Towards a Multi-views Approach for Software Requirement Engineering: Requirements Management Tools

Omer Dawood, Abd-El-Kader Sahraoui

Towards a Multi-views Approach for Software Requirement Engineering: Requirements Management Tools

Omer Salih Dawood[1], **Abd-El-Kader Sahraoui**[2]
[1] Department of Computer Science, College of Arts and Science, Wadi Aldawasir
Prince Sattam Bin Abdulaziz University, KSA
Sudan University of Science and Technology, Khartum, Sudan
o.dawood@psau.edu.sa, omercomail@gmail.com
[2] LAAS-CNRS, Université de Toulouse, CNRS,UT2J, Toulouse, France

ABSTRACT
This paper is on Requirements methods and associated tools. It provides general overview of requirement engineering approaches and propose a multi-view approach, and deep comparison on tools and techniques used to manage requirements. The comparison is based on five items, requirement traceability, requirements integration, requirement prioritizing, requirement status, and customer satisfaction. By this comparison it becomes easy to assist the requirement tools and techniques. The paper concentrates on tracing requirement and proposes to develop a model that can be used to handle and manage requirement traceability on large and complex systems.

Keywords
Requirement Engineering; Traceability; requirements tools RTM; DOORS.

1. INTRODUCTION
Requirement engineering is first step in software development. It has many steps elicitation, analysis, and requirement management validation. Its aimed to collect and managed the requirements in a good manner and best way to ensure that all requirements are gathered and analyzed in the way that allow to produce both products and services that satisfying quality attributes [1]. Requirements management (RM) is process of managing changes in the requirements throughout procedure of requirement engineering. Requirement management contains activities related to identification of change, maintenance of change, traceability and changes management of requirements [2]. Requirements management Tools are used to manage the requirements, and shows the relationship between the requirements, and so on. The following section introduces a tools and technologies used to manage the requirements and performs a simple comparative study between three tools DOORS, RTM, and Volere, by this comparative study we want to specify the properties and ability of each, and the problem of each one so that we can enhance the tools and methodology to produce a tool that satisfy the all quality attributes, and we provides multi-views for software requirement engineering, and idea for research in this area. The paper concentrates on requirement engineering traceability.

2. Volere Requirement Specifications:
Volere is used for requirement specifications. It is developed to manage and trace the requirements using volere shell. The shell consist from main attribute that needed when specifying requirement like requirement Id, type, priority, and dependency between requirements.

Volere divides the requirements to the five types each type has the sub components as in the flowing figure (1).

1. Functional requirements are the Basic functions of system that perform core operations of the system. The concrete means are used to measure these functional requirements.
2. Non-functional requirements are the properties that the specify behavior of function such as usability, look and feel, performance, etc.
3. Project constraints describe how the product delivered and fit into the world. The constraint in involves many things like needed interface with existing component for both software and hardware, practice of business, and budget defined at starting project or be ready by a defined date [3].
4. Project drivers are the related forces of business, that drives or control on the process of project development. The purpose of the product is project driver, for all stakeholders in the system in different levels and reasons.
5. Project issues define the conditions under which the project will be done[3]

3. Dynamic Object Oriented Requirements (DOORS)
It is the requirements management tool developed by Telelogic to support the software engineering lifecycle. DOORS is mentioned in several papers and is often referred to as very capable requirements management tool[4].It allows many Users to work together at the same time, and enabling access to the database server that contains information about requirement and links[4].

DOORs database contains many project and each project has its own users and information model. The information model contains a set of module used to keep information about actual requirements and link. DOORs has main three modules:

1. Formal modules Requirements has many artifacts, the artifact contains smaller object. Formal module used to store information about representation of requirement artefact.
2. Link module each formal module has relationship; this relationship is stored in the link module.
3. Descriptive module this module not used .basically to store actual requirement, but now actual requirement stored in formal module [4].object consists from the following [5]

1. General used for describing heading, short text, object text values for the object
2. Access is manage access right to the object
3. History is used as log for changing in object
4. Attributes: is value of object attributes
5. Links: are used to handle the relationships with other objects

4. REQUIREMENT TRACEABILITY MATRIX (RTM)

The requirement engineering has two parts, fist one is requirement development, this part is responsible for requirement elicitation, analysis, specification, and validation. Software Requirement Specification (SRS) is document that produced as output of the requirement development. It contains the requirement specification and it ready to the design phase .The second part of requirement engineering is requirement management, which is responsible for managing requirements and it has two part change management and traceability. Traceability process produces Requirement Tractability Matrix (RTM) as output [6]. The RTM handles requirements and relationships between these requirements in a single document [7].

REQUIREMENT TRACEABILITY

Traceability recognizable association between two or more logical entities like requirements, verifications, elements of system, or tasks. The main two types of traceability are horizontal and vertical traceability but there are other sub types [6].

Vertical Traceability: it shows the source of items and traces these items to Work Breakdown Structure (WBS), to project team and finally to the customers. It insures that the requirement can be traced till satisfied [6].

Horizontal Traceability: It shows the relationship between the related item and work group. It is aimed to avoid the conflicts.

Bidirectional Traceability: It is an association between two or more logical entities that is discernible in either direction. It effectively manages the relationship between requirement sources and requirements of product and its components. In other meaning bidirectional traceability happens between requirement to end product and vise versa [6].

Indirect Traceability: There are two directions for traceability. Firstly is Forward traceability, trace the requirements to its source. Secondly is backward traceability is trace to from product to requirement source [6]. Requirement Traceability Matrix (RTM): Is matrix that used to handle the complete user and system requirements, or a part of the system.

Requirement traceability is helpful in software engineering activities like validation of requirements, and impact analysis, also is useful in tracking the logical sequences and trade-offs for each Requirement.

No.	Comparison Topic	Volere	DOORs	RTM
1	Requirement Traceability	Known as dependability	Support different traceability types	Support different traceability types
2	Requirements Integration	No requirements integration	Support integration, on different levels	Not fully Supporting integration
3	Requirement Prioritizing	No priority	Support priority	Support priority
4	Requirement Status	No requirement status	Has requirement status	Has requirement status
5	Customer Satisfaction	Has customer Satisfaction	___	Depend on RTM Application

Table (1): Comparison of requirements tools and techniques

5. RELATED WORK

Renuka and et al [8] designed a novel methodology to design traceability of onboard software known as Software Requirements to Design Traceability Technique (SoRDeTT). Their methodology is based on two templates, Software Requirements Specification (SRS) and Software Design Document (SDD) as input to the methodology. SoRDeTT represent a common template for both requirement and design which is used to handle the data and information from these documents. There are two trace items; SRS Trace Item (SRSTI) is the template that populates data from SRS. The other one is SDD Trace Item (SDDTI) which if filled with data from SDD. This methodology applied to satellite system because it's complex and contains many subsystems. onboard software requirements and software design of many subsystems Represented in SRS and SDD. The main purpose of this methodology is to ensure the software design is done according to SRS. The methodology works as follow, data captured from SRS to build SRSTI, SDDTI, comparing SRSTI and SDDTI, if one to one mapping is not found then make tag mismatch, then generate mismatch report, analyze the inconsistency to make a correction, and finally repeat SoRDeTT for all new changes.

Filho and et al[9] propose traceability framework. The model aimed to visualizing the traceability among different tools , and they assume the models are represented in XML to support heterogeneity of tools and models, so that XML is became de facto standard for data interchange, XML supported by many tools, and they use XQuery as standard for traceability rules expression. They assume the model is generated in Native Format Models and converted to (XML_based Models) by using Model Translator.XML model and rules used as input to Traceability_Completeness_Checking Engine.

The traceability relations between the models are generated by the engine, also the sngine identify missing elements based on the rules. Engine uses WordNet component to ensure the identification of synonyms between the of element's names in the models. Traceability_Relations_Missing_Elements document is used to handle the traceability relationship and identified missing elements. This document is important because preserve the original models, to let the use of these models by other different applications and tools. The document used as input to Traceability_Completeness_Checking Engine component that used to support generation of dependent traceability relations. They developed simple prototype tool that shows the traceability relations, and they showed that there are many traceability relations can be generated .

Fab´ıola and Michel [10] extend the SysML requirements diagrams concentrating on traceability of both functional and non-functional requirements. Real-Time Systems can be modelled by this extension of the requirements. The metamodel of SysML is extended with new relationships and stereotypes, and applying the specification of a Road Traffic Control System using proposed metamodel is applied to a set of requirements for the specification of a Road Traffic Control System. SysML is a UML profile and Class diagram stereotype extended new attributes. It decomposes the requirements into smaller related elements in form of hierarchy so that the complexity is managed early. The hierarchy is based on master and slave who allow reusing the requirement. They propose seven new stereotypes to extend the relationships like, copy relationship is represented by master/slave relationship, derive relationship (deriveReqt), satisfy requirement shows how model satisfies requirements, a test case, represented by verify relationship, refine relationship show how a model element used to refine a requirement. The trace relationship act as a general purpose relationship.

6. Ontology for requirements elicitation

Such work based on research roadmap in systems engineering [11] that will be integrated to partially in our work as requirement ontology if well defined and formalised can give rise to better requirements tools . This can make changes to Volere requirements templates.

Problem Definition It is very important to build requirements elicitation on the form used for requirements expression. With the evolution of the Internet and electronic commerce, future business services will often be delivered by autonomous and collaborating parts or software agents inside or across organizational boundaries through negotiation and information exchange over a distributed data network. Efforts are needed to develop collaborative requirement engineering t as an associated need.

The semantics of different information sources are collected by their ontologies, i.e., both terms and the relationships between these sources. In many applications, the intended meaning of a term is often implicit, and understanding this in a collaborative environment necessarily is reliant upon mutual agreements and understandings. In an open environment mutual agreement is hard to achieve. Thus it is very important for the vocabulary, that describes the domain model, to be specified and maintained in such a way that other systems can process them with minimum human intervention. Ontology is used to manage and deal with this task. The ontology research now has more attention from both academia and industry.

It is generally very difficult to build a machine-definable ontology (vocabulary). The semantics of a term varies from one context to another and across different stakeholders. Ideally we need an approach that reduces the problem of knowing the contents and structure of many information resources to the problem of knowing the contents of specific domain of ontologies that user familiar with the domain and easily understands.

Not all requirements are known at starting of the system development. They cannot be specified completely up front in one voluminous document. But rather will evolve during analysis phases of a project and beyond. requirements elicitation involve all stakeholders: users, developers and customers; all see their way matured in the way the requirements are expressed from this step till maintenance; such acquired added value by the elicitation is used to improve the system instead of maintaining the myth that the requirements are to remain static.

Requirement elicitation is one of requirement engineering process. It represents one of the first critical phases. Requirement process is the first phase in systems development. The specific nature of such process is that the word "elicitation" is new as a technical term; its equivalent does not even exist in some natural languages as French; we use some times the close definition: capture or acquire requirements. Such phase was often neglected as the requirements were only expressed; effectively, we considered were considered to be available and needs only to be expressed. For long period till the 90's the research community was focussing on notation, methods and languages for expressing requirements. The debate was best to use for the sake for genuine expression and also for validation and verification. The debate was transported on the formal versus semi-formal specification of requirements. Most RE researchers have been concerned by such work on taxonomy of methods and adequacy of such methods and notations for expression requirements for various types of applications: in formation, systems,

The requirements elicitation is one of the most important steps in requirements engineering project. Experience over the last decennia has shown that incorrect, incomplete or misunderstood requirements are the most common causes of poor quality, cost overruns and late deliveries. The ability to use an adequate approach thought a method or systematic process is therefore one of the core skills in systems development. The GAO survey is a demonstration through figures on nine projects totaling about $7 millions.

A terminology (CMU): The procedure of understanding systems requirements can be defined and described by many terms. Requirements engineering can be used as a general terms including all activities related to requirements. In fact, requirements engineering consists from four specific processes

Requirement elicitation: Is first process allowed to understand, discover, reveal, and articulate the requirements to customers, buyers, users of a system.

Requirements analysis: This process is based on the requirement elicitation. It is reasoning the elicited requirements; it involves some activities such as checking requirements to ensure from both conflicts and inconsistencies, combining requirements that related to each other's, and specify missing requirements.

Requirements specification: In This process the requirements are recoding in the forms, this including the may be done in natural language, symbolic, formal, and diagrammatically representing the requirements, also the product that is the document produced by that process.

Research approach. The suggested research approach involves development of a shared ontology: A shared ontology can be in the form of a document or a set of machine interpretable specifications. Among possible contemporary research projects that deal with ontology-based approaches to resolving the semantic issues, the following seem especially appealing.

Requirements validation: In this process the requirements are confirmed with the users of systems, and customers to ensure that the specified requirements are valid, complete and correct.

In an actual situation, these four processes cannot be strictly separated and performed sequentially; they are interleaved and performed iteratively.

The term elicitation is not universally accepted for the process described; there are no similar term in other language, in example French language; the term acquisition, capture is often used; some companies use gathering, expressing, formulating. Each term has a different connotation. Acquisition supposes the requirements are already there like sensor value acquisition by I/O system of a computer system. Apart from the term used, all of these terms address implicitly the elicitation term.

i. Common domain model: although participating agents share a common domain that is the basis of their cooperation, they often have different views of the domain. In order for them to collaborate, a common domain model is required to facilitate their communications.
ii. Different levels of abstraction: different levels of information abstraction are required by a flexible enterprise. At the agent level, only high-level business process and service concepts are needed to form service level agreements, i.e., contracts. At the task scheduling level, processes and services must be viewed in term of individual tasks and task interfaces (methods and conditions). At the execution level, data representation must be explicit so that data can be transformed and fused correctly.
iii. Dynamic information integration: the underlying information systems are potentially large. New services may require only parts of the information systems to be integrated. Dynamic information integration is required as which parts to be integrated for what purposes cannot be determined beforehand.
iv. Service and contents description: agent services and information system contents must be formally described. The descriptions must be accessible and meaningful to all participating agents.
v. Information heterogeneity reconciliation: as flexible enterprises operate in an open environment, participating agents often use conflicting terms. In order for them to collaborate, the heterogeneity must be reconciled.

Expected results. The suggested research should result in several needed and useful outcomes.

i. Developing a requirements domain ontology environment for effective and efficient requirements elicitation will represent a considerable advance in requirements engineering. This will necessarily involve identification of appropriate support environments needed to assist ontology designers with the tasks involved in ontology management. It is

ISBN: 1-60132-446-4, CSREA Press ©

envisaged that such an environment would maintain an ontology repository that can be accessed. During the design phase to enable this, tools will be available to browse and reuse the terms from the repository. When new terms need to be added, checks should be performed to see that they do not cause inconsistency in the repository. This environment should also have a set of tools that help extract ontological information that is embedded in existing systems.

ii. Develop appropriate methods and tools to support the integration of process models and information systems from multiple organisations during requirements change.

iii. Extending XML in requirements for data sources and ontology extraction and retrieval. Integrate the ontology for requirements elicitation into a general framework and context to support systems engineering in a computer supported cooperative work environment.

7. RESEARCH OBJECTIVE

1. Better understanding of requirement management tools and techniques.
2. Evaluate mentioned tools to become easily when requirements management tool is needed.
3. Improving software quality by determine each tool capabilities to minimize the problems risk of requirement management.
4. Evaluate requirement traceability in each tool and techniques.

8. RESEARCH PROBLEM

From literature review there are many researches in requirement traceability but not detailed covered the one of the important topic in requirements validation and verification through traceability. It is expected to develop a requirement traceability model or tool that allows enhancing and improving software quality through tracing requirement in a good and best manner.

9. DISCUSSION

Requirement engineering is first step of software development its aim to collect and document requirements. The cost of detecting and managing errors in earlier stages is less than detecting errors in later stages. Traceability is very important to handle and managing requirements, because its allows easily tracking requirements. There are many research covered this area but still some gaps and missing are found. The previous research concisely covered the traceability issues and concentrate on small to medium systems. This research aimed to full the missing points on the previous studies and develop a model for requirement traceability that allow handle the traceability in big and complex system.

By developing the new model its expected to produce highly and well requirement modeling and techniques that produce software with high level of quality, and requirements of the complex system can be managed in easily way.

10. CONCULISION

This paper covered the concept of requirement engineering and comparing between some of the current tools and techniques that used to manage requirements. In the area of requirements engineering we concentrate to requirement traceability because it very important to manage and handle the requirement. Many previous researches are reviewed and concluded in this paper, and some interested areas are shown and discussed .this paper introduces some research problems like requirement traceability and requirement ontology. Its expected to solve the problem of requirements elicitations verification and validation through requirements traceability.

11. ACKNOWLEDGMENTS

The authors and mainly second author are indebted to many colleagues who contributed directly or indirectly to this work and mainly Late Professor Andy Sage from Georges Mason University and Professor Dennis Buede from New Jersey University.

12. REFERENCES

[1] Hummera, Yasir, Sohail, Muhammad, Asma, 2013.Effective Usage of AI Technique for Requirement Change Management Practices. 5th International Conference on Computer Science and Information Technology CSIT (Aman, 27-28 March 2013).978-1-4673-5825-5, 2013. DOI: 10.1109/CSIT.2013.6588769

[2] Dhirendra, U. Dhirendra, A. K. Ramani. 2010. An Effective Requirement Engineering ProcessModel for Software Development and Requirements Management. 2010 International Conference on Advances in Recent Technologies in Communication and Computing(Kottayam. 16-17 Oct. 2010) DOI: 10.1109/ARTCom.2010.24

[3] James and Suzanne Robertson. Volere Requirements Specification Template. (London, Aachen & New York).DOI: http://homepages.laas.fr/kader/Robertson.pdf

[4] B.J.M. Abma. 2009. Evaluation of requirements management tools with support for traceability-based change impact analysis.Master Thesis, University of Twente.

[5] Vassilka Kirova and Darshak Kothari. 2004. Getting the most out of DOORS for requirements management. Technical Report. Software Technology Center, BLAT.

[6] Surachet and Yachai.2010. Enhancement of Requirements Traceability with State Diagrams. 2010 2nd International Conference on Computer Engineering and Technology (ICCET) (Chengdu. 16-18 April 2010) DOI: 10.1109/ICCET.2010.5485417

[7] Software Testing-Requirements Traceability Matrix.DOI: http://www.etestinghub.com/requirements_traceability_matrix.php

[8] Renuka and et al.2014. NOVEL METHODOLOGY FOR REQUIREMENTS TO DESIGN TRACEABILITY OF ONBOARD SOFTWARE. 2014 International Conference on Advances in Electronics, Computers and Communications (ICAECC), (Bangalore, 10-11 OCT 2014)DOI: 10.1109/ICAECC.2014.7002386

[9] Fab´ıola and Michel. 2013. A Metamodel for Tracing Requirements of Real-Time Systems. 16th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC 2013)(Paderborn, 19-21 June 2013).DOI: 10.1109/ISORC.2013.6913189

[10] Gilberto and Maria. 2012. Towards a Traceability Visualization Tool. 2012 Eighth International Conference on the Quality of Information and Communications Technology. (Lisbon, 3-6 Sept. 2012).DOI: 10.1109/QUATIC.2012.60

[11] SAHRAOUI and et al. 2008. SYSTEMS ENGINEERING RESEARCH. Journal of Systems Science and Systems Engineering.(16 July 2008). DOI: 10.1007/s11518-008-5083-9