Proceedings of the Pakistan Academy of Sciences 52 (3): 187–200 (2015)

ISSN: 0377 - 2969 (print), 2306 - 1448 (online)

Pakistan Academy of Sciences

Research Article

An Ontology-based Approach for Handling the Issues in Requirement Engineering

Muhammad Ahsan[1*], Yasir Hafeez Motla[2], and Muhammad Waqar Azeem[3]

[1]Zarai Taraqiati Bank Limited, Islamabad, Pakistan
[1, 2, 3]UIIT-PMAS Arid Agriculture University, Rawalpindi, Pakistan

Abstract: Requirement engineering (RE) has a vital role in the whole software development life cycle. It is a proven fact that stakeholders involved in the requirements elicitation and specification process may have different backgrounds and source; this may result in inconsistencies, ambiguities in requirements due to lack of domain knowledge, conflicting and contradictory views, communication and coordination issues. A few examples in the implementation of software systems, reciprocally the resulting system did not fulfill stakeholder expectations. To deal with these problems, it is necessary that RE activities ought to be integrated and improved by higher domain, application and instance level knowledge of requirements, which must be able to facilitate understanding about the context and develop shared understanding, amongst stakeholders. This study aimed at evaluating the framework of knowledge management to address mentioned issues within the RE process.

Keywords: Requirement engineering; RE challenges; ontology; knowledge management; formalization; RE process improvement.

1. INTRODUCTION

Requirements engineering (RE) is the first phase of software engineering and considered the most crucial and essential part of the entire software development life cycle. According to "Object Watch" report published in 2010, the software failure cost has been reached to about $ 6 trillion or $500 billion per month and this trend continued further; some research has concluded that systems failure in approximately ninety percent (90 %) of large software products traced back is due to poor requirements elicitation and specification. In short, the poor requirement engineering activities are the major reason of project failure [1, 2, 3]. The RE domain holds certain challenges, especially in elicitation and analysis, which are lack of communication and coordination, incomplete and contradictory knowledge of requirements, conflicting views of stakeholders a few examples [4]. One of the major reasons for project failure is the incomplete, insufficient understanding and management of requirements. The designer of information system begins designing system too early before understanding the customer need. The cost of correcting the error after delivery stage has been higher than the cost of correction during analysis phase [5]. Software and Requirement Engineers usually are not domain experts, therefore it is necessary for them to learn the problem domain because different understanding may results to incomplete and an ambiguous specification. Therefore, all participants have shared understanding about the problem [6]. RE process must elicit and understand the background of requirement knowledge, established common terminology among the diverse stakeholders and develop shared understanding among the stakeholders. Thus, software engineers redesign

Received, December 2014; Accepted, August 2015
* Corresponding author: Muhammad Ahsan: Email: mailahsan@gmail.com

and iterate specification due to lack of information and interpretation [7].

Business knowledge is considered to be useful for Requirement Analyst to thoroughly collect business stakeholders' expectations of the system as initial requirements. It suggests that understanding and modeling the interaction between crucial users can help identifying unknown users, and also assists users to organize their thoughts and ideas purposefully [8]. Communication and coordination are the primary challenges facing global software development (GSD) industry, because there are different stakeholders with different background and sources involved in elicitation and specification process. The deficiency in communication and coordination can harm the relationship and trust, in these situations, knowledge management can be useful to handle these issues in the GSD and knowledge sharing is necessary, keeping the people update and improve shared understanding among members of the team [9]. Requirements can be viewed as statements that capture stakeholder demands, whose understanding requires domain knowledge to help bridge between a stakeholder's design on what a system needs to do and what is practically implementable in that system. The domain knowledge, which also contains rules and assumptions about the system's operating environment, offers us a practical means to identify inconsistencies and overlaps in requirements that may arise from the competing objectives and/or different stakeholders' preferences. Some types of requirement inconsistencies may not be detectable in the absence of such domain knowledge [10]. The weaknesses in the RE processes are misalignment or lack of RE knowledge with actual business process knowledge, misunderstanding, lack of coordination among the stakeholders, communication which are probable risk to bring the inadequate solution [4, 11]. Data information and knowledge are the essential building blocks of information, we exchange these concepts, however, there exists certain distinction among them, data is the assortment of facts in unstructured form and stored in un-organized way. Whereas to make this data meaningful, we process this data, examples are computed, summarized information, most of the information based application relies on this block. While knowledge is something different, it has a capability to link different information and present the meaningful knowledge which is clear in semantics, have a well-defined relationship, understandable by humans and as well as machines.

The comprehensive study shows that there is no practical approach for knowledge management to handle the shortcoming of RE, moreover, most of the literature contributes to develop the theoretical foundation, therefore lacks in providing end to end solution or applied approach. Our research objective was proposing and evaluating the framework of knowledge management for handling the issues in RE.

2. RELATED WORK

Gasevic et al [12] presented the literature review on the use of ontology in different phases of the software engineering life cycle, such as documenting, modeling of domain background, testing, artifacts, interaction / collaboration. They discussed on how ontology helps in solving issues face during the SDLC. Their study provides a good basis to work on the applied ontologies in the areas mentioned in their work. Reyes-Ortiz et al [13] presented knowledge representation in a medical diagnostics domain; their developed ontology cover three major components which are symptoms, diseases and their risk factors. Their paper described detailed usage of ontology, however, they did not described implementation detailed of their proposed model as well as evaluation has not been discussed in their presented work. Kayed et al [14] described the importance of ontology in the RE process for e-government applications. The key objective of their research work was to develop common and important concepts of e-government domain by using different tools, i.e., text-to-onto and developed their own tool KAON's which further refined the concepts. They emphasized that if concepts in a given domain of interest have sufficient and rich semantics, they will develop a shared understanding of the requirements of various e-government applications [14]. Nahar et al [4] presented step-wise refinement model for requirement elicitation and discussed the various issues in elicitation. Their elicitation model contains user request, domain analysis, feasibility study, stakeholder analysis, elicitation techniques and prototyping. They used the traditional approach for elicitation and demonstrated with the case study in Hospital domain. Castaneda et al [7]. discussed the usage of ontology in the requirement engineering life cycle. They elaborated its use in the development of requirement ontology, specification document, and application domain ontology. They presented the literature review of the benefit of applying ontologies in areas which is good for building the theoretical foundation. Reddy et al [15] described benefits of ontologies in developing the common understanding, reuse of domain knowledge, explicit management of the domain assumption. Ahsan et al [16] presented an approach of domain modeling and applied it in an area of agriculture by taking crop as case study; they argued that their proposed work is beneficial for farmers as well as agriculturist to understanding the semantics as well as reducing the inputs for searching. Barforush and Rahnama [17] presented the literature review and performed a comparison of different ontology learning tools to acquire the knowledge from semi structured and unstructured data. In case of ontology construction from the text they discussed various tool like Bole, OLE, Onto-cmap and Text2Onto. In the semi-structured data they discussed the tools, i.e. AEON, RelExt, OntoGen and GALEIN.

The different literature reviews and related study suggest that there is need to develop and demonstrate the model with an applied approach to handle the mentioned issues and challenges with the help of ontology.

3. PROPOSED METHOD & APPROACH

We have proposed a comprehensive approach for dealing with the various mentioned issues affecting the RE process by framework of knowledge management. We have used design science research methodology (DSRM) [18] to demonstrate the proposed model by applying in hospital registration and admission module and evaluated by the professionals to see if the proposed framework and its demonstration can help to develop shared consensus among the stakeholders and beneficial in understanding the problem domain so that developed system may minimize the risk of failure and expectations of stakeholders. We have divided our model into the following phases:

- Requirements Knowledge Acquisition;
- Requirements Knowledge Formalization;
- Persistent Storage;
- Knowledge Distribution; and
- Knowledge Integration.

4. DEMONSTRATION OF PROPOSED FRAMEWORK

Our objective is to develop and evaluate the framework of knowledge management to handle issues faced during the RE process. Fig.13 shows the framework of requirements knowledge management.

4.1 Requirements Knowledge Acquisition

The most important activity after obtaining set of high level requirements is the identification and selection of sources for knowledge elicitation to understand the current business process as well as to find the gaps and problems in current business model. There are different approaches which are used to acquire the domain knowledge from learning objects and non-learning objects. In the case of learning objects, types of input could be structured data, semi-structured data and unstructured data. In case of non-learning objects, human experiences, thoughts are the examples and direct and indirect approaches used to obtain this sort of knowledge [17]. In order to acquire knowledge of the domain we have used both techniques semi-automated as well manual. If the domain is of type of Learning then a lot of material is available in various locations such as internet, Wikipedia and other encyclopedias, to extract such types of concepts and relationships that exist among them in the given domain of interest, learning techniques and tools can be used for obtaining initial set of concepts and then refining these concepts with the help of domain experts. In case of an organization's business process, domain knowledge cannot be obtained from the learning objects because such knowledge is a combination of tacit and explicit which is an organization's intellectual property and hidden in the company manuals, confidential documents, workers experience and in their brain. This type of knowledge can only be obtained from the business stakeholders and documents through direct and indirect approaches and finally knowledge engineer with the help of stakeholders analyze about which type of knowledge should be the part of their knowledge base for implementation of proposed solution or business process re-engineering. The conceptual model is the output of the knowledge acquisition process in the shape of concepts and relationship among them along with their taxonomies. Some of the identified concepts in the given domain of interests are shown in Table 1.

Table 1. Top level concepts and relations.

Domain	Relation	Range
Patients	has Full Name	Literal
Patients	has Registraion No	Registration
Patients	has Identity	Literal
Patients	belongs To	Regions
Registration	paid Amount	Literal
Registration	Belongs To	Patient
Registration	Authorized By	Roles
Registration	Expires On	Literal
Registration	Received Documents	Artifacts
Registration	has Entitlement	Entitlements
Registration	has Panel Associated	Panels
Admission	requires Some	Registration
Admission	has Assigned	Doctor
Admission	has Diagnosed	Disease
Admission	has Alloted	Ward

Since we are going to formalize the Patient registration and admission process, Table 1 illustrates the conceptual model of some of identified classes and relationships among them, which we have extracted during the process of knowledge acquisition. Patient concept describes that each instance of the patient concept has its full name and other bio data, registration number allotted during the process of registration. Patient may or may not associate with the panel. Similarly admission process requires that patient must be registered and have valid registration process attached with him and for each admitted patient Role of doctor must be attached with him who will visit him and admission fee has been paid as initial entry amount. Each admission process is based on some disease diagnosed in the patient etc.

4.2 Requirements Knowledge Formalization

Knowledge acquisition is the first process which knowledge engineer performs by applying different knowledge acquisition techniques. In order to transform the conceptual model into formal model we have followed the process introduced by Noy [19], i.e.:

- Outline the Scope of the system
- Consider reusing formerly established ontologies.
- Extract the key terms in the ontology.
- Defining classes and its taxonomy.
- State the properties of classes, slots.
- Express the facets of the slots.
- Create the instances.

According to the first step, the scope of the system is to manage the application level knowledge of the patient registration and admission process. As we intend to model the part of application domain of the hospital business process, we could not be able to find the existing ontologies in the given domain of interests due to the fact that process is mapped to specific business needs owned by stakeholders and it is the property of the organization.

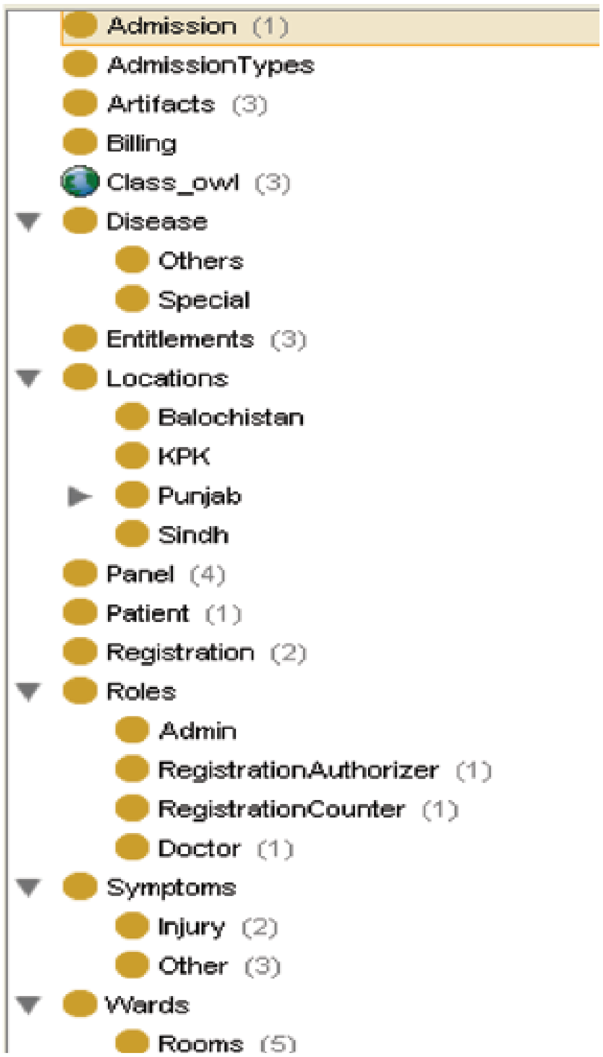


Fig. 1. Knowledge formalization.

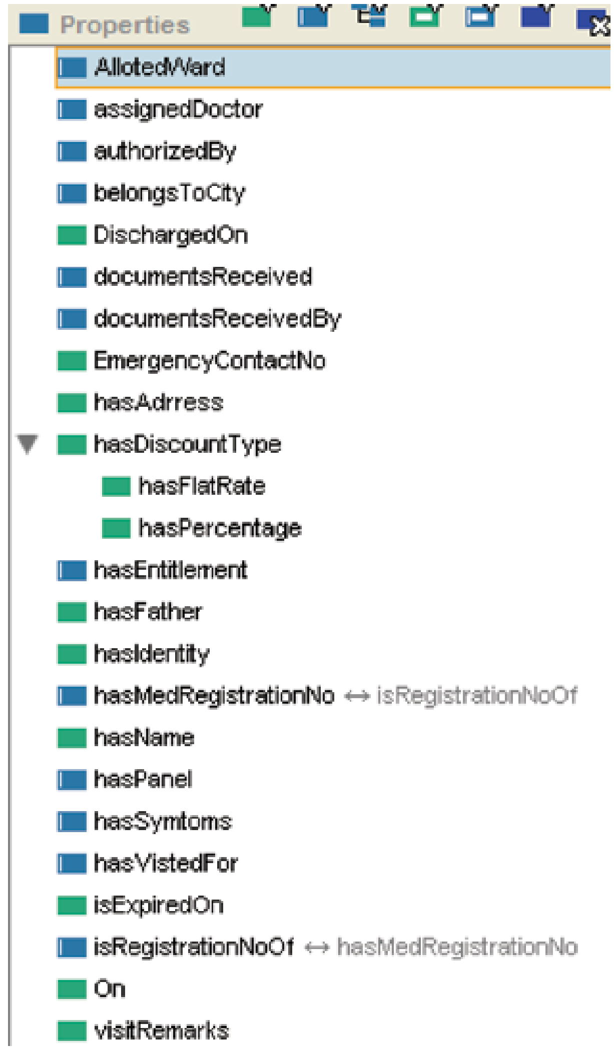


Fig. 2. Object and data properties.

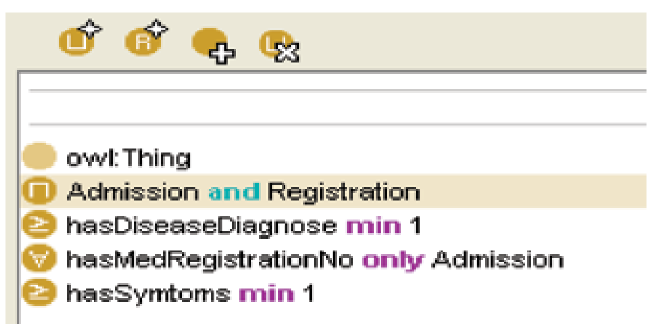


Fig. 3. Constraints.

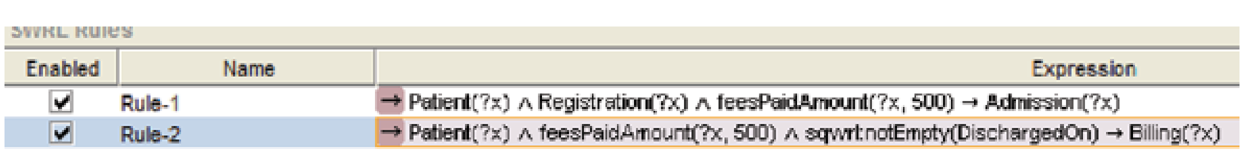


Fig. 4. Object and data properties.

Extraction of important terms is the process of identifying the part of knowledge which is needed to model, an organization's knowledge may be stored in the bundle of manuals, but our intention is to manage the application level knowledge which is consumed by the knowledge based system. So the all stakeholders have shared consensus about the process which needs to be implemented further.

Defining class and class hierarchy is the step which is performed by the knowledge engineer. The protégé ontology editor was chosen for this reason, because it provides strong reasoning and inference capabilities. We have adopted the top-down approach for building the concept taxonomy. Fig.1 illustrates the taxonomy of concepts in the domain of interest.

When the classes and its taxonomy is defined, then the next step is to identify and define the properties among the concepts. A property relates the domain concept with the range. There are two nature of properties which protégé supports to define: 1) An object property which holds the relationship among instances of concepts; 2) while the data property assign the Literal, i.e., string, date and numbers to an instance of concepts. There different types of properties, i.e. functional property for defining one to one relationship an example max one registration is attached with each patient and vice versa is inverse functional, similarly transitive property defined as the patient is linked with registration and registration is associated with admission, so the patient is associated with admission, similarly symmetric property same property name used for functional and inverse functional. Fig. 2 shows a list of properties associated with patient, registration and admission process.

Next step is to define the constraints on the properties, constraint limits the values. While Reasoners help us to ensure that the knowledge, fulfill all the constraints defined. Some of the restrictions are applied to the Admission concept.

Some rules and constraints depend on the conditions like if then else clause. Protégé offers SWRL semantic web rule engine to write more powerful rules and implicit knowledge. A typical example in this scenario is a patient can only get admission if he is having valid registration number and paid initial fees for admission. Similarly if the patient is associated with the panel then discount rate will be applied based on the discount rate defined in the agreement.

Final step is to instantiate the classes to see the actual behavior of the objects. This step is done by Knowledge Engineer which transforms the tacit knowledge and unstructured explicit knowledge in the knowledge base. Fig. 5 shows the instance of the registration and its associated properties while Fig.6 shows the instances of admission class and its associated properties.

A knowledge base is the combination of a T-Box and A-Box and often written as K = (T, A) where T is a set of axioms and A is set of facts. When all the instantiation is complete, it is necessary to check the consistency and completeness of the developed knowledge model if it conform all the constraints and rules. There are certain Reasoners available in protégé i.e. Pellet, Racer which check the consistency and completeness.

4.3 Knowledge Persistence Storage

There is a need for persistent storage of the ontology knowledge model for scalability and better performance and security features where knowledge base may accessible through the user interface. There are certain storage models discussed in the literature.

One way to store semantic data in the form of simple relational data structure approach and then perform simple SQL queries over the knowledge base. There are three main approaches deals with SQL based approach. 1) Triple store consists of three columns (SUB, PROP, OBJ), all data stores in a single Table. 2) N-array where the table is created on the basis of subject and their properties. Each subject and its associated properties are stored in a single Table. 3) Binary Tables where table is based on the properties, so the number of tables is directly proportional to the number of distinct relationships.

Table 2. Experts rating based on "Agreed/Partially Agreed/Not Agreed".

Category	Parameters	Agreed (0-3)	Partially Agreed (4-6)	Not Agreed (7-10)
Framework	Knowledge Management	8	1	1
Framework	Integration Support	9	1	0
Framework	Ease of use	7	2	1
RE Process	Develop Shared Consensus	8	2	0
RE Process	Customer Satisfaction	7	3	0
RE Process	Conflicts/Ambiguities Removed	7	3	0
RE Process	Inconsistencies Removed	7	3	0
RE Process	Improve Communication	9	1	0
RE Process	Improvement in Elicitation and Validation	7	2	1

The described techniques use the pure relational model to store the knowledge and do not have much inference, semantic and reasoning capabilities and it's also nearly impossible to define explicitly constraints and facet on relationships. So a need is raised for providers which can have ability to store semantic data with inference and reasoning support also have capability to embed SWRL rules [20].

A recent improvement in the ontology organization is the storage of semantic data in a shape of the URI so that native SPARQL queries may run over it and it may able to provide semantic capabilities and inference rule support that are not supported in the typical relational model. Virtuoso, Sesame, Oracle, 4store, Allegro Graph, Fuseki are the some of the semantic storage providers [21]. Oracle Database 11g Enterprise Edition, provides built-in provision for RDF/OWL /RDFS/ SKOS principles, this semantic based data storage provider facilitates developers and application programmer to take advantage of a scalable, open, integrated, secure and proficient platform for OWL and RDF-enabled applications. These semantic features of Database facilitate saving, accessing, loading, and DML access to OWL/ RDF data and ontologies, inference using RDFS, OWL and SKOS semantics and user-defined rules. Oracle has introduced the new column type named as SDO_RDF_TRIPLE_S to store the RDF data. Every RDF data model consists a set of triples as subject, object and predicate which are structured as an OWL/RDF graph of direct labeled edges. The edges are called relationship or links that joins a subject node with an associated object node and is labeled as a predicate. The normalized, compressed and partitioned storage architecture manages the complexity arising from repeated usage of typically long URIs and literal values associated with the subjects, objects and predicates across triples. This provides space-efficient storage that requires 75% less disk hardware than uncompressed semantic data, and scalable and perform ant loading, querying, and inference. In addition to that it provides user-defined rules, Fine grained security, indexing of documents, scalability and bulk load operations to import and export OWL/RDF data [22].

4.4 Knowledge Integration

The pattern of knowledge is more complex than data and information, as the knowledge should have the ability to link with other source of knowledge stored, distributed on heterogeneous servers so it must have a homogenous structure, that's the feature of ontology driven knowledge model. Since the develop knowledge model is exportable into RDF, XML, OWL format, it complies W3C recommendations and provide maximum interoperability. The developed knowledge base is in the shape of Triple so SPARQL queries are used to apply to the knowledge base stored as

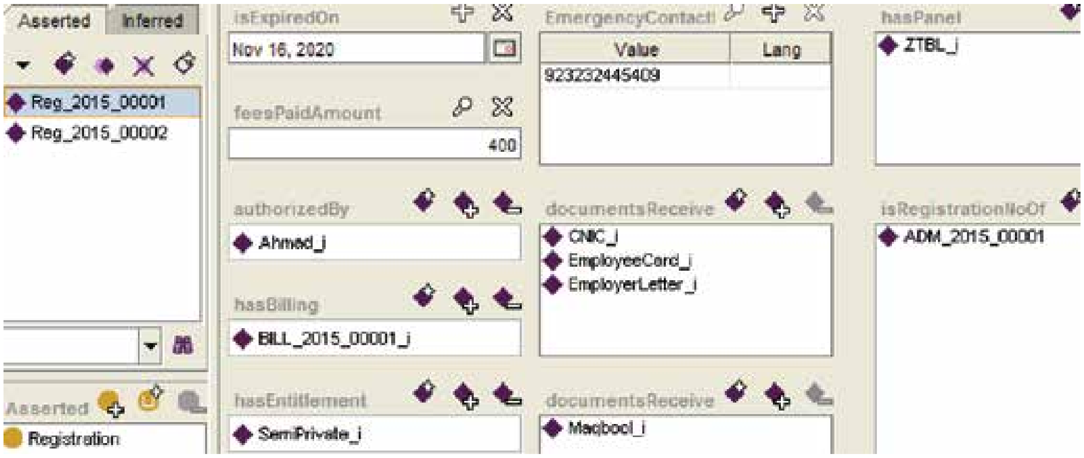


Fig. 5. Instances of registration class.

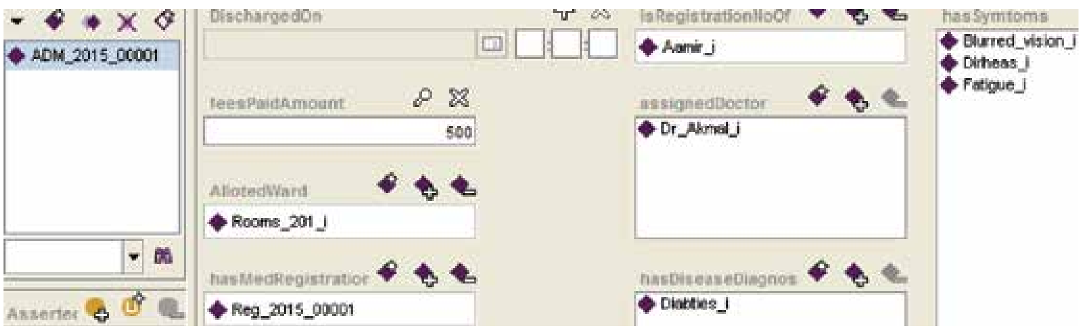


Fig. 6. Instances of admission class.

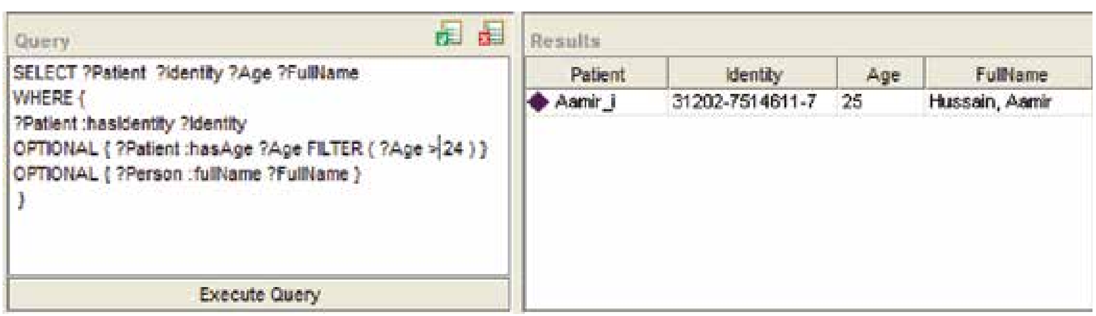


Fig. 7. SPARQL query showing patient.

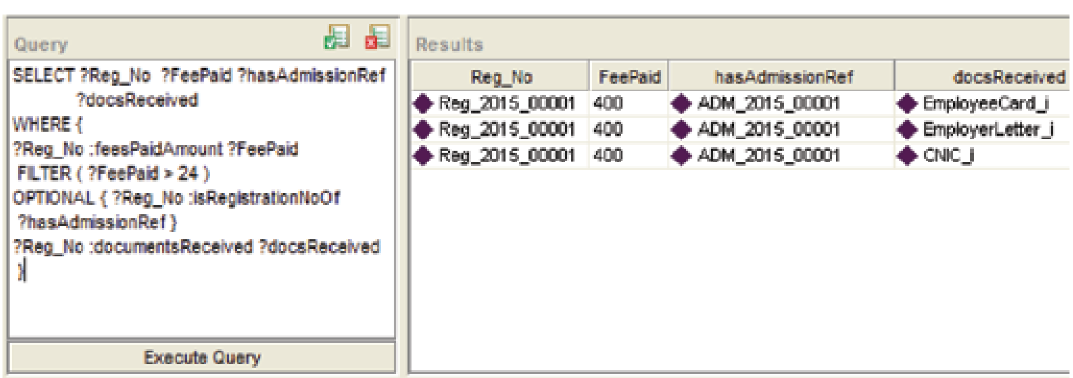


Fig. 8. SPARQL query showing registration.

Triple. RDF works and supports with the data to be distributed or decentralized. RDF graph and models can be combined easily, and RDF serialization can easily be made possible and easily exchanged over the HTTP. Applications can be loosely linked to various RDF enabled data sources over the Web. The SPARQL query matches with the pattern and return the result set in the XML, N3, RDF, OWL format.

A SPARQL query encloses a set of triple patterns, so-called a basic graph pattern. A triple form is similar to an RDF triple (subject, object and predicate), but any component can be a variable. We say that a basic graph pattern matches a sub-graph of the RDF data, when RDF terms of that sub-graph may be substituted for the variables; the result of the matching is an RDF graph equivalent to the sub-graph. The query tries to match the triples of the graph pattern against the RDF data model. Matching means find a set of bindings such that the substitution of variables for values creates a triple that is in the set of triples making up the RDF graph. Each matching binding of the graph pattern variables to the model nodes becomes a query solution, and the values of the variables named in the SELECT clause become part of the query results. Fig. 7 and Fig. 8 shows sample SARQL queries applied on the knowledge-base for the patient and registration process.

4.5 Knowledge Distribution

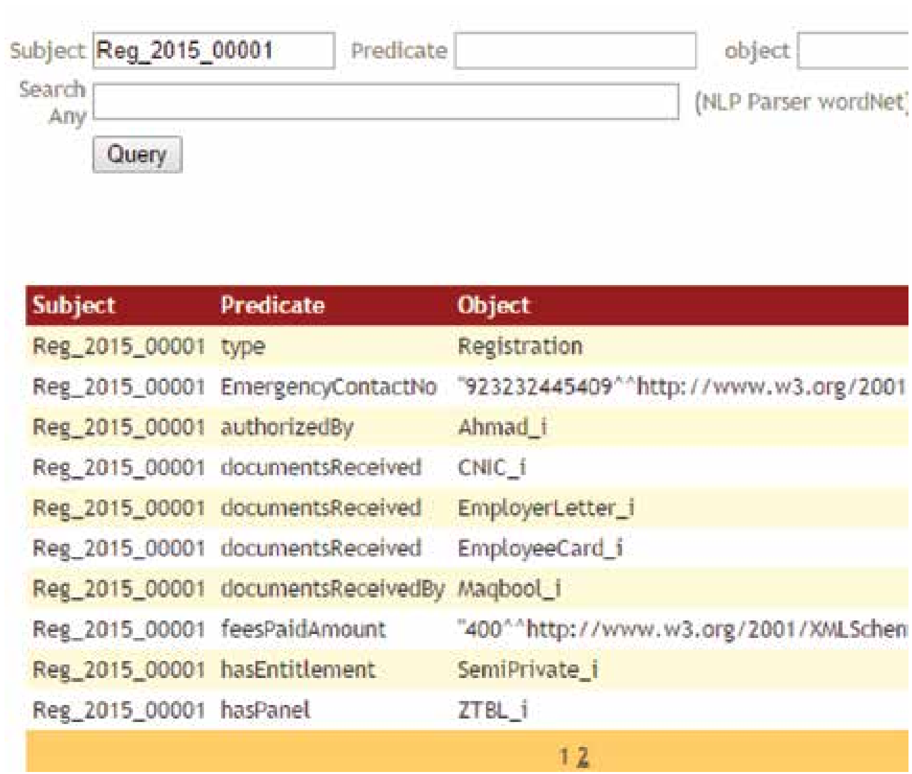
Knowledge distribution is the process of distributing the knowledge to the right person at the right time. Since the knowledge engineer is the person who develops the knowledge base, in order to distribute this knowledge, there is need to develop the interface through which all stakeholders may use the knowledge base easily. Since the developed knowledge base may not only be used by stakeholder for developing shared understanding and refinement in the current business process flow, but after completion of the requirement process this refined knowledge base will be used in the development, verification and validation team to check the software fulfill all the rules and flows defined in the knowledge base by the knowledge engineer with the help of stakeholders. We have develop the interface on the knowledge-base in ASP. NET with dotNetRdf [21] an open source library. The knowledge base initializes by the loading graph Graph graph = new Graph () statement. Fig. 9 represents the graphical user interface of instance level knowledge of registration and its relationship with other instances and literals.

Fig. 9 represents the instance level knowledge of the admission and its relationship with other instances and literals.

We have used an open source java based NLP parser to parse the text and then match with the developed ontology. We have used it for the part of speech tagging. The POS tagger namespace is used to parse a given sentence using NLP techniques and assign the parts of speech to the words and display into the graphical format in a tree shape. When the input string is given to this library it initially tokenize each word in the sentence using "English Maximum Entropy Tokenizer" and return pipe separated token in lower case and then part of speech tagger assign part of speech to each word and return an array of tokens. Fig.11 displays parse context using NLP parser, we can see that it has recognized malaria as NNP (proper noun singular) similarly VBD (verb past tense). Lithium library is used to display in graphical format.

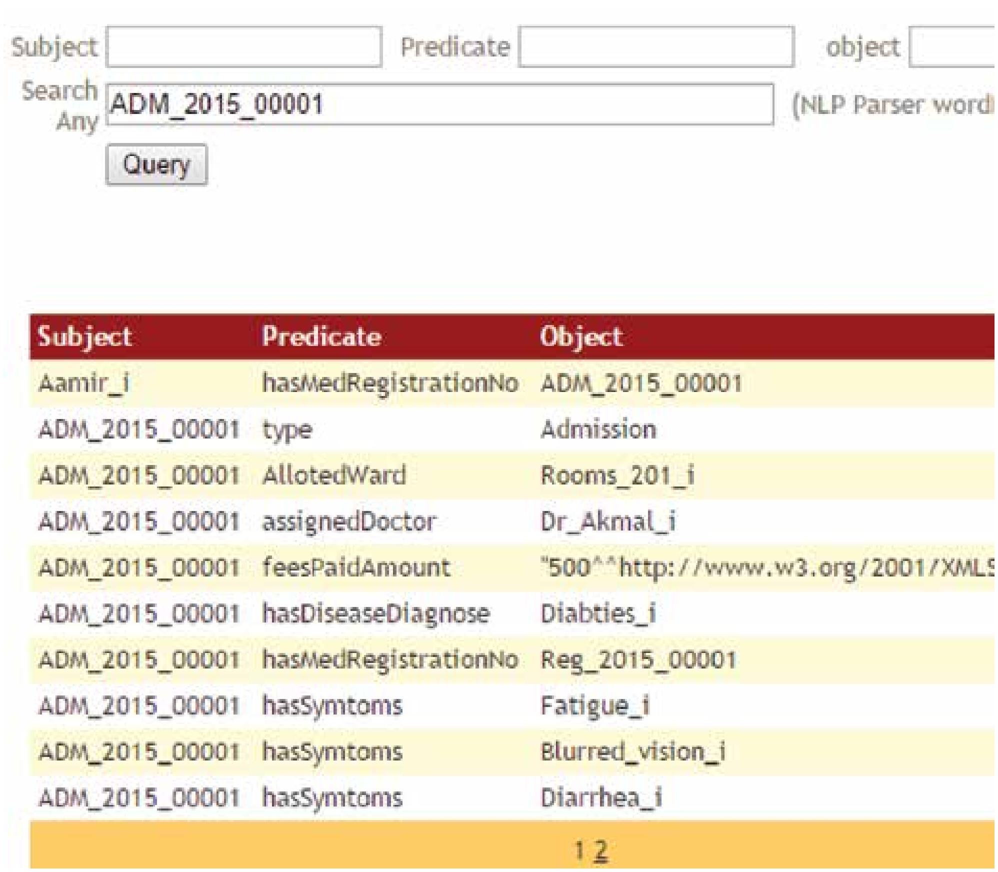
5. EVALUATION & DISCUSSION

We have evaluated our framework by the group of professional working in a software company 'X' provides healthcare solutions located in Islamabad, Pakistan. Their development team consists of five developers two Requirement Engineers one Business Analyst and two Quality Assurance Engineers. We have demonstrated our proposed framework, developed knowledge-base and its user interface which covers the process of registration, admission and billing to their software development team. We have asked them if such knowledge-base on given domain of interest has been developed by using our proposed framework of knowledge management,



Subject	Predicate	Object
Reg_2015_00001	type	Registration
Reg_2015_00001	EmergencyContactNo	"923232445409^^http://www.w3.org/2001
Reg_2015_00001	authorizedBy	Ahmad_i
Reg_2015_00001	documentsReceived	CNIC_i
Reg_2015_00001	documentsReceived	EmployerLetter_i
Reg_2015_00001	documentsReceived	EmployeeCard_i
Reg_2015_00001	documentsReceivedBy	Maqbool_i
Reg_2015_00001	feesPaidAmount	"400^^http://www.w3.org/2001/XMLSchen
Reg_2015_00001	hasEntitlement	SemiPrivate_i
Reg_2015_00001	hasPanel	ZTBL_i
	1 2	

Fig. 9. GUI shows instance level knowledge of regsiterd patient.



Subject	Predicate	Object
Aamir_i	hasMedRegistrationNo	ADM_2015_00001
ADM_2015_00001	type	Admission
ADM_2015_00001	AllotedWard	Rooms_201_i
ADM_2015_00001	assignedDoctor	Dr_Akmal_i
ADM_2015_00001	feesPaidAmount	"500^^http://www.w3.org/2001/XMLS
ADM_2015_00001	hasDiseaseDiagnose	Diabties_i
ADM_2015_00001	hasMedRegistrationNo	Reg_2015_00001
ADM_2015_00001	hasSymtoms	Fatigue_i
ADM_2015_00001	hasSymtoms	Blurred_vision_i
ADM_2015_00001	hasSymtoms	Diarrhea_i
	1 2	

Fig. 10. GUI shows instance level knowledge for admitted patient.

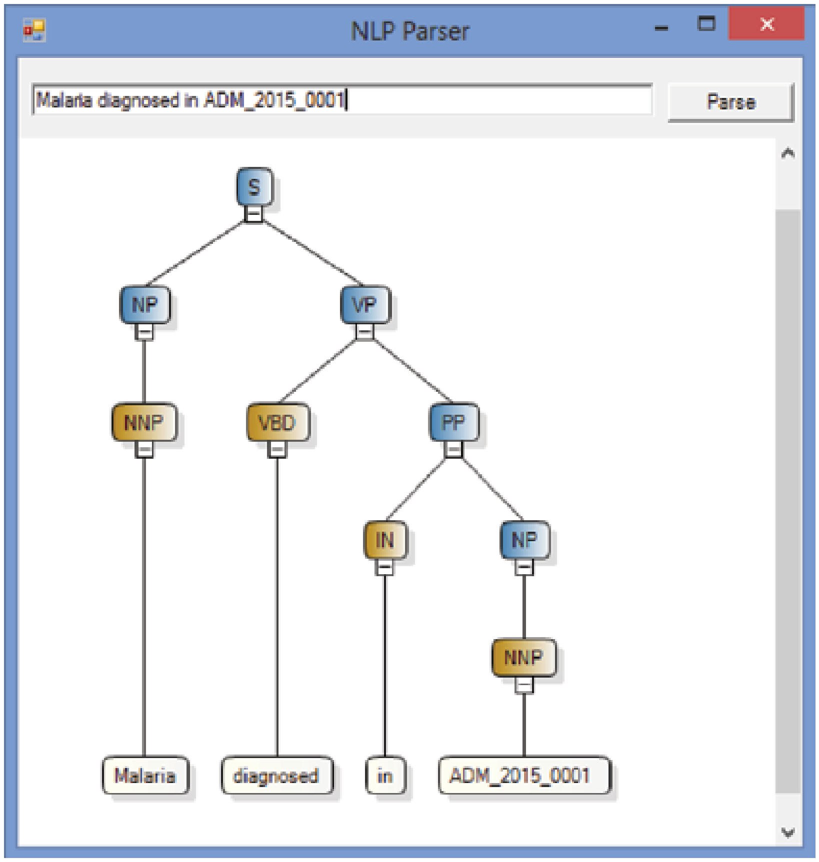


Fig. 11. NLP parser.

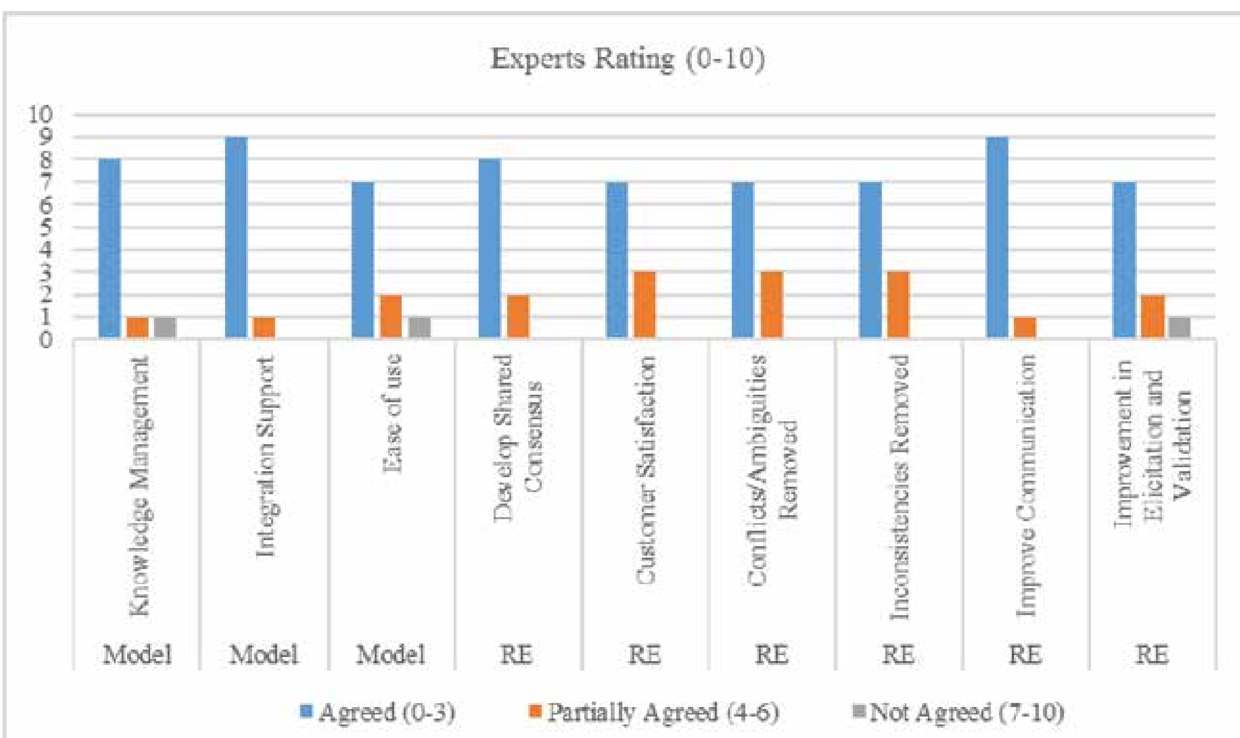


Fig. 12. Graphical representation of experts rating.

would it help in understanding the business domain as well as instance level knowledge along with whether, it would helpful in improving the process of Requirements Engineering by removing the ambiguities and conflicts among the stakeholders, understanding the context and knowledge about the domain. We have proposed the parameters which are based on the knowledge management model as well as parameters impacting on the requirement engineering process improvement. They have applied the rating ranging from 0 to 10. We have tagged rating like 0-3 as Not Agreed, 4-6 as Partially Agreed and 7-10 as Agreed.

We have summarized the result based on agreed, partially agreed and not agreed by applying the mean on all the resultant values of each parameter which is shown in Table 2.

Fig.12. presents the graphical representation of the summarized result of each parameter based on Agreed, Not Agreed and Partially Agreed.

Based on the feedback from the experts, they were mutually agreed that the proposed approach of knowledge management is better to manage the diverse knowledge in a homogenous structure, subject, object and predicate and is capable to manage the knowledge of any domain of interest hence due to homogenous structure it can easily integrate to any other source of knowledge. The developed interface on the knowledge-base is easy to use for searching, however, the initial cost to develop the knowledge model require some technical knowledge and require a knowledge engineer. The developed knowledge base help in shared understanding, removing the conflicts about the context requirement and domain as well as when all customers agreed on the same set of concepts the resultant system will require minimum changes. The improvement in communication has been observed among the stakeholders with different background, geographically distributed as well as among the software agents because it provides semantics and maximum interoperability and conform w3c recommendations. The knowledge based on the domain of interests will not only beneficial for all stakeholders, but the developers and verification

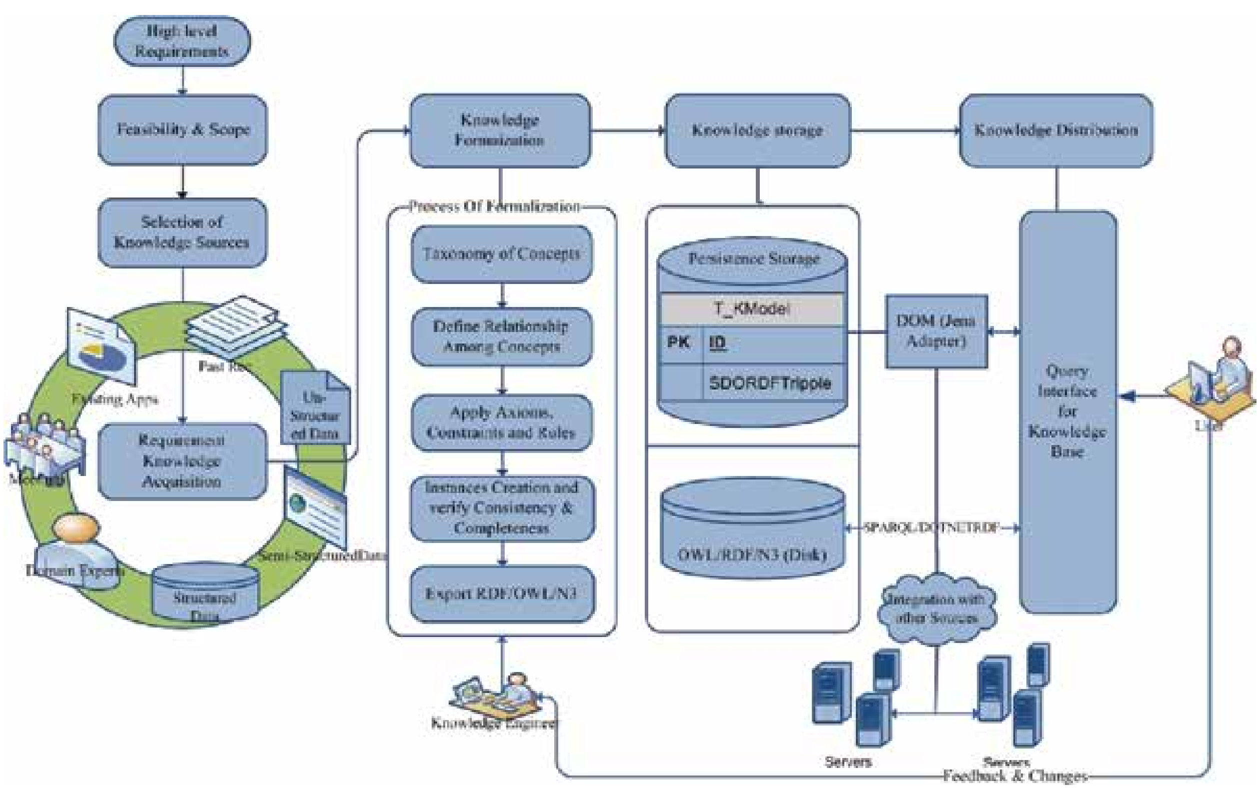


Fig. 13. Requirements knowledge management framework.

team may also use the knowledge base for managing their application level knowledge to check & validate the specification of the software. Hence the whole Requirement engineering life cycle would be improved.

6. CONCLUSIONS

This paper presents the framework of knowledge management. Our proposed framework of knowledge management comprises of five phases: 1) knowledge Acquisition; 2) knowledge Formalization; 3) Persistence Storage; 4) Knowledge Integration; and 5) Knowledge Distribution. The proposed ontology based knowledge management framework demonstration in a Health care domain and its evaluation from the experts shows that it will not only help stakeholders including Analysts, Requirements Engineers to understand the context about the problem domain with instance or behavioral knowledge, along with it will also help to remove the ambiguities and conflicts among business users and technical users by developing a shared understanding of concepts, removing ambiguities and conflicts, improve the communication especially for the stakeholders located geographically. Our next focus is to further evaluate the proposed approach of knowledge management by applying it in the domain of banking industry to see its effectiveness.

7. ACKNOWLEDGEMENTS

We would like to thank Zarai Taraqiati Bank Limited (ZTBL), Islamabad, Pakistan for the help and support regarding this study.

8. REFERENCES

1. Zhang, X. *An Interactive Approach of Ontology-based Requirement Elicitation for Software Customization.* Electronic Theses and Dissertations (University of Windsor): 347, (2011). http://scholar.uwindsor.ca/etd/347.
2. Isserman, S. *Challenges of Requirements Elicitation*. http://www.umsl.edu/~sir3b/Sean_Isserman_Requirements_Elicitation_Home.html.
3. Sethia, N.K. & A.S. Pillai. The effects of requirements elicitation issues on software project performance: An empirical analysis. *Requirements Engineering*: *Foundation for Software Quality, Lecture Notes in Computer Science* 8396: 285-300 (2014).
4. Nahar, N., P.K. Wora, & S. Kumaresh. Managing requirement elicitation issues using step-wise refinement model. *International Journal of Advanced Studies in Computer Science and Engineering* 2: 27-33 (2013).
5. Farfeleder, S., T. Moser, A. Krall, T. Stalhane, I. Omoronyia & H. Zojer. Ontology-driven guidance for requirements elicitation. *Lecture Notes in Computer Science (ESWC)*. 6644, 212–226 (2011).
6. Happel, H., & S. Seedorf. Applications of ontologies in software engineering. *5th International Semantic Web Conference (ISWC 2006)* Cite Seer x, 1–14 (2001). http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.89.5733
7. Castañeda, V., L. Ballejos, M.L. Caliusco & M.R. Galli. The Use of Ontologies in Requirements Engineering. *Global Journal of Researches in Engineering specification*. 10, 6, 2–8 (2010).
8. Jason, G. *An Ontology-based Knowledge Support System for Requirements Analysis,* Massey University, Manawatū, New Zealand (2013). http://hdl.handle.net/10179/4881
9. Humayun, M., C. Gang. An empirical study on improving shared understanding of requirements in GSD. *International Journal of Software Engineering and Its Applications*. 7, 1, 79–92 (2013).
10. Huan, T., B.Q. Vo, M. Lumpe & J. Grundy. KBRE: A framework for knowledge-based requirements engineering. *Software Quality Journal (c). Springer*, March, 35 (2013).
11. Haron, A., S. Sahibuddin. The strength and weakness of requirement engineering (RE) process. *2nd International Conference on Computer Technology and Development. Icctd*, 56–59 (2010).
12. Gasevic, D., N. Kaviani & M. Milanovi. Ontologies and software engineering. p. 1–22 (2008). *DOI: 10.1007/978-3-540-92673-3_27*
13. Reyes-Ortiz, J.A., A., L. Jiménez, J. Cater, C., A. Meléndez, P. Márquez, B. M.& García, Ontology-based knowledge representation for supporting medical decisions. *Research in Computing Science*. 68, 127–136 (2013).
14. Kayed, A., M. Nizar, & M. Alfayoumi. Ontology concepts for requirements engineering process in E-Government applications. 2010 *Fifth International Conference on Internet and Web Applications and Services*. p. 396–400 (2010).
15. Reddy, P.N. & K.P. Kumar. An efficient software engineering ontology tool for knowledge sharing. *IJCSI International Journal of Computer Science Issues.* 7, 4, 19–27 (2010).

16. Ahsan, M., Y.H. Motla & M. Asim. Knowledge modeling for e-agriculture using ontology. *8th IEEE International Conference on Open Source Systems and Technologies 18-20 December (ICOSST-2014),* University of Engineering & Technology (KICS), Lahore. 112-122 IEEE Explore (2014). *doi: 10.1109/ICOSST.2014.7029330*
17. Barforush, A.A. & A.L.I. Rahnama. Ontology learning: revisited. *Journal of Web Engineering*, Vol. 11, No. 4 (2012) 269-289 © Rinton Press. 11, 4, 269–289 (2012).
18. Peffers, K., T. Tuure, M. Rothenberger & S. Chatterjee. A design science research methodology for information systems research. *Journal of Management Information Systems* 24(3): 45–78 (2008).
19. Noy, N. & D. McGuinness. Ontology development 101: A guide to creating your first ontology. Development. 32, 1–25 (2001).
20. Levandoski, J.J., M.F. Mokbel, RDF Data-Centric Storage. 2009 *IEEE International Conference on Web Services*. 1, 911–918 (2009).
21. DotNetRDF Semantic Web, RDF and SPARQL Library for C#/.Net, https://bitbucket.org/dotnetrdf/dotnetrdf/wiki/UserGuide/Querying with SPARQL.
22. Corporation, O. ORACLE DATABASE SEMANTIC, http://docs.oracle.com/cd/E18283_01/appdev.112/e11828/sem_jena.htm, (2013)