International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
International Conference on Communication Computing & Virtualization 2016

SemAuRSpec: A Semi-Automatic Approach of Specifying Functional and Non Functional Requirements using RDS (Requirement Description Schema)

Tejas Shah
M.Sc. (I.T.) Programme
Veer Narmad South Gujarat University
Surat, India

S V Patel
Sarvajanik College of Engg. & Tech.
Gujarat Technological University
Surat, India

ABSTRACT

Software Requirement Engineering demands a granular level of requirement specifications with key objectives, design constraints and relevant artefacts of a system. There exist some structured approaches of requirement specifications, 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 SemAuRSpec (Semi-Automatic Requirement Specification), a semi-automatic approach of eliciting and specifying functional non-functional requirement (NFR) using RDS (Requirement Description Schema). The approach is a competent way of managing and transforming requirement metadata and comprehensive artefacts of requirements like status, priority, version, stability, elicitation source etc. The aim of this approach is to improve the requirements elicitation and specification processes with partial automation. The system entails the DOM (Document Object Model) parser for parsing the XML oriented requirements of a system. The paper comprises of implementation of case study for specifying requirements of bank loan system

Keywords

Software Engineering, SemAuRSpec (Semi-Automatic Requirement Specification), RDS, Requirement Description Schema, Requirement Artefacts, Non-Functional Requirement, DOM (Document Object Model)

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, vague, imprecise or neglected requirements. Failure to specify the requirements correctly can lead to major delays, cost overruns. Good efforts have been made for investigation of alternative elicitation paradigms beyond untainted 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].

There are merits and limitations of the existing requirement elicitation methods, which suggest the conjunction of different techniques to attain accuracy in the requirements elicitation phase. The traditional requirement engineering commonly relies on questionnaires and other paper-based methods of requirement collection. Unfortunately, questionnaires often fail to capture inherent facets of customer tasks that may be identified through face-to-face interactions.

The interviews, JAD and brainstorming session for eliciting requirements are rich source of requirements, but these methods are time consuming since it requires customers and software designers to be co-located in time and space.

Requirements Specification is one of the trivial RE tasks during which elicited and analyzed requirements are properly documented for use by their envisioned stakeholders. 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 Requirements specified in Natural Language can often be ambiguous, incomplete, and inconsistent because stakeholder interaction is of vital importance in requirements elicitation with manual onsite or offsite feedback, it is almost impossible and impractical to fully automate the elicitation process of RE.

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 SemAuRSpec for structured requirements elicitation and specifications using RDS. 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 SemAuRSpec claims to offer benefits such as (1) Eliciting requirements efficiently with form based components of GUI interface (2) Representing complex and unstructured requirement components in the electronic and interoperable form. (3) Exchanging requirements amongst stakeholders, business analysts and developers in internal as well as in external environment. (4) Reduce ambiguities in Requirements (5) Management and traceability of requirements with the help of appropriate XML elements. This paper offers a semi-automatic approach for the transformation of functional and non-functional requirements of a system into XML specification. The goal of this approach is to integrate the requirement elicitation and specification to help requirement analysts and software designers in the system design process from XML repositories.

The paper is organized as follows: Section II entails relevant research in the field of requirement elicitation, requirement markup languages and specifications. Section III concerns the design structure of RDS specification. Section IV discusses the main components and architecture of SemAuRSpec approach. Section V further illustrates the specification approach with the case study of bank loan system. The last section comprehends the concluding remarks and future work aiming at the extension of SemAuRSpec.

2. RELATED WORK

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). The GRC Graphical Requirement Collector [10] provides better direct communication between software engineer and clients followed by questionnaire procedural methods of requirement gathering.

The AbstFinder is a prototype natural language text abstraction finder for use in requirements elicitation [11]. Deeptimahanti et al. in [12] describes a domain independent tool, UML Model Generator from Analysis of Requirements, which generates various UML models with categories of Use-case Diagram, Collaboration diagram and Design class model from natural language text using NLP tool.

The work mentioned in [13] developed a prototype tool to be used separately or jointly by customers, end users, software engineers, and domain experts with partial automation of requirement elicitation. The ElicitO [14], a requirement elicitation tool envisioned at empowering requirement analysts with a knowledge repository that supports to incarcerate precise NFRs specifications during requirement elicitation interviews. The work described in [15] investigated a dynamic approach to compress XML data using a hybrid compression tool, which allows the compression of XML data using variable and fixed length encoding techniques.

In the empirical study of how software architects deal with NFRs [16] shown that software architects did not use any definite tool for NFR management and NFRs were not often documented. An integrated approach for requirements elicitation called iREA has been described with the objective of specifying and documenting functional and technical requirements for new information systems such as bridge, hub and cockpit measured as the latest efforts in engineering [17].

There are only a few markup languages and methodologies available in the literature covering functional and non-functional requirement description along with requirement artefacts. During the previous 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 [18], however, it does not include specification and activities of NFR.

In [19] SRS template is represented in XML with the consideration of the object oriented environment. 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. As modelling requirements with use cases is proven useful, the authors Dimitris et al in [20] presented the structure of use cases with appropriate tags. The work mentioned in [21] represents the requirements in the graphical and tabular way to fill the gap between requirement documents and use case through SysML. The work mentioned in [22] 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 [23]. The paper based on RQML in [24] addresses the problem and solution of collaboration on managing and documenting the software requirement elicitation focusing on creation of elicitation assistant. The work mentioned in RQML, RGML do not cover all the metadata of the requirements. (For e.g. Requirement status, priority 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 only.

Some of the approaches facilitate requirement elicitation and specification with NLP which is time consuming and complex. It may be noted that none of the above approaches totally cover all aspects of requirement artefacts, functional and non-functional requirements. Some of the approaches facilitate requirement elicitation and specification with NLP which is time consuming and complex.

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 with SemAuRSpec.

3. RDS DESIGN STRUCTURE

The schema which is used in SemAuRSpec is described in this section. 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.

3.1 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 Figure1.

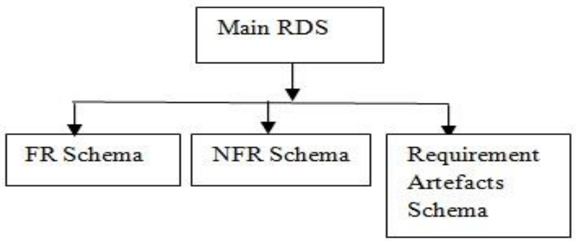

Fig 1: RDS Schema

3.2 Functional Requirement Schema

The functional requirement describes the desired key features of a system. As shown in Figure 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 behaviour 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.

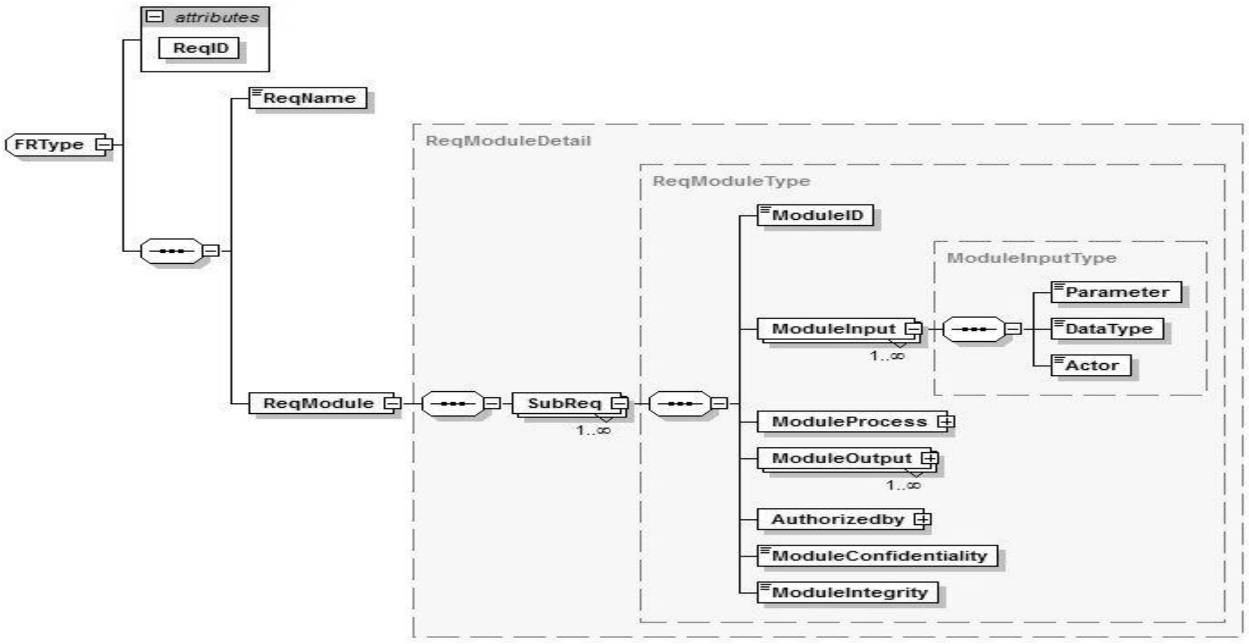

Fig 2: FR Schema

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: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>
```

3.3 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 Figure 3.

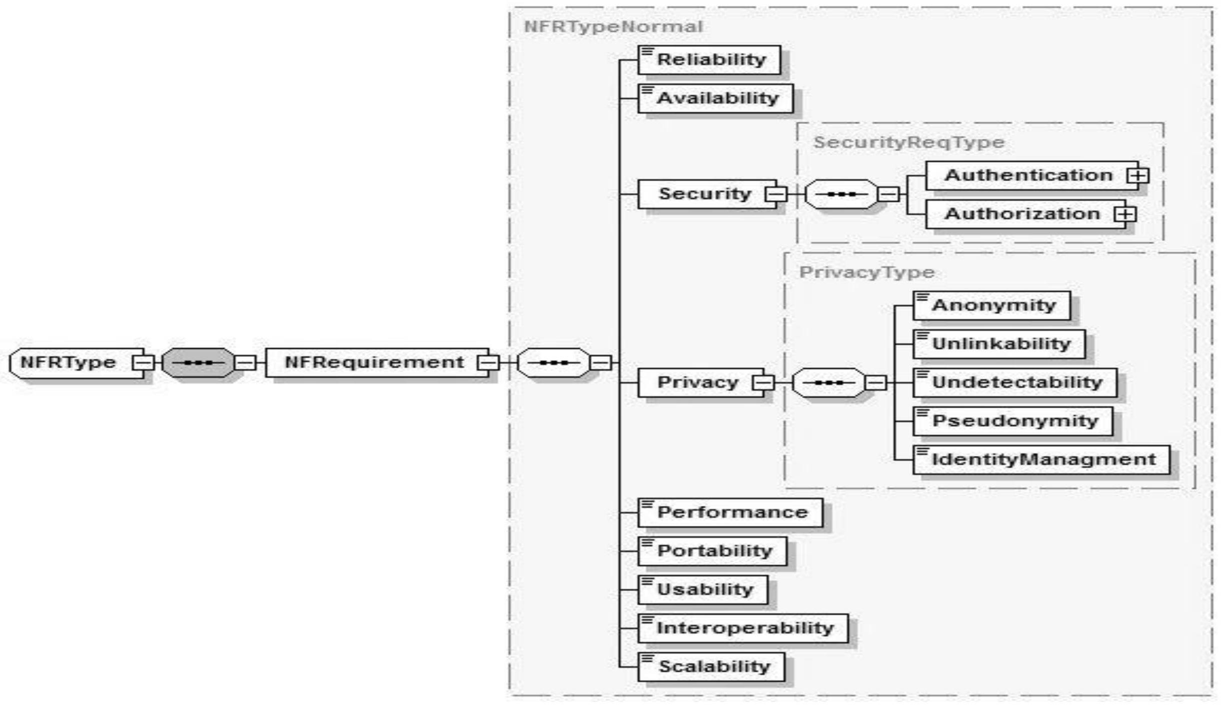

Fig 3: NFR Schema

Sample Element tags for NFR Schema

```
<xs:complexType name="NFRTypeNormal">
        <xs:sequence>
                <xs:element name="Reliability" type="xs:string"/>
                <xs:element name="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>
```

3.4 Requirement Artefacts Schema

The reuse of existing requirement artefacts makes the RE task more prescriptive and systematic as described in [25], 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 Figure 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 customer can store the keywords of the requirement and rationale also. The requirement source maintains the list of elicitation sources.

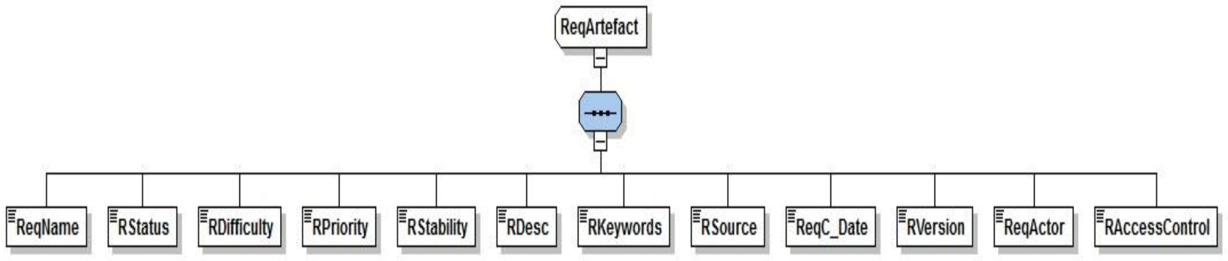

Fig 4: Requirement Artefacts Schema

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:sequence>
</xs:complexType>
<xs:simpleType name="ReqPriority">
        <xs:restriction base="xs:string">x
                <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="BrainstormingSession"/>
                        <xs:enumeration value="Scenario"/>
</xs:restriction>    </xs:simpleType>
```

4. SemAuRSpec ARCHITECTURE

The architecture of SemAuRSpec depicted in Figure. 5 give insights into transformation of requirements from participatory UI to XML files. The SemAuRSpec is having mainly three components for eliciting and specifying the requirements.

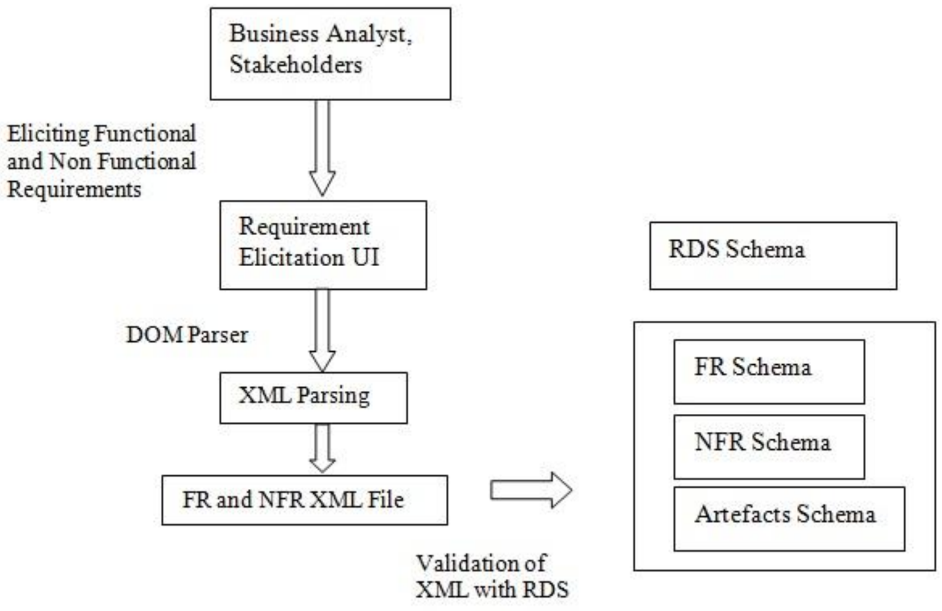
Business Analyst, Stakeholders
Eliciting Functional and Non Functional Requirements
Requirement Elicitation UI
DOM Parser
XML Parsing
FR and NFR XML File
Validation of XML with RDS
RDS Schema
FR Schema
NFR Schema
Artefacts Schema

Fig 5: SemAuRSpec Architecture

4.1 Elicitation with Dynamic UI
The XML based requirement specification gives abstraction to the stakeholder in mentioned requirements with efficient user interface. It's very hard to use a generic system for requirement collection; hence stakeholder (customer) concerns the business analyst in elicitation form design. This approach entails more stakeholder involvement in eliciting and specifying the requirements. Based on system's requirements and user needs, the business analyst can change the UI provided for elicitation. The form design is developed using Netbeans IDE 7.0[26].

4.2 Transforming Requirements with DOM Parser
The SemAuRSpec uses the Document Object Model (DOM), a type of tree-based API that allows users to generate a memory representation of an XML document [27]. This internal tree structure allows the users to navigate the tree of functional and non- functional requirements and retrieve information contained within the elements of the requirement schema. The DOM parser nodes are the memory equivalent of elements and have the tracing facility of parsed data. The elicited requirements are stored into XML repositories, as the requirement document needs to be loaded completely into memory and accessed by the analyst and stakeholder multiple times. The SAX parser operates sequentially without being able to keep track of parsed data [15]. The transformation process is implemented in Netbeans IDE 7.0[26].

4.3 Validation of XML files with RDS Schema
The SemAuRSpec generates multiple XML files pertaining to functional and non-functional requirements with metadata preservation of requirements of a system. The XML schema works as a template in backend. After the validation phase, the XML repositories can be used for querying the requirements with XQuery mechanism. Moreover, the selection and discovery process of relevant web services can be done from specific XML files.

5 CASE STUDY AND RESULT
To specify the requirements using SemAuRSpec, a case study of a typical bank loan system is presented. The scope of specifying requirements is kept limited to home loan only. The Table 1 includes the columns like actor, module name and XML elements used in specification. The validity and well-formed property of RDS Schema has been checked in trial version of AltovaXMLSpy Editor [28].

Table 1. Key Requirement Elements for Bank Loan System

No.	Actor	Module Name(Requirement)	Schema	XML Elements
1	Borrower (Customer)	Request Loan	FR Schema NFR Schema	ReqModule: Input Parameter: Customer Detail, Property Detail, Loan Detail Output Parameter: Request Accept/ Reject Security:Authentication, Authorizatioin
2		Provide Security Papers	FR Schema NFR Schema	ReqModule: Input Parameter: Customer Detail, Property Detail, Loan Detail Output Parameter: Papers Received Security: Authentication, Confidentiality Privacy: Identification, Unobservability
3		Pay EMI	FR Schema NFR Schema	ReqModule: Input Parameter: Loan Detail, Payment DetailOutput Parameter: Security: Confidentiality, Integrity: Required Privacy: Anonymity, Identification
4	Bank Officer	Check Security Papers	FR Schema NFR Schema	ReqModule: Input Parameter: Property Detail, Loan Detail, Security Paper Detail Output Parameter: Paper Verified, Customer Security: Authentication, Confidentiality Privacy: Identification, Unobservability
5		Process Loan Request	FR Schema	ReqModule: Input Parameter: Property Detail, Loan Detail Output Parameter: Customer, Accept-Reject Status
7	Loan	Interest Parameter	FR Schema	ReqModule: Input Parameter: Loan Detail, Interest

	Processing System	Selection		Provision, Different Plan Output Parameter: Interest Finalization
8		EMI Calculation	FR Schema	ReqModule: Input Parameter: Customer Detail, Loan Detail, Interest Plan, Loan Tenure, Branch Detail Output Parameter: EMI Detail, Payment Detail
9		Loan Payment	FR Schema NFR Schema	ReqModule: Input Parameter: Loan and Customer Detail Output Parameter: Payment Status, Update in LoanSecurity: Authentication, Confidentiality Privacy: Identification, Authorization, Anonymity

6 CONCLUSION AND FUTURE SCOPE

Acquiring and specifying requirements from end users with SemAuRSpec garnered valuable information and technical insights to the system designer and system analyst. The SemAuRSpec approach revealed the initial design of obtaining user requirements with RDS (Requirement Description Schema) specification suitable for the use by requirement engineer, business analyst, service consumers and naive customers.

In future, the advanced architecture of SemAuRSpec can be extended to the application of Service Oriented RE with features of evolutionary elicitation process, preventive NFR processing, requirement data exchange, tagging and integration. SemAuRSpec tool can extract potential domain knowledge from interactive RE interface and transform the requirements to appropriate web services and thus change the way of designing RE system in the near future. Another direction is development of algorithm for mapping requirement metadata to discover web services automatically.

7 REFERENCES

[1] Sommerville I (2004) Software engineering. 7th edn. Addison-Wesley, Harlow

[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

[10] Moore, J. Michael, and Shipman III, Frank M., "A Comparison of Questionnaire-Based and GUI-Based Requirements Gathering", The Fifteenth IEEE International Conference on Automated Software Engineering, 2000.

[11] L. Goldin, D.M. Berry, Abstfinder: A Prototype Natural Language Text Abstraction Finder for Use in Requirement Elicitation, Automated Software Engineering Journal, 4(4), October 1997, 375-412.

[12] Deeptimahanti, D. K. and Babar, M. A., An Automated Tool for Generating UML Models from Natural Language Requirements, IEEE / ACM Int. Conf. on ASE, 2009

[13] N.W. Kassel, B.A. Malloy, An approach to automate requirements elicitation and specification, in: Proc. of the 7th IASTED International Conference on Software Engineering and Applications, Marina Del Rey, CA, USA, 3–5 November 2003.

[14] T.H. Al Balushi, P.R. Falcone Sampaio, D. Dabhi, P. Loucopoulos, ElicitO: a quality ontology-guided NFR elicitation tool, in: Requirements Engineering: Foundation for Software Quality, LNCS, vol. 4542, Springer, 2007, pp. 306–319

[15] Antonio Kheirkhahzadeh, "On the performance of markup language compression", Doctoral thesis, University of West London. 2015

[16] David Ameller, Claudia Ayala, Jordi Cabot, Xavier Franch, Non-Functional Requirements in Software Architecture Practice, Report ESSI-TR-12-1, Departamentd'Enginyeria, March, 2012

[17] FrédéricDemoly, DimitrisKiritsis, An integrated requirements elicitation approach for the development of data management systems, PLM11 8th International Conference on Product Lifecycle Management, IFIP Working Group 5.1, 2011

[18] Ahmed Sidky, RGML: A Specification Language that Supports the Characterization of Requirements Generation Processes, Master thesis, Virginia Tech, 2003

[19] KenzaMeridji, Documentations and validation of requirements specifications – An XML approach,Other Thesis, Concordia University, 2003,

[20] D.Dranidis, K. Tigka, Writing Use Cases in XML,9thPanhellenic Conference in Informatics, 2003.

[21] 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

[22] Shreyaschavda, Dr. ShantharamNayak, Modern Technique To Build Software Requirements Specification, IJSRD - International Journal for Scientific Research & Development, Vol. 2, Issue 03, 2014 | ISSN (online): 2321-0613

[23] SasmitoAdibowo, Rambutan, Requirements Management Tool for Busy System Analysts, Technical Report, July 2003

[24] Eko K. Budiardjo, SasmitoAdibowo, RQML Based Mobile Requirement Elicitor Assistant, SITIA (Seminar on Intelligent Technology and Its Applications), Surabaya, May 9th – 10th, 2007

[25] 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

[26] URL: https://netbeans.org/downloads/7.0.1/

[27] Nicol, G., Wood, L., Champion, M., and Byrne, S. (2001). Document object model (DOM) level 3 core specification.

[28] URL: http://www.altova.com/download-trial.html