A method for identifying unobvious requirements in globally distributed software projects

Smita Ghaisas

Tata Research, Design and Development Center
Tata Consultancy Services
54 B Hadapsar Industrial Estate
Pune 411013, India
smita.ghaisas@tcs.com

Abstract: We present a method and an assisting toolset to identify unobvious requirements in globally distributed projects. We demonstrate use of social software principles and technologies in distributed and collaborative requirements elicitation with a purpose to detect unobvious requirements. In our experience, stakeholders are rarely clear on their requirements. During requirements elicitation workshops (that involve face-to-face communications), they are not able to visualize all the pertinent scenarios. They can typically articulate 'first-order' scenarios resulting out of direct interactions (among stakeholders and between stakeholders and proposed systems), but they find 'second-order' scenarios that result out of multiple stimuli hard to detect a-priori. We refer to such 'hard to detect' requirements as the 'unobvious' ones. In globally distributed situations, this challenge becomes even more pronounced due to lack of frequent and informal communications that allow iterative refinements of stakeholder inputs. We describe an approach that addresses this problem. Our approach includes identification of representative roles and construction of method chunks that include a step-by-step guidance for roles to practice the method chunks relevant to them. Towards this purpose, we have developed a web 2.0 based toolset that presents the method chunks to appropriate roles, synchronizes activities performed by the roles by providing relevant notifications and automates some of the activities recommended in the method. Web2.0 has been chosen as a platform to deliver our method in the light of architecture of participation that it offers. The identification of unobvious requirements has been demonstrated through the results of a case study in Insurance domain.

1 Introduction

Requirements elicitation is a difficult activity even when executed in a co-located mode. Organizations providing IT services invariably need to involve globally dispersed teams. Such teams attempting to elicit requirements lack opportunities for informal and frequent face-to-face communication. This results in unobvious requirements staying assumed in the minds of customers and unstated for developers. It can further lead to dissatisfaction

of the customer and loss of goodwill. Missing requirements' has been known to be one of the top five causes for poor estimation [D1]. In spite of an extensive research [e.g. SFG2-MR7] in this field, recent studies indicate [SFG2] that requirements continue to be a problem. Industry veterans attribute more than half of rework to missing or misunderstood requirements and stress the importance of making use of the high energy (of involved stakeholders) in the elicitation phase [M8,M9]. The reason for missing requirements is lack of "bridging the dialects" or in other words, communication gaps as a result of different interests (hence different foci), competence, and expertise of multiple stakeholders [M8]. In the global context, lack of common understanding of requirements and reduced opportunity to share work artifacts are found be significant factors that challenge the effective collaboration of remote stakeholders [DZ10]. In a recent review, Herbsleb highlights requirements elicitation and communication as one of the four critical research challenges in the context of global software engineering [H11].

We have observed oftentimes that stakeholders have little clarity on their requirements. Even in situations when they are clear about their requirements, they are not in a position to anticipate all the pertinent scenarios. Typically, they are able to visualize the 'first order' scenarios (such as "Reserve book for claimant") that are results of direct interactions among stakeholders, but 'second order 'scenarios (such as "upon cancellation of a claim, make book either available to library if there are no claimants or reserve for one of the other claimants based on certain priority criterion if there are more than one claimant") that result out of multiple stimuli are difficult to detect a-priori. The lack of frequent and informal communication typical of globally distributed projects further complicates this problem and calls for suitable methods and collaborative tools that specifically address the needs of geographically separated multiple stakeholders.

In this paper, we present our work to address this need. Specifically, we have

1. Identified representative roles participating in requirements elicitation in globally distributed projects and constructed method chunks that include a step-by-step guidance to help detect unobvious requirements

2. Developed a web2.0 based toolset that assists the approach by presenting the method chunks to appropriate roles, synchronizing their activities by providing relevant notifications and automating some of the activities recommended in the method

We have used guidelines from design science research [HMPR12-MS14] in our work

2 Related Work

Requirements elicitation in the global context has been studied extensively [e.g.BGM15-CA22]. Bhat et al [BG15] have studied the requirements engineering challenges from the people-process-technology angle and proposed a framework. Other authors [DZ10,D16-DZ18] note that aspects such as a lack of a common understanding of requirements, together with reduced awareness of working local context, trust level and

ability to share work artifacts significantly challenge the effective collaboration of remote stakeholders. Hanisch and Corbitt [HC19] have discussed reasons for impediments to requirements engineering in the global software development and attributed them to differences in shared meaning and contexts. Hsieh [H20] also stresses on culture and shared understanding in distributed requirements engineering. Aranda et al examine technology selection to improve global collaboration. They propose use of cognitive psychology to define strategy to select technology. Herbsleb [H11] identifies environment and tools as another important critical challenge in the context of software engineering. Sinha et al [SSC21] have developed a tool for collaborative requirements management. Globalisation has indeed been noted as a major research challenge in requirements engineering [H11, CA22].

While research in this filed has addressed and analyzed multiple issues, there are no specific approaches designed to identify unobvious requirements especially in the global context. With reference to globally distributed requirements elicitation, we emphasize identification of unobvious requirements as the principal focus of our work. Our approach is different from the ones described above in the following aspects:

1. We identify representative roles involved in globally distributed requirements elicitation and provide each role with a view of the method chunk relevant to him/her. We assist identification of unobvious requirements in multiple ways such as manual validations, automated analysis that uses UML models along with formal semantics and domain specific guidance in the form of seed specifications.

2. We orchestrate identification of unobvious requirements by providing a web 2.0 based solution that assists in effective collaboration among stakeholders.

In our method, we explicitly correlate inputs from stakeholder categories and use these correlations to identify inconsistencies and gaps in requirements specifications. In the process of establishing such correlations, we are able to reveal important details such as missing business rules, and (typically second-order) scenarios that are otherwise undetected during requirements elicitation. We additonally use tool-generated scenarios and counter examples as a medium of communication.

3 Research approach

Kolos-Mazuryk et al [KWE23] examine several existing approaches in design science research [e.g. HMPR12-MS14] and combine the insights offered therein. We broadly follow their route of awareness, suggestion, and circumscriptive development, evaluation and conclusion.

The **problem awareness** stage in our case included (1) inspection of existing methods and tools in the market with a view to understand solutions proposed by other researchers and (2) Interactions with various stakeholders in our organizations with a view to understand their difficulties in terms of identifying unobvious requirements and

(3) actual participation in the process of requirements elicitation and analysis to experience first-hand the problems encountered. The **suggestion stage** consisted of devising a method that is focussed on identification of unobvious requirements through effective stakeholder participation. To achieve this, we identified roles of participants in globally distributed requirements elicitation. We categorized users (of software to be developed) and determined the types of inputs we can receive from them. We also devised a mechanism to compare information elements received from these different sets of users and establish correlations between them. The details of the underlying model of our method are published elsewhere [GSV24-GVKKR26]. The **development stage** included constructing method templates for different roles and developing a toolset to deliver these templates. In the spirit of architecture of participation, we have used web2.0 as the underlying platform for our tool. We used our method and tool in real-life situation involving large-scale global development in the **evaluation stage**. The **conclusion stage** included commercialisation of two modules of our tool and identification of future enhancements.

4 Method details

We identify the following roles relevant to requirements elicitation in globally distributed projects and provide method chunks that would help detect unobvious requirements.

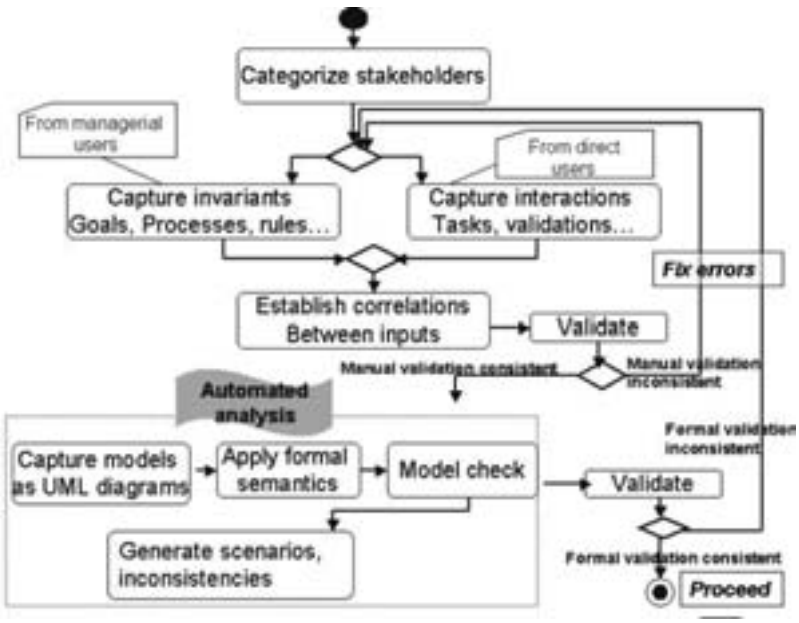

Figure 1 Method chunk for the role ‘ On-site Requirement analyst’

4.1 Method chunk for an onsite requirement analyst

In this role, one needs to interact with customers to elicit requirements. Figure 1 depicts a simplified view of this method chunk designed for the requirements analyst.

4.2 Method chunk for an offshore development lead

This role has little opportunity to interact directly with customers and is yet responsible for delivering the software that meets customer's requirements. The development lead receives specifications and models from the on-site requirements analyst. His team is responsible to develop and deliver the application. Therefore,he needs to verify if the requirement specification is complete, consistent and correct. Prepare a report of inconsistencies and gaps and share it with requirement analyst and customer. Figure 2 shows a simplified view of this method chunk.

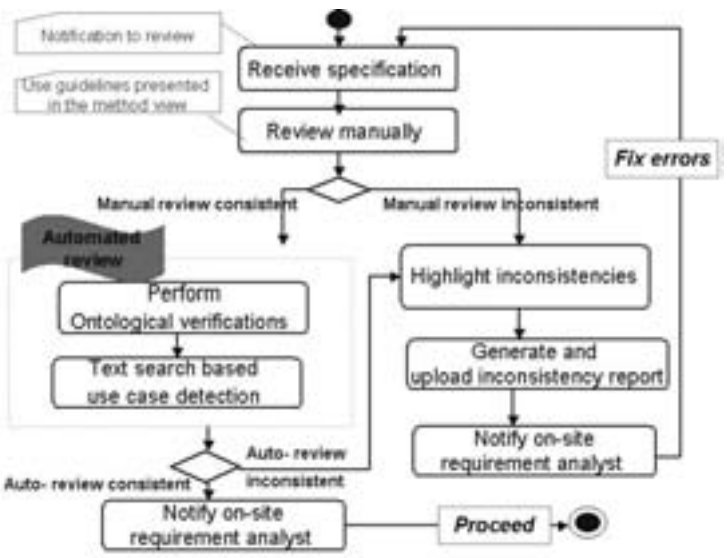

Figure 2: Method chunk for the role 'Offshore development lead'

4.3 Method chunk for the customer

Customer needs to interact with requirement analyst. He answers queries and provides inputs to close gaps raised by the analyst. After the offshore development lead reviews the specification and raises further queries and inconsistencies if any, the customer needs to collaborate with both- the onsite requirement analyst and offshore development lead. He compares the original specification and the review report and answers queries raised by the offshore team and gives inputs for closing the gaps. The customer may at times decline to divulge certain sensitive details that his organization may not permit to be

publicized. In such cases, the report highlights the unanswered queries and record reasons for not providing the details.

4.4 Pilot findings and initial feedback

Pilot experiments indicated that –since we are dealing with geographically distributed teams, we needed a tool that incorporates the social software principles into requirements elicitation activities. The tool should provide relevant guidance on adopting the method. Additionally, the tool should reduce work associated with certain tasks by automating them. For example, it should be possible to generate requirements specification documents, models, and status reports. In Section 5 we present the details of the web 2.0 based tool.

5 The web 2.0 based requirements elicitation tool

Figure 3 shows high-level architecture of our collaborative requirements elicitation tool.

The 'method content' component contains guidelines from the method. The 'Facilitation' component' is used to generate requirement documents and models using industry standard tool.

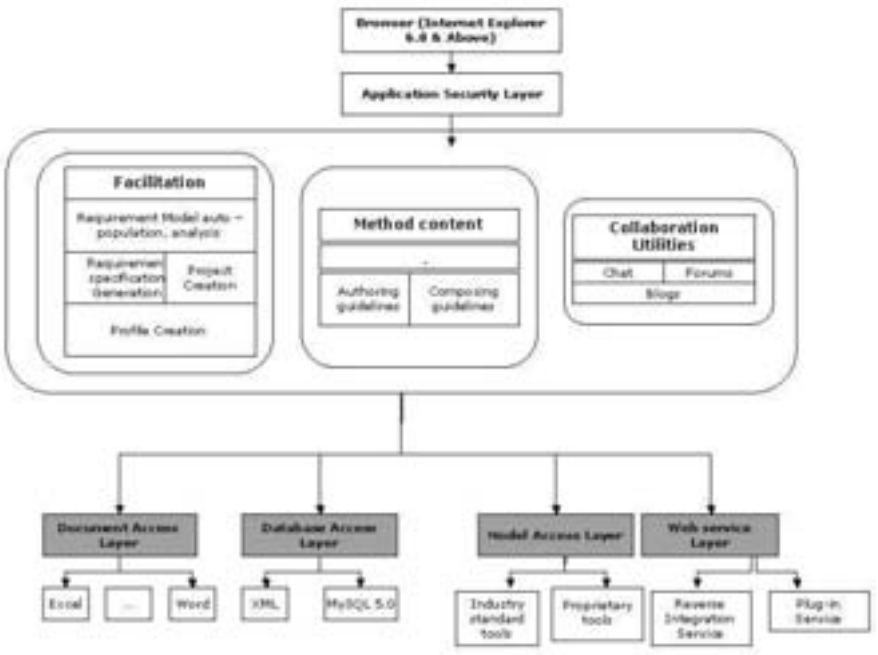

Figure 3: High-level architecture of the collaborative requirement elicitation tool

The ‘Collaboration utilities’ component embodies the social software components such as chat sessions with experts, forums, blog, RSS alerts. The document access layer, database access layer and model access layer facilitate storing of data, generation of requirements specification and requirement models respectively. From the ‘web services layer’, the ‘reverse integration service’ makes it possible to work directly with modeling tools and documents and communicate the latest state of the model to the respective user control libraries.

6 Evaluation of our approach on field

In this section we discuss our experience in applying our method and toolset in a collaborative environment. We present results from a large insurance project.

6.1 Using the method and toolset in a collaborative environment

- The *requirements analyst* selects to work in the ‘insurance domain’. The tool presents templates designed to capture business goals, business processes, and business rules specific to the insurance domain.

- The in-built crosschecks and verifications are presented in the form of alerts that appear after she completes a set of activities. For example, the tool gives her reminders to align stakeholder expectations with business goals when she identifies a *managerial user* and his expectation

- The tool also guides her through identification of use cases by presenting a specific definition of granularity of use cases and their detailing using lightweight templates defined in our method.

- She ensures that the terms used in business entity models; glossary and use case specifications are consistent. She is able to identify gaps in the originally captured requirements using manual and tool-assisted analysis. The tool generates scenarios and inconsistencies. She shares these results with the *development lead* and *customer* in the form of queries.

- The development team lead and customer receive notifications about the uploaded artifacts. They can download the specifications and models and review them. The requirement analysts discusses generated scenarios with the *customer* and the *development* lead. The scenarios communicate in the ‘language of examples ‘and thus are helpful in identifying unobvious requirements. If sections in a specification are not satisfactory or unclear with respect to specified verification &validation criteria (such as alignment with business goals, compliance with business rules, scope as defined by focus of current engagement), they can adds comments in the specification. A review report that highlights the kind of discrepancies mentioned above is generated and shared with the *requirement analyst* and the *customer* for cross-verification.

- The *requirement analyst*, *development lead* and *customer* communicate using chat sessions and close gaps in the original specification iteratively. The improved requirement specifications- documents and models are shared with all participants.

- Throughout the exercise all stakeholders can can start discussion threads on important topics that necessitate discussions. This helps in informal communications and a rapport that is otherwise difficult due to the distributed nature of the development.

6.2 Project overview

In a group insurance system, the insurance company 'LifeCare' provides Quotations to Employers through the Employer's Agents, and if accepted by the Employers, a Scheme is set up for the employees of the Employer with all relevant details. After a Scheme is set up, periodic renewals of the Scheme are carried out. The insurance project had Quotations, New Business, Renewals and Alterations modules. The Quotations module was related to providing quotations to Employers. The New Business module was related to setting up of a new Scheme for the Employer. The Renewals module was related to periodic renewals of these Schemes and Alterations was related to *ad hoc* changes to any of these details. New Business which was more complex than Quotations, was taken up for the exercise. In the New Business module, Schemes are created by Users to cater to Employers who wish to cover pension and life of their selected employees – Members of the Scheme. Schemes can be of different types like TypeA, TypeB and TypeC. Agents are people or companies that service Schemes. Agents can be grouped under Intermediaries. A User can be either an Internal User, an Agent or an Intermediary. A Scheme can have a Legal Arrangement and the Legal Arrangement is managed by a Trustee. A Trustee can be the same person/company that is the Employer. A Trustee has a Status, which can be either Active or Inactive. Members are added to the Scheme with details of the pension and/or life assurance and other benefits that are to be provided. The details are sent for automatic underwriting, which sometimes need manual intervention. Once the underwriting is successfully completed, the Scheme goes Live and is said to be operational

6.3 Results of manual verification

While capturing and mapping process steps to tasks, several instances of inconsistencies were detected between the process steps specified by managerial users and tasks performed by direct users. We present an example next.

The following task captured from clerks at LifeCare did not correspond to any of the process steps narrated by managerial users. The managers were interviewed to fill in the gaps and corresponding process details were captured.

Task: For a scheme 'type A', employees of grade 'Director' are identified for 'manual intervention'.

Corresponding missing process step detail: Employees of grade 'Director' (assumed to earn more than certain specified income limit p.a.) are not eligible for scheme 'type A'. However, some of the employers –typically small sized companies desirous of covering their staff for pension and life insurance had employees of the grade 'Director' on board who earned less than the specified limit and should therefore have been eligible for the scheme.

This special scenario was documented in the requirements specification and later semi-automated by way of sending a notification to LifeCare managers after identifying employees of grade 'Director' so that they could make a decision about the eligibility immediately.

The process 'Cancel policy' as detailed by managers seemed to indicate a money refund within 10 working days after cancellation. However while mapping tasks (outlined by direct users) to the process steps, it was discovered although the management assumed that this Service level Agreement (SLA) compliance was being monitored, it was actually not executed as a part of process. As a result, 50% of the cancelled policies had pending 'money refund cases' associated with them. Automated notifications and reminders after 5 days to all the clerks (and their supervisors) handling cancellation policies were included in the new to-be system.

While capturing rules/policies for process "Create Employee" from managers and validations performed by direct users, we detected some inconsistencies. The following correspondence was established between some of the business rules and validations (Table 1). The missing rule/ policy or validation was identified as a result of this explicit cross verification.

Business rule/policy (captured from managers)	**Validations captured from direct users (e.g. clerks)**
Professionals in category 'Sports' alone can avail of benefits at and below 40 yrs of age.	**Missing validation:** Verify profession of entrants wishing to retire at age <40
Missing policy: Professionals in 'high risk' category need to be scrutinized for financial stability	Verify profession of entrants wishing contribute > specified limit

Table 1: Manual verification across business rules/ policies and validations

6.4 Results of tool-assisted automated analysis

The details of working principles of the tool and this field study are published elsewhere [M27]. Four processes " Create scheme", "Create Trustee", "Create Employee" and "Create Legal Arrangement " were analyzed using the tool. The analysis identified 150 gaps/ unobvious requirements and 70 business rules. Interestingly, answers to the queries

raised as a result of our analysis could not be gathered from the documentation. The team did not know answers to more than half of the queries. Of the requirements elicited in the exercise, 30% were detected by our approach and would have otherwise been unobvious and undetected requirements resulting in 'requirements errors' that are uncovered in later phases. The web2.0 based tool made it easy to interact with the involved stakeholder in an informal way frequently to raise queries, close gaps and inconsistencies.

7 Conclusions

In this work, we have addressed the problem of identification of unobvious requirements in globally distributed projects. We have identified key roles involved in the requirements elicitation process and have devised appropriate method chunks relevant to each role. We have tested our suggestions through pilot experiments and used the findings to enhance our method and to build a web2.0 based collaborative requirements elicitation tool. The tool presents to each role, the method chunk that is relevant to it, assists in carrying out the activities, provides notifications about activities ongoing and completed and thus helps to orchestrate the requirements elicitation exercise. We have evaluated our approach in several real-life large global development exercises. As indicated in the case study, we have contributed to identification of unobvious requirements and thus helped in reducing the pain at the time of acceptance testing.

8 Scope of future work

Another important aspect of requirements elicitation in the social software context is semantic assistance to this collaborative exercise by way of addressing (1) synonymy of terms (2) most commonly accepted terminology in a domain (3) additional terms that complement detected terms in a given context and (4) meaning of a term in a given context. We are working towards developing a method and framework that provides context-sensitive knowledge assistance taking into account the prevalent terminologies (folksonomies that evolve thereof) in a domain and mapping them with domain ontologies that employ the semantic web concepts.

9 Acknowledgements

We thank G. Muralikrishna for his efforts in taking our method to field and sharing the results with us.

References

[D1] Davis A.M., Software Requirements: Objects, Functions and States. Prentice Hall, 1995

[SFG2] Sharp H., Finkelstein A., and Galal G.. Stakeholder identification in the requirements engineering process. In Proc.of the 10th Int. Work. on Datab. & Exp. Sys. Appl., 1999, 387–391

[P3] Potts C., Metaphors of intent. In Proc. of the IEEE Int.Req. Eng. Conf. (RE), 2001,31-39

[A4] Aoyama M. Persona-and-scenario based requirements engineering for software embedded in digital consumer products. In Proc. of the IEEE Int. Req. Eng. Conf. (RE), 2005,85–94,

[CE5] Cohene T. and Easterbrook S., Contextual risk analysis for interview designs. In Proc. of the IEEE Int. Req. Eng. Conf (RE), 2005, 95–104

[SFS6] Sutcliffe S., Fickas M. and Sohlberg M., PC-RE a method for personal and context requirements engineering with some experience. Req. Eng. J., 11(3) (2006), 1–17

[MR7] Maiden N.and Robertson S., Integrating creativity into requirements processes: Experiences with an air traffic management system. In Proc. of the IEEE Int. Req. Eng. Conf (RE), 2005, 105–116,

[M8] Morasco J, Software development productivity and project success rates: Are we attacking the right problem? http://www-128.ibm.com/developerworks/rational/library/feb06/marasco/index.html

[M9] Morasco J., The requirements translation challenge --- can you meet it? http://articles.techrepublic.com.com/5100-3513-6128696.html

[DZ10] Damian, D.E.; Zowghi, D**.,** The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization**,** Req.Engg, 2002. Proc. IEEE Joint International Conference on Volume, Issue, 2002, 319 – 328

[H11] Herbsleb, James D., Global Software Engineering: The Future of Socio-technical Coordination, Future of Software Engineering, 2007. FOSE apos;07, Volume ,23-25 May, 2007.188 – 198

[HMPR12] Hevner A.R., March S.T., Park J., Ram S., MIS Quarterly Vol. 28 No. 1, March (2004), 75- 105

[VK13] Vaishnavi V, Kuechler W, Design research in Information Systems , http://www.isworld.org/Researchdesign/derisISworld.htm

[MS14] March, S.T., and G. Smith, "Design and Natural Science Research on Information. Technology," Decision Support Systems, 15, 4, 1995, 251-266,

[BGM15] Bhat, J.M., Gupta, M., and Murthy, S.N., Overcoming Requirements Engineering Challenges: Lessons from Offshore utsourcing. IEEE Software, 23, 5 2006,. 38-44.

[D16] Damian, D., Stakeholders in Global RE: Lessons learned from practice. IEEE Software, (2007), 21-27

[DM17] Damian, D. and Moitra, D., Global Software Development: How Far Have We Come? IEEE Software, 23,5 (2006),17-19.

[DZ18] Damian, D.E. and Zowghi, D., Requirements Engineering challenges in multi-site software development organizations. Req Engg Journal 8,(2003),149-160.

[HC19] Jo Hanisch and Brian Corbitt , Impediments to requirements engineering during. Global software development,Euro.Journ.ofInf.Systems(2007)16**,** 793–805.

[2H0] Hsieh, Y., Culture and Shared Understanding in Distributed Requirements Engineering. in Int Conf on Global Software Engg.2006.Florianopolis, Brazil.

[SSC21] Sinha, V., Sengupta, B., and Chandra, S., Enabling Collaboration in Distributed Requirements Management., IEEE Software, 23, 5 (2006), 52- 61.

[CA22] Cheng, B. and Atlee, J., Research Directions in Requirements Engineering, in Future of Software Engineering, 2007, FOSE apos;07, Volume 23-25 May, 2007.285-303

[KWE23] L. Kolos-Mazuryk, R. Wieringa, P. van Eck. , Development of a Requirements Engineering Method For Pervasive Services. RE '05, Paris France. August 2005.

[GSV24] Ghaisas S, Shrotri U and Venkatesh R. , Requirement-centric method for Application Development. Workshop in Engineering methods to support Information Systems Evolution (EMSISE'03).September2003,Geneva. http://cui.unige.ch/dbresearch/ EMSISE03/Rp03.pdf

[SBV25] Shrotri, U.; Bhaduri, P.; Venkatesh, R., Model checking visual specification of requirements, First International Conference on Software Engineering and Formal Methods, 2003.Proceedings. Volume , Issue , 25-26 Sept. 2003, 202 – 209

[GVKKR26] Smita Ghaisas, R Venkatesh, Alka Kanhere, Amit Kumar, Manish Kumar and Preethu Rose, Identifying unobvious requirements- an exemplar of the Design Science approach, 2008, Proc of the Third International Conference on Design Science Research in Inf. Systems and Tech. (DESRIST 2008), 335-340

[M27] G. Muralikrishna, Requirements Modeling -- Experience from an Insurance Project, Fourth IEEE International Conference on Software Engg and Formal Methods (SEFM 2006), 11-15 September 2006, Pune, India. IEEE Comp. Soc. 2006, 157-166