

# Requirements Analysis Based on Ontology Techniques

Roshan Bande, Prof. Kapil Hande

**Abstract**— We have designed a method by using which one can analyse the requirement document of the software. It is mapping between software requirement specification and the knowledge base. To represent the knowledge base we have chosen ontology techniques. Our designed system consists of Natural language processing part in which we can separate the parts of speech by tagging them. Natural language generator develops sentences for putting output in the form of natural language. More precisely we developed a system in which input is the SRS document and the output will be a report in which the detailed information will be presented about the SRS document. We have taken care of incompleteness inconsistency, accuracy on these criteria a final verdict is presented whether the SRS document is acceptable or not. Experimental results shows RABOT is indeed a good analysing technique for SRS document analysis.

**Index Terms**— ontology, analysis, knowledge base, artificial intelligence, Expert system, Automation, SRS document.

## 1 INTRODUCTION

One of the goals of requirements analysis is to develop a requirements specification document of high quality.

There are several methods to achieve this goal and their supporting tools are going to be used in practice, e. g., goal oriented requirements analysis methods, scenario analysis, use case modelling techniques and so on. One of the most crucial problems to automate requirements analysis is that requirements documents are usually written in natural language, e. g. English or Japanese. Although techniques for natural language processing (NLP) are being advanced nowadays, it is hard to handle such requirements documents sufficiently by computer. However, semantic processing in requirements is indispensable for producing requirements specifications of high quality. To overcome the problem, there are several approaches, but each of them has its inherent problems. In some studies, a semi-formal notation for representing requirements, e. g. restricted natural languages was introduced, but it was difficult for human engineers to write syntactically and semantically correct requirements sufficiently by using this notation. Rigorous formal notations with axioms and inference system seem to be suitable, but its usage is very limited to practitioners because of their difficulty and complexity in the practitioners' learning and training.

We use an ontology system to develop a software requirements document of high quality. Ontology technologies are frequently applied to many application domains nowadays, because concepts, relationships and their categorizations in a real world can be represented in ontology. Ontology can be

used as resources of domain knowledge, especially in a specific application domain. By using such ontology, several kinds of semantic processing can be achieved in requirements analysis without rigorous NLP techniques.

In this paper, we design a requirements analysis method by using an ontology technique, where we establish a mapping between a requirements specification and ontological elements. This technique allows us to have the possibility of automating semantic analysis with lightweight processing, not heavyweight NLP techniques. By mapping requirements descriptions in a requirements document onto ontological elements, which represents fragments of meaning in a problem domain, each description can be semantically interpreted. By applying inference rules to the ontological elements, we can achieve semantic processing about the requirements document.

## 2 RELATED WORK

"Requirements Analysis and Prototyping using Scenarios and State charts approach" uses precise action semantics, supports changing requirements and enables seamless generation of a fully functional prototype for end user requirements validation. The method is currently being implemented in the STAMP tool (State Modelling and Prototyping).

"Real-time fault diagnosis using knowledge-based expert system" demonstrates that diagnosis methodology is comprised of three steps (Fig. 1) to detect fault. Very first step is acquiring the real-time process information, from critical equipments, such as boilers, compressors, separators or reactors. Temperature, pressure, level, and flow rate are the most important process variables to be monitored and have the capability of representing the state of operation in a variety of equipments. The disorder in these objects can affect the stability and safety of the whole process system. The second step is making inferences (diagnosis) judge on acquired process information. The last step is acting as per inference instruction, such as informing operators, raising alarms, shutting down equipment, activating higher layer protections and trying to

- 
- Roshan Bande is currently pursuing masters degree program in computer science and engineering in RTM Nagpur University, India., E-mail: roshan.bande@rediffmail.com
  - Prof. Kapil Hande, Assistant Professor, Dept. of Computer Science & Engineering E-mail: kapilhande@gmail.com

bring the system back to normal condition.

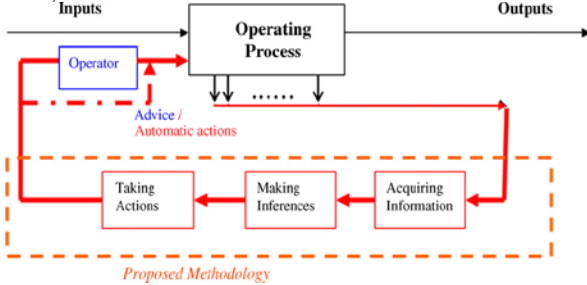


Fig. 1. Three steps of methodology

“ONTOLOGY FOR MOBILE PHONE OPERATING SYSTEMS” is the ongoing study deals with an important part of a line of research that constitutes a challenging burden. It is an initial investigation into the development of a Holistic Framework for Cellular Communication (HFCC). The main purpose is to establish mechanisms by which existing wireless cellular communication components and models can work holistically together. It demonstrates that establishing a mathematical framework that allows existing cellular communication technologies (and tools supporting those technologies) to seamlessly interact is technically feasible. The longer-term future goals are to actually improve the interoperability, the efficiency of mobile communication, calls quality, and reliability by applying the framework to specific development efforts.

"An Automatic Quality Evaluation for Natural Language Requirements" states there is need of software quality analysis. This system uses natural language processing technique to check the construction of sentences and structure of SRS document.

### 3 EXISTING SYSTEM

“Advanced and Innovative Models And Tools for the development of Semantic-based systems for Handling, Acquiring, and Processing knowledge Embedded in multidimensional digital objects” by Information society technology pursued innovations towards digital representations of shapes capable of modelling not only the visual appearance of objects but also their meaning or functionality in a given knowledge domain. In this setting, shape knowledge has been concerned with the geometry (the spatial extent of the object), the structure (object features and part-whole decomposition), attributes (colours, textures), semantics (meaning, purpose), and has had interaction with time (morphing, animation). The harmonization of shape modelling approaches in Computer Graphics and Computer Vision has been pursued via the definition of shared vocabularies and ontologies, not only for the above-mentioned specific domains, but also on a higher level as the basis for the project's eScience platform, the Digital Shape Workbench. As the project's main technological innovation, this workbench served the role of an operational, large-scale, distributed and web-based software system serving as common infrastructure. The scientific innovation sought by this project is focused on modelling the semantics of digital shapes at each stage of their lifecycle.

### 4 METHODOLOGY

The designed system is shown in the flowchart given. In which all the modules are separated on each step. Reading line of SRS document is line by line and each line will be processed to extract keywords. Each keyword will be then fired on ontology the working of it is shown in fig. 3 and fig. 4.

The result generated on query firing will be presented in natural language form using NLG techniques.

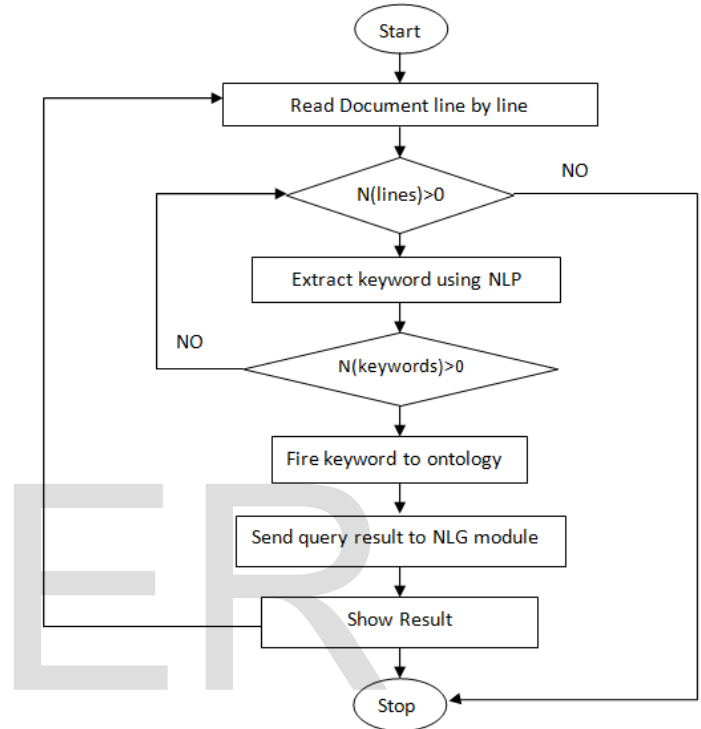


Fig2 Flowchart of the designed system

Designing and development of ontology will be the major task as the ontology will work as the knowledge base in the designed system. The inference engine will be either prolog or any other existing system which will be used as inference engine.

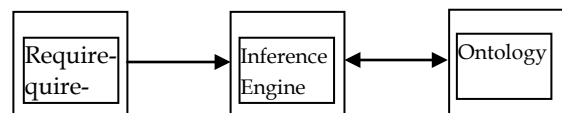


Fig3 . working of designed system

Fig 2 shows the block structure of the designed system . Requirements are input for inference engine inference engine will then perform the guided operation to analyse the requirements in the ontology and show the output as the result of this operation. More detailed working is shown in Fig 3.

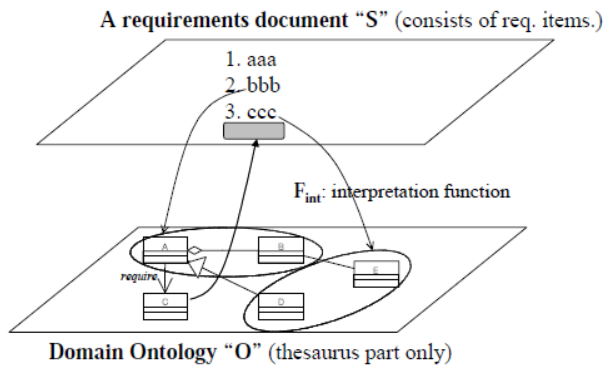


Fig 4. Mapping from requirement to ontology

Fig 3 illustrates mappings from requirements items (statements) in a requirements document to elements in ontology. The requirements document may be described in advance, or it may be described incrementally through the interaction between a requirements analyst and stakeholders. The requirements document is analysed by using this kind of mappings. For example, OBSRA may suspect a requirements document is incomplete when not all elements in an appropriate ontology are related to items in the document. The mapping between the statements and ontology has to be done by using a frame of natural language. OBSRA checks whether a requirements document is consistent or complete by using an ontology system each requirements item (statement) is mapped onto a set of elements (concepts and relationships) in the thesaurus of the ontology system. To detect inconsistency of a requirements document, designed system try to find mutually contradicting elements where requirements items are mapped. For example, designed system decide the document is inconsistent if there is a relationship "contradict" between two concepts where the document is mapped. To detect incompleteness of a requirements document, designed system follow specific relationships from concepts where the document is already mapped. For example, designed system follow "require" relationship and find a concept that does not appear in the current document. Then, designed system add new requirements items (statements) corresponding to the concept.

For the sake of example, we assume a requirement document

1. Home page should have Logo
2. Home page should have name of organization
3. Home page should contain information about organization
4. Homepage should have copyright information
5. Homepage should have Image slider showing the work of organization
6. Home page should have quick form to get the information of user.

Fig 3 shows the input to the system will be requirement documents similar as stated above.

Requirement document is analyzed by our system to find if the keyword is present in the ontology or not if it is present then the linked classes will be put in front of the user in terms

of suggestions so that user can think whether they have to consider for writing the document. These keywords are checked against the ontology of the organization, all the possible linked aspects will be covered which will lead to approximately perfect requirement document.

The expected output of the system is the classes that are related to the keyword present in the requirement document. Such as

1. Name of organization has relation with tagline, phone number, email address.
2. Copyright information has relation with license document.

Ontology development requires thorough study of domain in which it is being developed also it requires keeping account of all relationships. The importance of protégé is demonstrated clearly while developing the ontology. There are several features that distinguish Protégé from other knowledge base editing tools. To the best of our knowledge, no other tool except Protégé has all of the following features: Intuitive and easy-to-use graphical user interface. Scalability: Protégé's database back-end loads frames only on demand and uses caching to free up memory when needed. There is virtually no deterioration in performance as you go from several hundred frames to several thousand frames. Extensible plug-in architecture: We can easily extend Protégé with plug-ins tailored for our domain and task. Some ideas for plug-ins are: Small user-interface components that are particularly well suited to displaying and acquiring values in our domain. Such components could be used on Protégé forms. Custom back-end plug-ins that use our own storage mechanisms. New applications intricately linked with a knowledge base as a Protégé tab.

Prolog rules are used for the knowledge representation, and the Prolog inference engine is used to derive conclusions. Other portions of the system, such as the user interface, must be coded using Prolog as a programming language.

One tentative proposal to achieve the structure in fig. 2 might be that we develop an Ontology using Protege and we load it into Prolog, one of the most obvious consequences of this will be the ontology is accessible by prolog now proper programming will help us to achieve what we have designed.

## 5 EXPERIMENT AND RESULTS

An experiment designed in which ontology named "college.owl" is first build which will work as knowledge base. A SRS document has been developed with a standard template of the SRS document.

SRS document is given as input to the designed system and analysed over the knowledge i.e. ontology. It was expected that the SRS document will be analysed line by line and each line will be process i.e. NLP by which keywords will be extracted and each keyword will be fired as query on ontology the result will then be sent to generate natural language sentences i.e. NLG The result found is as follows:

Entity	RelatedEntities
Library	Library corresponds College, LibraryCard, Validity, ISBN, Book, LibraryStaff, BookStock and Borrower.
Borrower	Borrower exists Library.
Branch	Branch inheres Student.
College	College touches Management, Student, Offices, Staff, Library, Exam, Bus, Departments, Account, Magazine, Events, Celebrity, Infrastructure and Partners.
Account	Account bears upon College, AccountNumber, AccountName, CurrentBalance and Department.

Fig.5 Relation of considered entities with entities not covered.

Entities	Neglected Entities	Total entities	Accuracy	Approval
5	31	36	13%	NO

Fig.6 Number of entities and missing entities its accuracy and approval.

As you can see the first entity in fig 5 is Library which is generated from the statement of the SRS document "The library is the storage of books" in which a keyword library is found and related entities of the Library in the ontology is shown in the Related entities section

## 6 CONCLUSION

In this paper, we design a requirements analysis method by using ontology. Even though the method does not support rigorous natural language processing techniques (NLP), the method enables us to detect incompleteness and inconsistency about a requirements document, to measure the quality of the document, and to predict requirements changes in the future versions of the document. After defining the process to use our ontology approach, we will design and implement its supporting tools. There are many studies using NLP for requirements engineering. For example, inconsistencies in natural language requirements are discovered, conceptual models are semi-automatically generated by linguistic analysis, or formal method and lightweight natural language processing are used together. However, it seems to be unclear how to handle domain knowledge and quality of requirements document itself in such studies. Studies to handle ambiguity in use case descriptions written in natural language exist but they also unclearly handled domain knowledge. How to develop ontology is to be studied. However, most methods for building ontology are ambiguous, thus the quality and efficiency of building ontology depend on the skills of each engineer. Therefore, we have to explore systematic procedure to build ontology. Normally, we focus on the frequency of the occurrences of words or phrases in the documents when we build ontology. In contrast to source codes, there are no unified and formal languages in requirements documents thus it is hard to analyse them in requirements analysis. In our study, ontology plays a role to relate different versions of documents and their change histories with each other, thus we can predict changes in requirements documents. In our study, quality characteristics are also represented as concepts in ontology.

However, such characteristics are represented in a goal model and such goal model and ontology are combined in a study. We also have our own goal oriented requirements model, thus we try to explore the possibility to combine a goal model and ontology. With respect to extending a model for semantic processing, we have to take implementation issues into account. To add knowledge about implementation into ontology, tasks in design and implementation phases could be supported by the ontology.

## REFERENCES

- [1] K. K. Breitman and J. C. S. do Prado Leite. Ontology as a Requirements Engineering Product. In 11th IEEE International Requirements Engineering Conference (RE'03), ages 309-319, Sep. 2003.
- [2] D. Zowghi, V. Gervasi, and A. McRae. Using Default Reasoning to Discover Inconsistencies in Natural Language Requirements. In Eighth Asia-Pacific Software Engineering Conference (APSEC'01), pages 113-120, Dec. 2001.
- [3] N. Guarino and C. Welty. Evaluating Ontological Decisions with Ontoclean. Communications of the ACM, 45(2):61-65, Feb 2002.
- [4] M. Khedr and A. Karmouch. Negotiation Context Information in Context-Aware Systems. IEEE Intelligent Systems, 19(6):21-29, Nov/Dez 2004.
- [5] O. Khriyenko and V. Terziyan. Context Description Framework for the Semantic Web. In Context Representation and Reasoning Workshop, Jul 2005.
- [6] Jason I. Hong and James A. Landay, "An Infrastructure Approach to Context-Aware Computing", Human-Computer Interaction, Vol. 16, 2001.
- [7] Harry Chen and Tim Finin, "An Ontology for a Context Aware Pervasive Computing Environment", IJCAI workshop on ontologies and distributed systems, Acapulco MX, August 2003.
- [8] Anand Ranganathan and Roy H. Campbell, "A Middleware for Context-Aware Agents in Ubiquitous Computing Environments", In Proceedings of ACM/IFIP/USENIX International Middleware Conference, Rio de Janeiro, Brazil, June 2003.
- [9] M. Smith, C. Welty, and D. McGuinness, Web Ontology Language (OWL) Guide, August 2003.
- [10] Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, Paul Webster, "The Anatomy of a Context-Aware Application", Wireless Networks 8(2-3): 187-197(2002).
- [11] H. Wu, M. Siegel, and S. Ablay, "Sensor Fusion for Context Understanding", Proceedings of IEEE Instrumentation and Measurement Technology Conference, Anchorage, USA, May 2002.
- [12] Henrikksen K, Indulska J, Rakotonirainy A., "Modeling Context Information in Pervasive Computing Systems", In Proceedings Pervasive Computing, Zurich, August 2002.
- [13] Karen Henrikksen, Jadwiga Indulska, and Andry Rakotonirainy, "Generating Context Management Infrastructure from High-level Context Models", In Proceedings of the 4th International Conference on Mobile Data Management, Melbourne, January 2003.
- [14] Held, A., Buchholz, S., Schill, A., "Modeling of Context Information for Pervasive Computing Applications", In Proceedings of the 6th World Multiconference on Systemics, Cybernetics and

- Informatics (SCI), Orlando, FL, July 2002.
- [15] Ian Horrocks, "DAML+OIL: a Reasonable Web Ontology Language", In Proceedings of the 8th International Conference on Extending Database Technology (EDBT), Prague, March 2002.
- [16] Dan Brickley, R. V. Guha, RDF Vocabulary Description Language 1.0: RDF Schema, World Wide Web Consortium, January 2003.
- [17] Jena 2 - A Semantic Web Framework, <http://www.hpl.hp.com/semweb/jena2.htm>
- [18] Tao Gu, H. C. Qian, J. K. Yao, H. K. Pung, "An Architecture for Flexible Service Discovery in OCTOPUS", In Proceedings of the 12th International Conference on Computer Communications and Networks (ICCCN), Dallas, Texas, October 2003.
- [19] Henricksen K, Indulska J, Rakotonirainy, "Infrastructure for Pervasive computing: Challenges", Workshop on Pervasive Computing INFORMATIK 01, Viena, September 2001.
- [20] F. Fabbri, M. Fusani, S. Gnesi, G. Lami, "An Automatic Quality Evaluation for Natural Language Requirements", Istituto di Elaborazione dell'Informazione del C.N.R. - Pisa, Italy
- [21] Dey, A. and Abowd, G., "Towards a Better Understanding of Context and Context-Awareness", Workshop on the what, who, where, when and how of context-awareness at CHI 2000, April 2000.
- [22] Dey, A. K., Salber, D. Abowd, G. D., "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications", Human-Computer Interaction (HCI) Journal, Vol. 16(2-4), pp. 97-166, 2001.
- [23] T. Kindberg and J. Barton, "A Web-based Nomadic Computing System", Computer Networks, 35(4):443-456, 2001.
- [24] Information Society Technology - "Advanced and Innovative Models And Tools for the development of Semantic-based systems for Handling, Acquiring, and Processing knowledge Embedded in multidimensional digital objects" [http://cordis.europa.eu/ist/kct/aimatshape\\_synopsis.htm](http://cordis.europa.eu/ist/kct/aimatshape_synopsis.htm).
- [25] C Nan, F Khan, MT Iqbal "Real-time fault diagnosis using knowledge-based expert system" - process safety and environmental protection, 2008 - Elsevier
- [26] E. Kamsties, B. Peach Taming Ambiguity in Natural Language Requirements. ICSSEA 2000-5.
- [27] Fantechi, M. Fusani, S. Gnesi, G. Ristori. Expressing properties of software requirements through syntactical rules. Technical Report. IEI-CNR, 1997.
- [28] Fantechi, S. Gnesi, G. Ristori, M. Carenini, M. Vanocchi, P. Moreschini, "Assisting requirement formalization by means of natural language translation", Formal Methods in System Design, vol 4, n. 3, pp. 243-263, Kluwer Academic Publishers, 1994
- [29] J. Krogstie, O. I. Lindland, G. Sindre. Towards a deeper understanding of quality in requirements engineering. In 7th International CAISE Conference, vol. 932 of Lecture Notes in Computer Science, pages 82-95, 1995.
- [30] G. Lami. Towards an Automatic Quality Evaluation of Natural Language Software Specifications. Technical Report. B4-25-11-99. IEI-CNR, 1999.
- [31] Ben Morris, the Symbian OS Architecture Source book Design and Evolution of a Mobile Phone OS, John Wiley & Sons, Ltd 2007.
- [32] Stefan Brahler, "Analysis of the Android Architecture", Karlsruhe institute for technology, October 2010.
- [33] Niles, I & Pease A. (2001) "Towards A Standard Upper Ontology." In Proceedings of FOIS 2001, October 17-19, Ogunquit, Maine, USA.
- [34] Wang, P., Xu, B. LILY: "The Results for the Ontology Alignment Contest" OAEI 2007. In Proceedings of ISWC 2007 Ontology Matching Workshop. Busan, Korea.
- [35] O. Khriyenko and V. Terziyan. Context Description Framework for the Semantic Web. In Context Representation and Reasoning Workshop, Jul 2005.
- [36] O. Lassila and D. Khushraj. Contextualizing Applications via Semantic Middleware. In 2nd Ann. Int'l Conf. on Mobile and Ubiquitous Systems: Networking and Services, Jul 2005.
- [37] D. L. McGuinness and F. van Harmelen. OWL Web Ontology Language Overview. W3C Recommendation, Feb 2004.
- [38] Jawad Makki, Anne-Marie Alquier, Violaine Prince, 'An NLP-based ontology population for a risk management generic structure', 5th International Conference on Soft Computing as transdisciplinary Science and Technology 2008, pages 250-355.
- [39] Christian Cuske, Tilo Dickopp, Stefan Seedorf 'An ontology based platform for knowledge based simulation modeling in financial risk management', European simulation and Modeling Conference 2005
- [40] Dillon, T. S., Chang, E., Wongthongtham, P, 'Ontology-Based Software Engineering - Software Engineering 2.0', 19th Australian Conference on Software Engineering, 2008, 26-28 March 2008, Page(s):13-23