71-74).

De Nicola, A., Missikoff, M., & Navigli, R. (2009). A software engineering approach to ontology building. Information systems, 34(2), 258-275.

De Nicola, A., & Missikoff, M. (2016). A lightweight methodology for rapid ontology engineering. *Communications of the ACM*, 59(3), 79-86.

Fernandez-Lopez, M., Gomez-Perez, A., Juristo, N. (1997) METHONTOLOGY: from ontological art towards ontological engineering. In: Proceedings of AAAI'97 spring symposium series, workshop on ontological engineering, Stanford, CA, page no. 33-40.

Gomez-Perez, A., Corcho, O., Fernandez-Lopez, M. (2003) *Ontological engineering: with examples from the areas of knowledge management.* E-commerce and the Semantic Web, Springer-Verlag, UK. 2003, pp. 125-140.

Gruber, T.R. (1993a). “Toward Principles for the Design of Ontologies Used for Knowledge Sharing (Revision: August 23, 1993)”, *In International Journal Human-Computer Studies* 43, p. 907-928.

Gruber, T.R. (1993b), A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* 5, 199-220, 1993.

Gruninger, M., & Fox, M. S. (1995). Methodology for the Design and Evaluation of Ontologies.

Guarino, N. (1998). Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy (Vol. 46). IOS Press.

Hadzic, M., Wongthongtham, P., Dillon, T., Chang, E. (2009) Introduction to Ontology, Studies in Computational Intelligence, Volume 219, *Ontology-Based Multi-Agent Systems*, 2009, Pages 37-60.

IEEE (2006). IEEE Standard for Developing a Software Project Life Cycle Process. IEEE-SA Standards Board, The Institute of Electrical and Electronics Engineers, Inc. (IEEE), New York, 2006.

Latiff, A. S. A., Haron, H., & Anamalai, M. (2015). Grievances on Islamic Banks: A Survey. In *Islamic perspectives relating to business, arts, culture and communication* (pp. 295-306). Springer Singapore.

Latiff, A. S. A., Haron, H., & Annamalai, M. (2016, August). Characteristics and development criteria for Islamic banking ontology. In Information Retrieval and Knowledge Management (CAMP), 2016 Third International Conference on (pp. 136-142). IEEE.

Peroni, S. (2016, November). A simplified agile methodology for ontology development. In *International Experiences and Directions Workshop on OWL* (pp. 55-69). Springer, Cham.

Pinto, H.S. & Martins, J.P. (2004). Ontologies: How can They be Built? *Knowledge and Information Systems* (2004) 6: 441-464. Springer-Verlag London Ltd.

Pinto,H.S., Staab, S., Tempich, C. (2004). Diligent: Towards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies. In ECAI, pages 393–397, 2004.

Sommerville, I. (2011). Software Engineering, 9th Edition. Pearson Addison-Wesley, Boston, 2011.

Studer, R., Benjamins, V.R., Fensel, D. (1998). Knowledge Engineering: Principles and methods, Data & Knowledge Engineering 25 (1998) 161-197.

Suárez-Figueroa, M.C., Gómez-Pérez, A., Villazón-Terrazas, B. (2009) How to write and use the Ontology Requirements Specification Document. In Proceedings of ODBASE 2009. On the OTM 2009. LNCS 5871. Pp: 966-982. Vilamoura, Algarve-Portugal. November 2009.

Team, C.P. (2010), CMMI for Development, v.1.3., Technical Report, CMU/SEI-2010-TR-033, Carnegie Mellon Software Engineering Institute.

Uschold, M., Gruninger, M. (1996). Ontologies: Principles, methods and applications. *Knowledge engineering review*, 11(2), 93-136.

Uschold, M., & King, M. (1995). *Towards a methodology for building ontologies* (pp. 15-30). Edinburgh: Artificial Intelligence Applications Institute, University of Edinburgh.

Yu, L. (2011). *A developer's guide to the semantic Web*. Springer Science & Business Media.