

Managing Complexity in Requirement Elicitation

Abd-El-Kader SAHRAOUI (1,2)

1 CNRS, LAAS, 7 avenue du Colonel Roche, F-31400 Toulouse, France

2-Univ de Toulouse, UTM, LAAS, F-31100 Toulouse, France

E-mail: sahraoui@laas.fr

Abstract : an approach for managing complexity through requirements elicitation and traceability model. The approach makes use of two approaches development. The first on a traceability model that have been developed and applied to a complex industrial system and a requirement elicitation approach and tool. The paper makes use of these approaches and show how a traceability even at requirement elicitation can handle complexity and may be expanded to the whole lifecycle of system development

Keywords : elicitation, requirements, complexity, traceability

1. Introduction and problem statement

The complexity in systems development is observed when linking artifacts between themselves , these artifacts items can pieces of requirements, properties, pieces of design , stakeholders. We address two issues that were used separately in previous studies. The first issue is part of requirements engineering as the first subprocess.: Requirements elicitation to make difference with the requirements acquisition.

The second concerns part of requirements managements named traceability. There are two types traceabilities ans syntactic links between items and the more semantic based concerns the coverage that item as design does Implement the re stated requirements

2. Requirement elicitation approach

Requirements elicitation can be broadly defined as the activities, typically performed after project initiation and before system design, related to the acquisition and elaboration of goals, constraints, and features for a proposed system, by means of investigation and exploration. Furthermore, it is generally understood and accepted that requirements are elicited rather than just captured or collected. This implies both a discovery and development element to the process. In practice requirements elicitation is often performed poorly, the major reasons being inadequate expertise on the part of the participating requirements engineer, and the insufficient allocation of time and resources within the larger development project. The consequences of this situation frequently include costly rework, schedule overruns, project failure, poor quality systems, and general stakeholder dissatisfaction [3].

In response, much of the relevant research performed over the past two decades has focused on the development of numerous techniques for requirements elicitation as surveyed in [1], and more recently in [8]. Of these, one of the more successful in producing quality requirements has proven to be facilitated workshops [2]. However, most projects typically require more than one technique to be used for requirements elicitation [6]. Furthermore, a major problem in requirements elicitation today is the significant gap between expert and novice analysts. A lack of awareness by analysts of the state of the art techniques and tools for requirements elicitation, combined with a general unwillingness to adopt them is largely responsible for this situation. This situation is further aggravated by the current shortage of systematic guidelines and flexible methods.

Subsequently the work described in this paper investigates how an improved approach for the early stages of requirements elicitation can be developed that combines various techniques based on a detailed information meta-model and process framework for collaborative workshops. The approach was developed based on an extensive literature review, seven structured interviews with practitioners widely regarded within the Requirements Engineering community as elicitation experts, and a review of requirements related documentation produced within fifteen successful system development projects. The paper is therefore structured as follows. Section 2 describes the information types contained in the meta-model used as the foundation of the approach. Section 3 presents the approach with an overview of the structure, content, and process. Section 4 offers a broad discussion of the approach, and finally Section 5 provides some general conclusions on the research.

A meta model for requirements elicitation

The foundation of the proposed approach is based on a knowledge meta-model consisting of a specified set of information types with corresponding attributes. As the name implies, information types are different types or categories of information or knowledge that must be addressed during the requirements elicitation phase of the software development lifecycle in order to collect and capture all the necessary details to produce a quality requirements specification document. As can be seen in Table 1 below, fifteen ‘core’ information types have been identified from the review of current theory and practice as being relevant to most application domains, and typically necessary in most software engineering projects.

Table 1: The Fifteen Core Information Types

<i>No.</i>	<i>Title</i>	<i>Description</i>
1	Project	Problem, mission, vision, context, and scope of the project
2	Deliverable	Desired result of the process, its audience, objectives, and overview
3	System	Background, perspective, context, and scope of the system
4	Objectives	Objectives of the business with respect to the project and system
5	Assumptions	Underlying assumptions upon which the project and system are based
6	Constraints	Constraints that must be applied to the project and system
7	Environment	Social and physical environmental characteristics of the project and system
8	Opportunities	Possible opportunities for improvements to be delivered by the system
9	Challenges	Possible challenges which may be encountered during the project
10	Risks	Potential risks to both the project and the system
11	Stakeholders	Stakeholders in the project, and sources of system information

12	Processes	Detailed work process which the system must support
13	Functional	Functional aspects which must be provided by the system
14	Non-functional	Non-functional aspects which must be provided by the system
15	Implementation	Implementation details relating to the system including design solutions

Information types can have multiple levels of detail, and relationships between these different information types and levels, such as linking individual non-functional requirements to system objectives, are handled by specific attributes within the template for those information types. Table 2 provides an example of a template for one of the core information types. Within the approach, an individual template has been developed for each of the information types with specific attributes and instructions.

Table 2: Information Type Template Example – ‘4. Objectives’

<i>Attribute</i>	<i>Description</i>
ID	Unique numerical identifier for the objective
Name	Unique textual name for the objective
Description	Detailed description of the objective
Type	Classification of the objective selected from a standard list or specified by the analyst
Source	Source of the objective, possibly a document, a person, or an organization
Rationale	Justification for the objective in terms of reasons for its inclusion
Priority	Importance of the objective selected from a standard rating or specified by the analyst

In order to promote a more rigorous approach and resultant document from the requirements elicitation process, a number of additional information types are required to provide all the necessary support information for the knowledge elicited for the core types. These can be seen in Table 3 below.

Table 3: The Seven Support Information Types

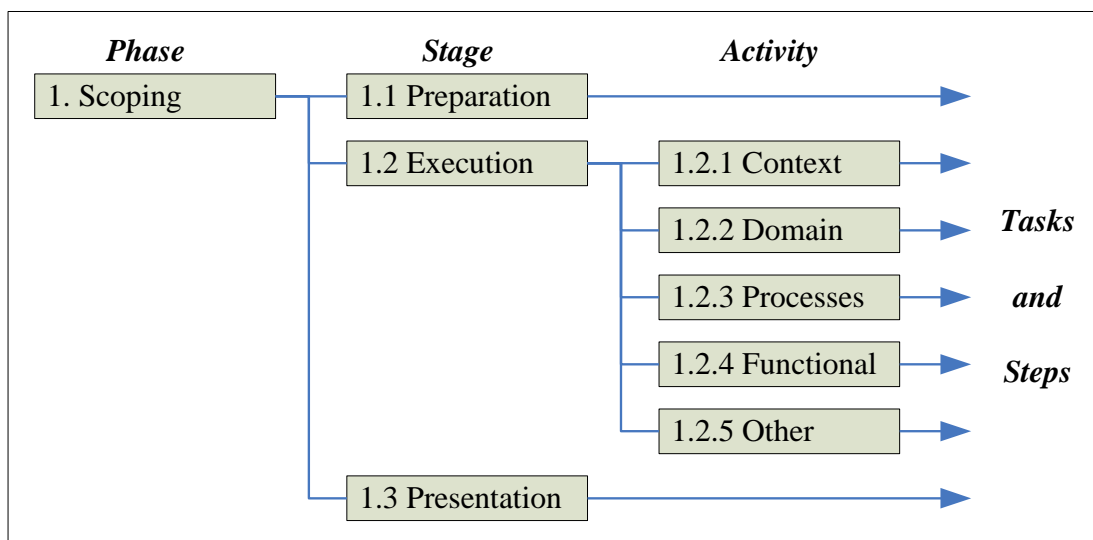
<i>No.</i>	<i>Title</i>	<i>Description</i>
1	Glossary	Definition of terms, abbreviations, and acronyms

2	Dictionary	Data definitions relevant to the system including type, size, and format
3	Issues	Prioritized list for project and system related issues
4	Actions	Prioritized list for project and system related actions
5	Ideas	Possible suggestions and potential solutions related to the project and system
6	References	Cited references made to information in other documents and sources
7	Appendixes	Required appendixes for the resultant document

The combination of these information types, as well as additional ones that may be specified by the requirements engineers based on the needs of the individual projects, form the information meta-model used as the foundation for the workshops and guidelines.

A guided requirements elicitation workshop

The proposed approach consists of three key workshop phases being 1) Scoping, 2) High-level, and 3) Detailed, as explained in the following sub sections. As can be seen in the example of the Scoping phase shown in Figure 1 below, the Execution stage of each phase is divided into five activities. These activities, as well as the Preparation and Presentation stages for each phase, are composed of a set of tasks in a prescribed sequence (100 tasks in total for all 3 phases). The steps for these tasks, being the next and final level in the process hierarchy, are determined by which of the techniques within the approach is selected to perform that particular task.

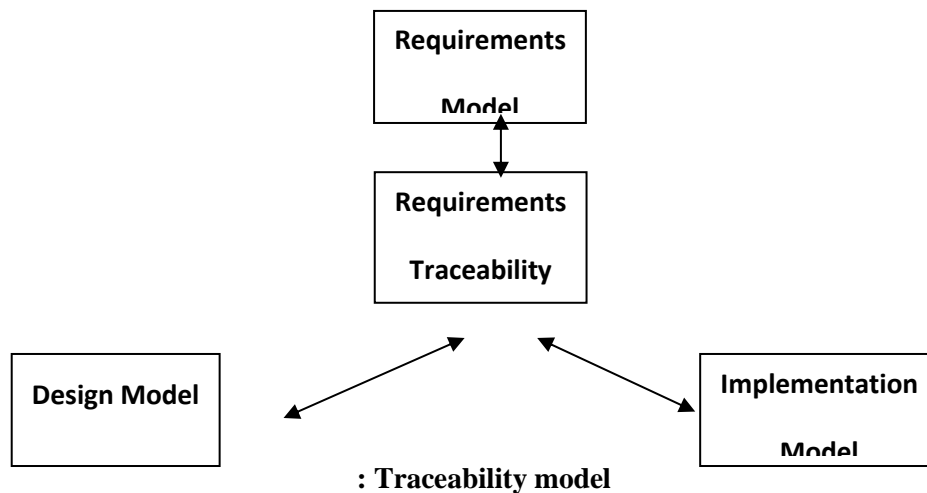


Structured Workshop Process Hierarchy – ‘1. Scoping Phase’

Each of the phases may be completed over a number of sessions depending on the complexity of the project, and the availability of the relevant stakeholders, facilitated by a requirements engineer, also referred to as the analyst. Furthermore, the same information type may be addressed by more than one task in different stages. In these cases the level of detail investigated and the attributes elicited are different but complimentary. Each task has one or more ‘available’ techniques, meaning that established instructions exist within the approach as a set of sequences steps for that technique, which can be utilized to perform that particular task within a workshop environment.

3. A general Traceability model

What is prone here is the separation of concerns principles, as the model can be made generic for new systems and enhanced for existing systems. The approach to be discussed is illustrated with the following figure



Such a reference model is discussed later in the paper. The main idea behind such a class diagram is to set the main essence of traceability that covers not only the requirement models in terms of basic and refined requirements but also others models as for implementation and design. The other advantage is to ease the traceability implementation in any tool based on object analysis.

Requirement traceability deals with tracing requirements at two orthogonal aspects.

The first aspect is in the requirement refined/derivation up and down. It means low level requirements (child requirement) can be traced back to at least a high level requirement (ancestor or parent requirement). This traceability is denoted through abstraction. On the contrary, requirements induced through refinement by a high level requirement can be traced from it. Every requirement has an identified origin (source) : it can be another requirement or coming from the external context known as stakeholders, standards, accumulated knowledge, etc.

The other orthogonal aspect concerns links with design and implementation. Two directions are also distinguished. The forward direction concerns traceability from a requirement to design elements and components. This traceability is denoted development traceability for design and respectively for implementation. The backward direction is to trace back from either a designed module or a component to original requirements. Thus it is denoted the reverse traceability from design and respectively from components.

As discussed earlier, providing traceability of requirements to their sources and the outputs of the system development process can be along several dimensions. Different stakeholders contribute to the capture and use of traceability information, often with different perspectives. A user has a different vision from an audit specialist, a system designer or a validation engineer. Some typical questions are often asked :

What are the systems components that are affected by a specific requirement?

Why are the components affected by such requirements?

How are the components affected by such requirement?

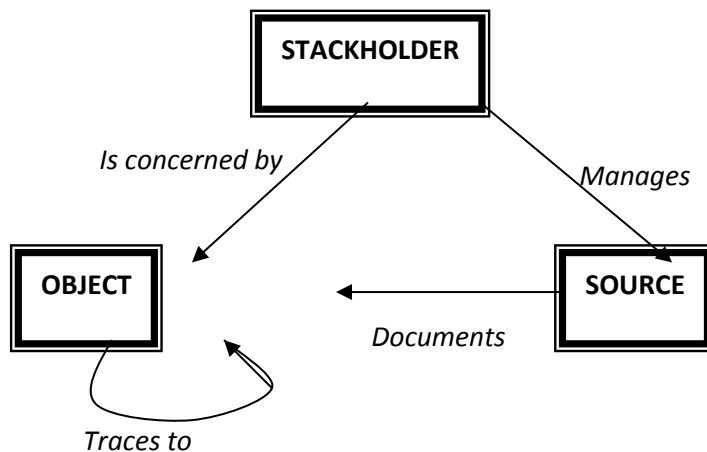
What are the source of a low level requirement?

Why and how two requirements are related?

And so on ...

Some traceability issues like those through abstraction and refinement can be handled by formal methods capture and description, like VDM. However it can be applied only for a small number of requirements. The concept of a link: a link can be made equivalent to class relation. A link between a requirement and a stakeholder is equivalent to class relations. A class is an abstraction of many entities that have common attributes. Classes are considered for various elements. The traceability model is equivalent to an information model that consists in a class diagram, a dynamic diagram.

A meta model of requirement traceability is used. Reference models are available and they can be fitted to some existing tools. The present meta model has been integrated in SLATE . The meta model is described by the following classes diagram shown on figure 10.



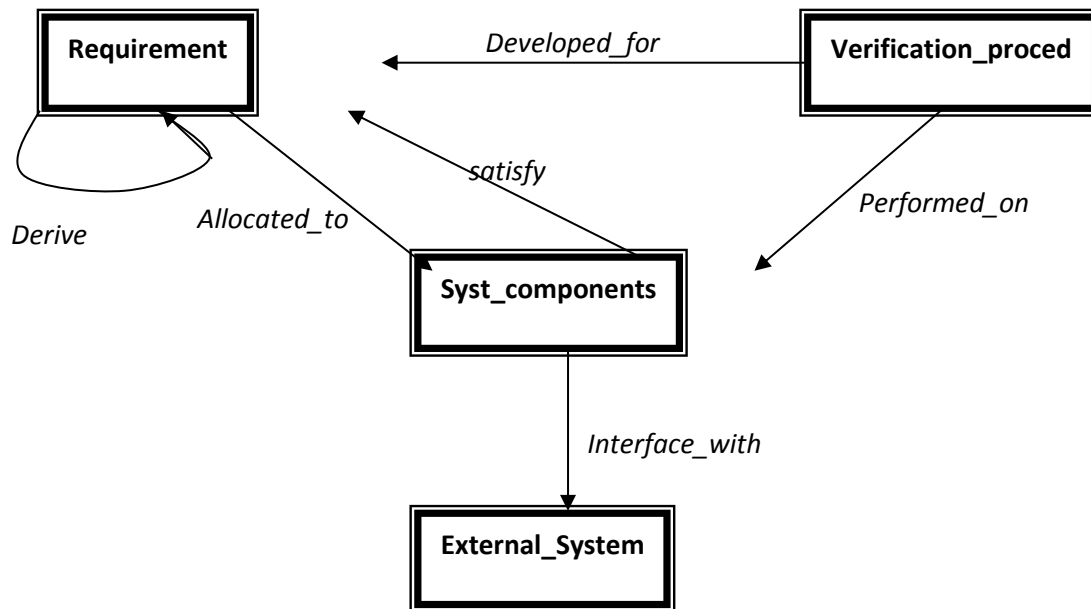
A requirements meta model

An object can belong to one of the following classes: requirement, design, components, system/subsystem, etc . Attributes and operations (activities) are associated with each class, subclass. Links are either between a design and a requirement, a component and a requirement, two requirements, and are represented by the relation *traces to*. Such relation is an abstraction of many links. Consider an audit activity to check for requirement satisfaction with a specific design (reverse design traceability).

Sources are all available information as documents, phone call, e-mail about the object lifecycle. Traceability concerning specific decision made can be found through the relation *documents*.

Stakeholder represents all actors involved in producing the source related to an object; Requirements R1 has been captured by user_1 and being document in requirement file Doc_R1.

All three meta-classes can be used to create specialized classes in order to adapt the meta model to any needs for a traceability model for any requirement process as the following basic traceability model which shows the traceability link through refinement/abstraction



traceability at low level

An important use of requirements traceability is to ensure that the system meets the current set of user requirements. Representing this information is considered to be critical from a project management perspective. It is usually accomplished by creating a link between the set of requirements and a set of system components that *SATISFY* them. Such links may also be indirectly derived. An important concern of the study participants was the lack of support in many CASE tools for the automated identification of derived links ("I don't have the time to link every requirement with everything produced at different stages. These links must be automatically derived whenever possible"). For example, requirements may be linked to designs that are intended to satisfy them. Designs, in turn, may be linked to relevant system components. Then, a derived link is created from requirements to system components.

Such a model is used to identify all traceability links related to requirement-requirement, requirement-implementation (component). A link can be added on system_component to develop decomposition relation at the system, subsystem and component level.

High-level traceability can be modelled by integrating other classes as organisation, system mission, standards. Change proposal can be a specialised class.

4. Traceability in requirement elicitation process

The model can be deployed with respect all subprocesses mentioned in part 2.

Therefore, what is needed to improve our understanding of requirements elicitation is a more detailed investigating into the common and underlying activities of typical requirements elicitation processes. To this end and to present our own overview of the requirements elicitation process, as once again there is very little uniformity in the research literature and practice concerning the names given to the activities often performed during requirements elicitation. Subsequently, we have divided the various individual requirements elicitation tasks into five fundamental and interrelated activities as listed below and described in the following subsections. The five requirements elicitation activities described are:

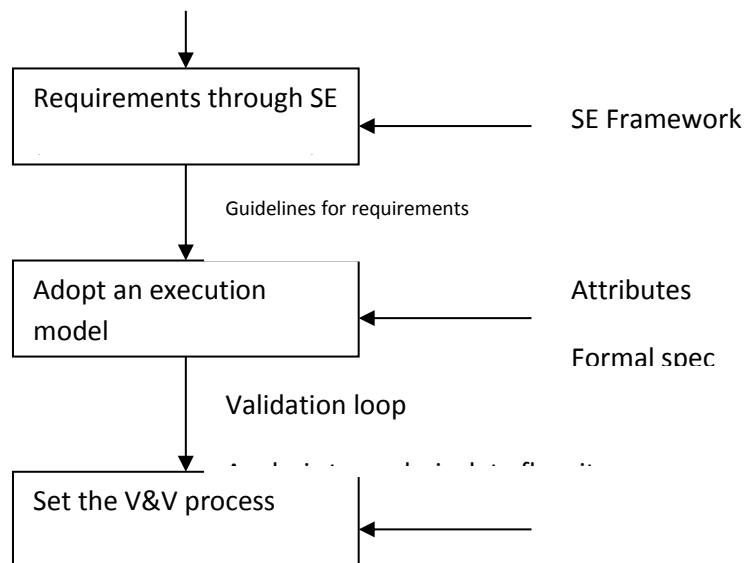
1. Understanding the Domains
2. Identifying the Sources
3. Selecting the Methods
4. Eliciting the Requirements
5. Organizing the Information

Traceability customized model

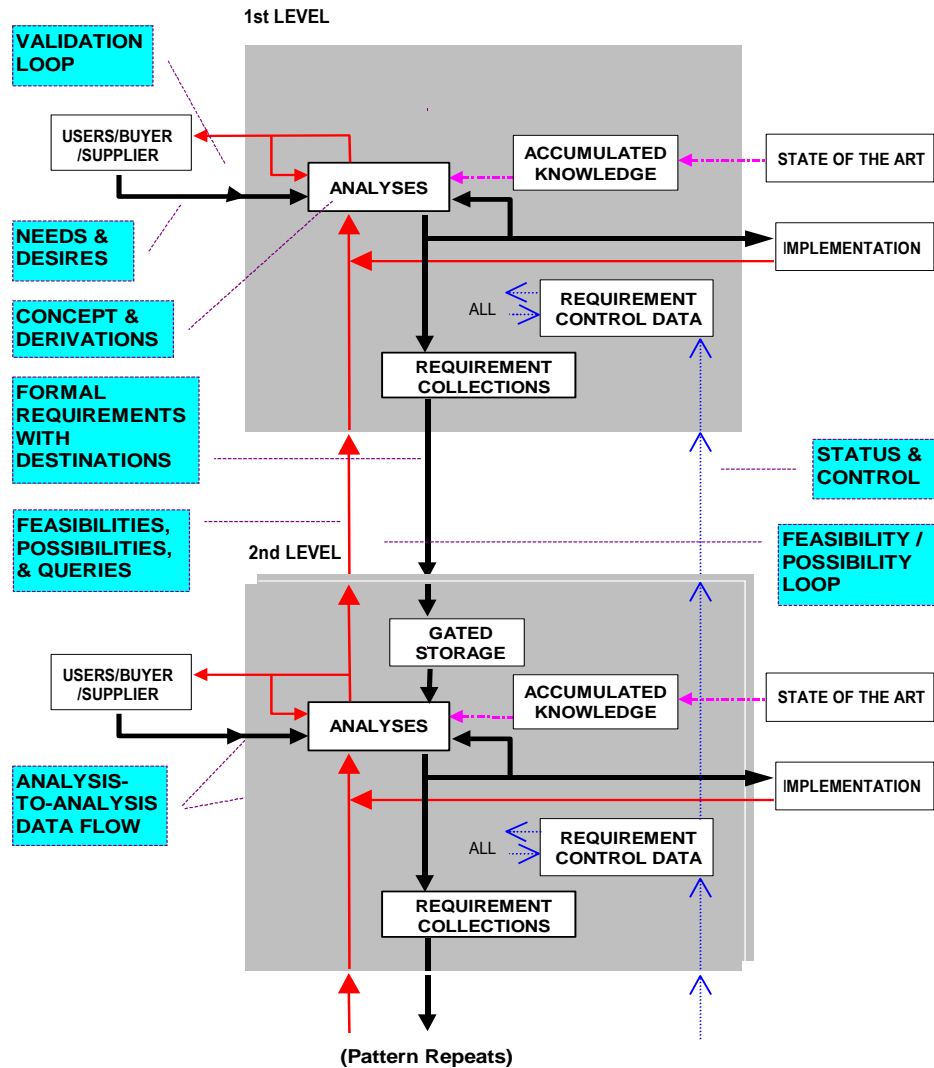
domaine -> sources(stakeholders -> methods -> Elicitation step requirement fixing -> information format

5. Case study Enabling Model for validation and verification with respect to systems engineering standard

We give in the sequel a case study in requirement elicitation of temporal properties



The temporal properties can be set at different levels of abstraction. We propose to map such properties into a general approach developed for requirement elicitation for avionics systems [7]. The approach is best illustrated by the following figure



Such structuring enabled to refine temporal properties, to propagate these in both directions depending on the source level of the initial requirement. The main problems arose when dealing with depending temporal properties [8].

Conclusions

The presented approach takes advantage of the benefits gained from using facilitated collaborative workshops whereby all the relevant stakeholders can cooperatively contribute to the results, and the combination of complementary techniques used to support the main activities, as well as within the actual requirements elicitation workshop environment; traceability remaining as a tool for managing relations between requirements artifacts till possible IEEE SRS model for requirements specification. These strengths are further enhanced by the integration of the entire process into a prescribed set of detailed guidelines based on the underlying knowledge meta-model of information types, thereby ensuring that the process is systematically performed in order to produce a high quality requirements document. We are of the opinion that the resultant approach can produce a requirements

elicitation process that is profitable in terms of offering value for effort, therefore encouraging its acceptance and adoption into industry by organizations and analysts.

References

- [1] Goguen, J. A., Linde, C. (1993): Techniques for Requirements Elicitation, International Symposium on Requirements Engineering, pp. 152-164, January 4-6, San Diego, CA.
- [2] Gottesdiener, E. (2002): Requirements by Collaboration: Workshops for Defining Needs, Addison-Wesley: Boston, MA.
- [3] Hickey, A. M., Davis, A. M. (2002): The Role of Requirements Elicitation Techniques in Achieving Software Quality, International Workshop of Requirements Engineering: Foundation for Software Quality, September 9-10, Essen, Germany.
- [4] IEEE (1998): Std 830 – Recommended Practice for Software Requirements Specifications.
- [5] IEEE (1998): Std 1362 – Concept of Operations Document.
- [6] Maiden, N. A. M., Rugg, G. (1996): ACRE: selecting methods for requirements acquisition, Software Engineering Journal, 11(3), pp. 183-192.
- [7] Sahraoui, AEK, Jones D. A framework for requirements management. International conference systems engineering, Las Vegas, Oct 1999
- [8] Coulin C. , Zowghi, D Sahraoui, AEK: A situational method engineering approach to requirements elicitation workshops in the software development process. Software Process: Improvement and Practice 11(5): 451-464 (2006)
- [7] Sahraoui, AEK, Jones D. A framework for requirements management. International conference systems engineering, Las Vegas, Oct 1999
- [8] Sahraoui, AEK "Requirements Traceability Issues: Generic Model, Methodology And Formal Basis", ; International Journal of Information Technology and Decision Making, 2005, pp.59-80.