Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 79 (2016) 852 – 860

Procedia Computer Science

7th International Conference on Communication, Computing and Virtualization 2016

A Novel Approach for Specifying Functional and Non-Functional Requirements using RDS (Requirement Description Schema)

Tejas Shah[a] *, S V Patel[b]

[a]*MSc (IT) Programme, Veer Narmad South Gujarat University, Surat, 395006, India*
[b]*Veer Narmad South Gujarat University, Surat, 395006, India*

Abstract

Requirement Engineering demands a granular level of requirement specifications with key objectives, design constraints and relevant artefacts. There exist some structured approaches, but still these are not complete and do not have open formats that describe requirements of a system/project with its artefacts. This paper introduces RDS (Requirement Description Schema), an XML-based versatile specification approach for the structural representation of functional and non-functional requirements (NFR). The approach is an efficient way of managing requirement metadata and comprehensive artefacts of requirements like status, priority, version, stability, elicitation source etc. The paper comprises a case study of online examination system for validating the instances with RDS.

Peer-review under responsibility of the Organizing Committee of ICCCV 2016

Keywords: Requirement Engineering; RDS; Requirement Description Schema; Requirement Artefacts; Non-Functional Requirement

1. Introduction

Requirement Engineering (RE) means activities involved in discovering, analysing, verifying, documenting and preserving a set of requirements for a system[1]. RE is considered to be critical and complex process within the software intensive systems development[2,3,4]. It is critical because the quality of the systems is strappingly affected by the quality of the requirements. It is complex as the diverse set of product demands from the diverse set of stakeholders has to be considered. Many errors can originate/propagate from requirements phase, caused by poorly written, ambiguous, unclear or missed requirements. Failure to specify the requirements correctly can lead to major delays, cost overruns.

* Tejas Shah. Tel.: + 91-987-911-0707;
*E-mail address:*proftejas@gmail.com

Peer-review under responsibility of the Organizing Committee of ICCCV 2016
doi:10.1016/j.procs.2016.03.083

Good efforts have been made for investigation of alternative elicitation paradigms beyond a pure automation approach as well as semi-automated requirement elicitation[5]. Furthermore, as the IoT (Internet of Things) applications are vulnerable to security and privacy attacks, web applications and sensor networks require special attention to NFR[6,7].

Requirements Specification is one of the utmost important RE tasks during which elicited and analyzed requirements are properly documented for use by their envisioned stakeholders. The traditional approach of RE may produce the document (like a word document) during the initial requirement phase of a project. The manual requirement specification approach is time-consuming, expensive and difficult, and resultant specification typically becomes inconsistent, incomplete, ambiguous and hard to trace[8]. The traditional approach does not consider the metadata of the requirements. The requirements in textual form are difficult to process or it requires the NLP support. Therefore, it is wise to have a new approach with structured requirement specification format. An additional benefit is such format can be easily transformed into requirement repositories for requirement management and requirement traceability.

The paper presents a novel approach Requirement Description Schema (RDS) for structured requirements specifications. It makes extensive use of standardized XML Schema Definition Language[9] as XML is most appropriate to define unique vocabularies tendered to suit the various domains. In addition to incorporating conventional functional requirements, the RDS also incorporates a variety of non-functional requirements properties with attributes of security and privacy.

The RDS aims to offer benefits such as (1) Representing complex and unstructured requirement components in the electronic and interoperable form. (2) Exchanging requirements amongst stakeholders, business analysts and developers in internal as well as an external environment in a uniform format (3). The form based requirement elicitation in the form of XML can be easily validated against various RDS schema. (4) Reduce ambiguity (5) Requirement management and traceability is feasible with the help of appropriate XML elements.

This paper is organized as follows: Section II entails relevant research in the field of requirement markup languages and specifications. Section III includes potential areas of RDS. Section IV includes the design structure of RDS specification with the case study of the online examination system. Section V compares the RDS specification with other markup languages. The last section comprehends the concluding remarks and future work aiming at the extension of RDS.

2. Related Work

There are only a few markup languages and methodologies available in the literature covering functional and non-functional requirement description along with requirement artefacts. Most of the methodologies proposed for the specification of requirements pay less attention to requirement artefacts and NFR coverage. During the past few years, several research efforts have focused on specifying key functional requirements only.

The RGML (Requirement Generation Markup Language) has created the formal specification mechanism for characterizing the structure, process flow for the process of requirements generation. The work focuses on characterization of application instantiation, the use of templates and the productions of artefact to assist the requirement engineer[10], however, it does not include specification and activities of NFR.

The SRS template is represented in XML with the consideration of the object oriented environment[11]. The template contributed to the simplification and standardization of the procedure for writing requirements and the validation of the domain against use case models. It only focuses section wise SRS representation. The semantic part of use case descriptions are represented in XML. As modelling requirements with use cases is proven useful, the authors Dimitris et al presented the structure of use cases with appropriate tags[12]. Michel dos Santos Soares and Jos Vrancken represents the requirements in the graphical and tabular way to fill the gap between requirement documents and use case through

SysML[13]. The work mentioned in[14] focuses on the formal and informal classification of requirement and specifying those requirements with the XML Schema. However, it lacks coverage on requirements metadata.

Requirements Markup Language (RQML) is an XML dialect for specifying software requirements which overcome the drawback of natural language requirement specification. RQML is implemented as the representation of requirements document with rich element types, but the structure representation is in DTD format[15]. The paper based on RQML addresses the problem and solution of collaboration on managing and documenting the software requirement elicitation focusing on creation of elicitation assistant.[16]

The work mentioned in RQML, RGML do not cover all the metadata of the requirements. (For e.g. Requirement status, priority, complexity, and version). The RGML approach covers the process description language also but not covering the NFR properties of the requirements. The RGML is using requirement generation, describing the process structure, flow of control. However, the requirements artefacts for priority, version management is not included. The RQML is implanted to run on Palm OS with requirement representation class in DTD form.

It may be noted that none of the above approaches totally cover all aspects of requirement artefacts, functional and non-functional requirements. Therefore, instead of preaching the use of one markup language while neglecting potential benefits of the other, we present an integrated notion of requirement artefact representation including functional and non-functional requirements to map the business processes into effective requirement interchangeable elements.

3. Potential Areas of RDS

The Requirements can be simple or complex in the sense that they represent the functional and non-functional behaviour of a system often requiring an interaction, dependency and priority selection. RDS is meant to support both categories of requirements, but complex requirements have provided the primary motivations for the design of this schema. The following key points give concrete objectives and potential area of RDS schema.

3.1. Requirement Specification with Interchangeable Format

The requirement engineering tools manage the enormous repositories for the requirements. Most of the companies are storing their requirement in office document which creates traceability and granularity problems. The RDS specification format can bridge the gap for migrating and transforming the functional and non-functional requirement for different projects. Consequently, there will be fewer problems with the domain as the data format is the XML file.

3.2. Mapping RE Phase to Design Phase

The RDS data in the form of XML can be easily transferred to the designer, programmer and service developer for progression towards design phase. The design and composition process will be efficient as the requirement and non-functional requirement data are presented in XML format only. This makes it possible to trace the requirements from customer level statements to final module level element.

3.3. Requirement Artefacts Selection

Requirement artefacts like the priority, status, rationale, validity, version tracing can be proactively found by firing a query to the requirement repository. The RDS will use the declarative API for executing the Requirement selection task based on their priority, version and elicitation source.

3.4. *Transformation of Requirements to RDS Format with GUI*

The majority of available RE tools processes the large text of requirements and the process of requirement collection and specification are quite complex and based on NLP (Natural Language Processing). Through RDS, it is very convenient and simple to transform field wise requirements along with artefacts to XML schema instance using requirement elicitation form. The specified requirements should conform to the RDS schema structure. This semi-automatic approach reduces the efforts, development time and assist business analyst and developer to process the validated XML schema file. With the help of XML parser, the minute requirement is even transferred in interchangeable format.

3.5. Linkage with XML-based Security Standards

The NFR element tag includes different security and privacy related elements. There are plenty of security standards available to secure the transactions. The linkage between NFR element tags and XML-based security standards ease the security implementation.

4. RDS Design Structure

Each section includes sample element structure for the various RDS schema. The case study of online examination system is validated to different RDS schema with few key elements and relevant elements are described in this section.

4.1. Main RDS Schema

RDS provides a broad set of XML elements that define functional and non-functional requirements of a system, requirement artefacts. The RDS schema is an integration of 3 different schemas depicted in Fig. 1.

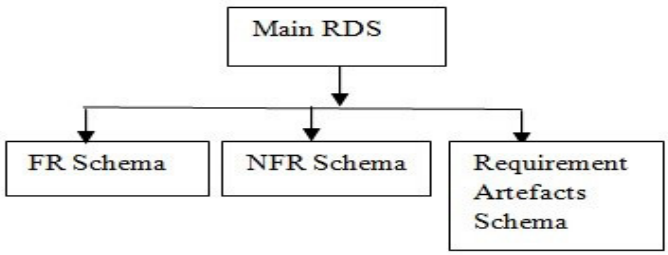

Fig. 1 RDS Schema

4.2. Functional Requirement Schema

The functional requirement describes the desired key features of a system. As shown in Fig. 2, this schema represents the sub-requirement of the system in modular form. The module input and output are having the same schematic description with parameters, data type and which actor has performed that operation. Some of the non-functional behavior should be maintained with the module also. The requirement engineer can assign the authorization right to the particular or set of actors ensuring the confidentiality. The process which will process the input and generating output with dependency like data flow diagram is represented. The mapping of module data to web service and identification of web service is accessed from this element. Furthermore, the data flow diagram design can be linked with the sub requirement module where processes are described implicitly.

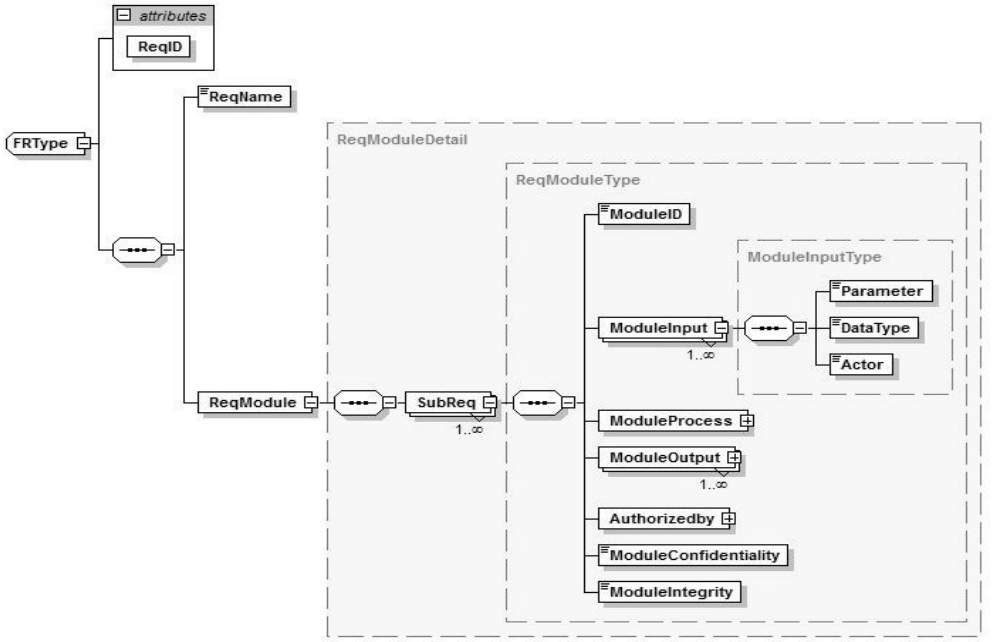

Fig. 2 FR Schema

Sample Element tags for FR Schema

```
<xs:complexType name="FRType">          <xs:sequence>
                    <xs:element name="ReqName" type="xs:string"/>
                    <xs:element name="ReqModule" type="ReqModuleDetail"/>
                                        </xs:sequence>
        <xs:attribute name="ReqID" type="xs:integer" use="required"/>
</xs:complexType>
<xs:complexType name="ReqModuleType">          <xs:sequence>
        <xs:element name="ModuleID" type="xs:integer"/>
        <xs:element name="ModuleInput" type="ModuleIOType" maxOccurs="unbounded"/>
        <xs:element name="ModuleProcess" type="ModuleProcessType"/>
        <xs:element name="ModuleOutput" type="ModuleIOType" maxOccurs="unbounded"/>
        <xs:element name="ModuleConfidentiality" type="Conf_Inte_Required"/>
                        </xs:sequence>  </xs:complexType>
<xs:complexType name="ModuleIOType"> <xs:sequence>
                    <xs:element name="Parameter" type="xs:string"/>
                    <xs:element name="DataType" type="DataType"/>
                    </xs:sequence>  </xs:complexType>
```

4.3. NFR Schema

When it comes to defining non-functional requirements, the business users are less aware and do not recognize the importance of NFR. The non-functional requirements like security, privacy, reliability, performance are inherent and more important in the applications running on the internet. Each element is having its impact in all spheres of the requirement specification. Keeping in line with the context of security and privacy in software development, this schema focuses on different components of non-functional requirements shown in Fig. 3.

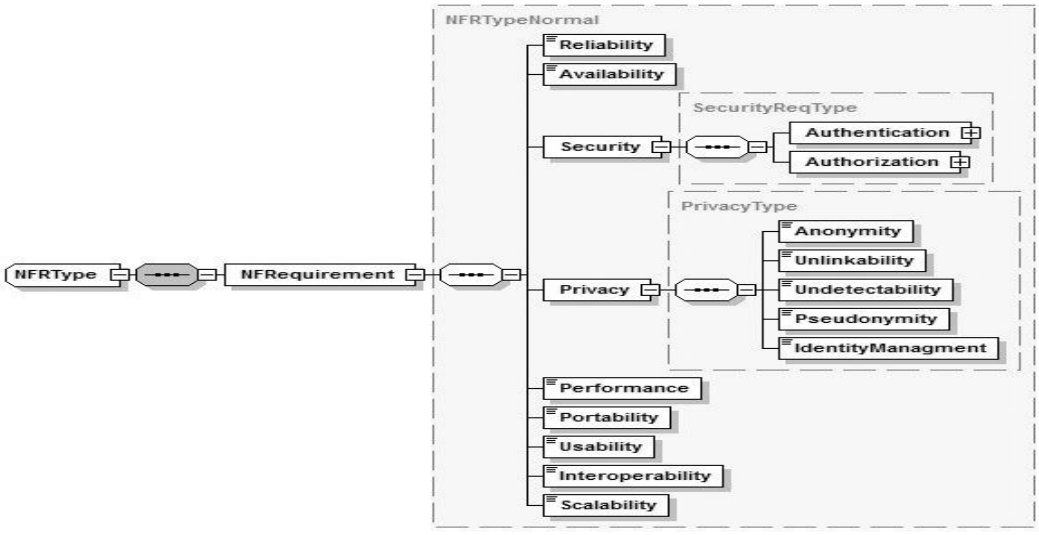

Fig. 3 NFR Schema

Sample Element tags for NFR Schema

```
<xs:complexType name="NFRTypeNormal"> <xs:sequence>
        <xs:element name="Reliability" type="xs:string"/>
        <xs:element name="Availability" type="xs:string"/>
        <xs:element name="Security" type="SecurityReqType"/>
        <xs:element name="Privacy" type="PrivacyType"/>
        </xs:sequence> </xs:complexType>
<xs:complexType name="SecurityReqType"> <xs:sequence>
        <xs:element name="Authentication" type="Authentication_Type"/>
        <xs:element name="Authorization" type="Authorization_Type"/>
                </xs:sequence> </xs:complexType>
<xs:complexType name="PrivacyType"> <xs:sequence>
        <xs:element name="Anonymity" type="xs:string"/>
        <xs:element name="Unlinkability" type="xs:string"/>
</xs:sequence> </xs:complexType>
```

4.4. Requirement Artefacts Schema

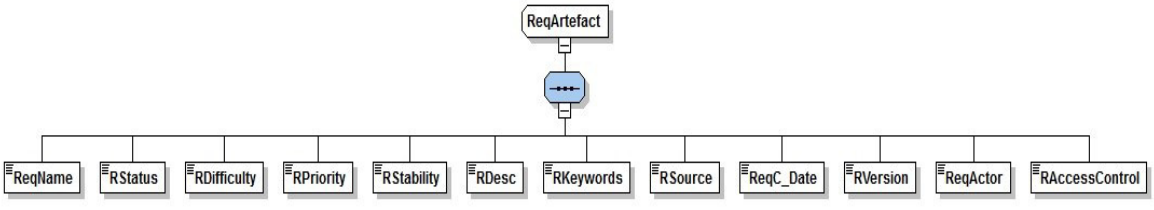

Fig. 4 Requirement Artefacts Schema

The reuse of existing requirement artefacts makes the RE task more prescriptive and systematic as described in[17], hence the requirement artefacts inclusion is relevant over here. A few artefacts that help in determining requirement

metadata are requirement priority, status, source, version, and stakeholder. As depicted in Fig. 4, the different elements cover almost all requirement artefacts pertaining to a specific functional requirement. This schema attempts to cover properties of a functional requirement to efficiently manage traceability and management phase of RE. The requirement artefact tag consists of elements like priority, type, version, description of the requirement. The use of XQuery gives the listing of the requirement by providing values in the parameters. The customer can store the keywords of the requirement and rationale also. The requirement source maintains the list of elicitation sources.

Sample element tags for Artefacts Schema

```
<xs:complexType name="ReqArtefact">
    <xs:sequence>
        <xs:element name="RStatus" type="ReqStatus"/>
        <xs:element name="RPriority" type="ReqPriority"/>
        <xs:element name="RDesc" type="xs:string"/>
        <xs:element name="RSource" type="ReqElictSource"/>
        <xs:element name="RVersion" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="ReqPriority">
    <xs:restriction base="xs:string">
        <xs:enumeration value="MustHave"/>
        <xs:enumeration value="IMP"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ReqElictSource">   <xs:restriction base="xs:string">
        <xs:enumeration value="Interviewsummary"/>
        <xs:enumeration value="QuestionnarieFile"/>
        <xs:enumeration value="Emailcontent"/>
        <xs:enumeration value="UserStories"/>
        <xs:enumeration value="BrainstormingSession"/>
    </xs:restriction>  </xs:simpleType>
```

4.5. Case Study and Result

Table 1 Key Requirement Elements for Online Examination System

No.	Module Name(Requirement)	Type: FR/NFR	RDS Schema	Element
1	Submission of Paper	Both	FR Schema NFR Schema	ReqModule: Input Parameter: Faculty, Subject Output Parameter: Status, Security: Authentication, Authorizatioin
2	Marks Verification	Both	FR Schema NFR Schema	ReqModule: InputParameter: Student,Subject Output Parameter Security: Confidentiality: Required Privacy: Identity of Faculty
3	Course, Subject Selection	FR	FR Schema	ReqModule: Input Parameter: Course, Student
4	Appear in Examination	NFR	NFR Schema	Privacy: Anonymity [Student]
5	Student Registration	NFR	FR Schema	ReqModule: Input Parameter: Course,Student Output Parameter: Registration ID
6	Examiner Detail	FR	FR Schema	ReqModule: Input Parameter: Examiner, Course
7	View Result	FR	FR Schema	ReqModule: Input Parameter: Course, Subject
8	Grade Calculation	FR	FR Schema	ReqModule: Output Parameter: Student, Grade

To specify the requirements with RDS, the online examination system case study is considered. The scope of specifying detailed requirements is limited; hence some of the elements are mentioned in table 1 with respect to customer requirements. The table includes the name of requirement, its type, which schema is used and few elements of that schema. The suitability and effectiveness of specifying functional, non-functional requirement including artefacts of online examination system are justified. The results have shown the online examination XML file validation against RDS schema which is valid and well formed.

5. Comparison of RDS with other Markup Languages and Specification

The following table 2 summarizes the comparison of RDS with other specifications and markup languages. The criteria are selected from the survey done on XML-based policy languages[18]. There are 2 aspects for comparison.

- Language Aspect
- Requirement Engineering Aspect

Please note that each markup language mentioned here is in a different state of development. The RE aspect includes functional and non-functional requirement, use case and interface support in different specifications.

Table 2 Comparison of RDS with other markup languages and specifications

Language Aspect	Specification with XML [11]	RGML	RQML	RDS
Syntax	XML Schema	XML Schema	XML DTD	XML Schema
Complexity Level	Medium	High	High	Low
Expressiveness	High	High	Medium	High
Ambiguity	Relatively Low [Represents the SRS in sections]	Relatively Low [Represents activities of Requirements	Relatively Medium [Presented in terms of code for palm top]	Relatively Low [Represents requirements at module level with artefacts]
Content Access	YES	YES	Not Relevant	YES
Interoperability Level	High	High	Not Relevant	High

RE Aspect	Specification with XML [11]	RGML	RQML	RDS
Requirement Artefacts and Attributes like priority, version	≈ Attributes are not specified in the SRS template	≈	√ [Included as a first class elements]	√ [14 different requirement artefact elements with enumerated values]
NFR Inclusion[Security, privacy]	≈ [As a part of only SRS Section in text form]	≈ [Indirectly with use case representation]	≠	√ [At module level and separate also for NFR]
NFR [Performance, Scalability and other attributes]	≠	≈ [Indirectly with use case representation]	≠	√ [As NFR element tag]
Module wise input output	≠	≈	≈ [Included as a part of use case interface code]	√ [Separate element tag for module of Sub requirements]
Use case Attribute	√ Use cases are included at specification level	√	√ [Use case as first class RQML document]	≈ [Role of Actor for accessing module is described]
Interface Interaction	≠ [SRS in XML format only]	√	√ [Handheld and desktop based interface]	≈ [Data collection in XML format with form based interface]

√ Included with strong support ≈ Some support of the feature ≠ No support in the specification

The idea of separation of concerns for the functional and non-functional requirement is the key characteristic of almost all RE methodologies. However, this separation of concerns was not taken care of in RGML, RQML languages.

Some of the markup languages compared does not include the sub requirement description phase in their RE element tags. The comparison reveals that the RDS addresses both functional and non-functional requirements. Further, it also emphasizes on metadata of the requirements.

6. Conclusion

This paper reveals the initial design of the RDS (Requirement Description Schema) specification suitable for the use by requirement engineer, business analyst, service consumers and naive customers. With some RDS conventions that facilitate translation and mapping to requirement representation systems, the semi-automatic process of eliciting the requirements can be developed. The advanced architecture with RDS can extend the application of RE and Service Oriented Requirement Engineering to automated requirement elicitation form processing, preventive NFR processing, requirement data exchange and integration, and web service mapping with the processing of module element tag. Matured RDS applications may change the way of designing RE system in the near future. Further validation of the language by the community and by using a wide range of requirement tools is in progress.

References

1. Sommerville, I. & Sawyer, P. 1997. Requirements engineering: A good practice guide. *John Wiley & Sons.* ISBN-13: 978-0471974444
2. Firesmith, D. G. 2005. Quality requirements checklist, *Journal of Object Technology*, Vol. 4, No. 9, November–December 2005, pp. 31–38
3. Damian, D. 2002. The study of requirements engineering in global software development: as challenging as important. *In: Proceedings of Global Software Development, Workshop #9.* Organized in the International Conference on Software Engineering (ICSE) 2002, Orlando, FL, ISBN 1-86365-699-5
4. Standish Group 2006. The CHAOS Report published on http://www.standishgroup.com 1996, 1998, 2000, 2002, 2004 and 2006. The Standish Group International, Inc
5. Meth, H., Brhel, M., Maedche, A., 2013. "The state of the art in automated requirements elicitation", *Information and Software Technology*. 55 (10), 1695–1709. March 2013
6. L. Atzori, A. Iera, G. Morabito, *The Internet of Things: a survey*, *Computer Networks* 54 (2010) 2787–2805
7. J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami. Internet of things (IoT): A vision, architectural elements, and future directions, *Future Gen. Comput. Syst.*, vol. 29, no. 7, 2013, pp.1645 -1660
8. Donald Firesmith, Modern Requirements Specification, *Journal of Object Technology*, Vol. 2, No. 1, March-April 2003
9. W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures, W3C Recommendation 5 April 2012, URL: http://www.w3.org/TR/xmlschema11-1, Accessed on 17 December 2015
10. Ahmed Sidky, RGML: A Specification Language that Supports the Characterization of Requirements Generation Processes, *Master thesis, Virginia Tech*, 2003
11. Kenza Meridji, Documentations and validation of requirements specifications – An XML approach, *Other Thesis, Concordia University*, 2003,
12. D.Dranidis, K. Tigka, Writing Use Cases in XML,*9th Panhellenic Conference in Informatics*, 2003.
13. Michel dos Santos Soares, Jos Vrancken. Requirements Specification and Modeling through SysML, *IEEE International Conference on Systems, Man, and Cybernetics - SMC 2007*, ISBN 1-4244-0991-8, IEEE, 2007, pp. 1735-1740
14. Shreyas chavda, Dr. Shantharam Nayak, Modern Technique To Build Software Requirements Specification, *IJSRD - International Journal for Scientific Research & Development*, Vol. 2, Issue 03, 2014 | ISSN (online): 2321-0613
15. Sasmito Adibowo, Rambutan, Requirements Management Tool for Busy System Analysts, *Technical Report*, July 2003
16. Eko K. Budiardjo, Sasmito Adibowo, RQML Based Mobile Requirement Elicitor Assistant, *SITIA (Seminar on Intelligent Technology and Its Applications)*, Surabaya, May 9th – 10th, 2007
17. Betty H.C. Cheng, Joanne M. Atlee, Research Directions in Requirement Engineering, *Future of Software Engineerig (FOSE'07) IEEE*, 0-7695-2829-5/07, 2007
18. Mariemma I. Yague, Survey on XML-Based Policy Languages for Open Environments, *Journal of Information Assurance and Security*, 2006, pp 11-20