

Full Length Research Paper

An enhanced requirements elicitation framework based on business process models

Rana Yousef* and Tamara Almarabeh

Computer Information Systems Department, KASIT, the University of Jordan, Jordan.

Received 23 November, 2014; Accepted 27 March, 2015

Requirements elicitation is a central and critical activity in the requirements engineering process. Completeness is among the most difficult challenges facing requirements engineers. Missing requirements is one of the major causes of software failure; they often result from the lack of anticipation of all possible relations between elements of the system-to-be. In this paper, we propose a requirements elicitation framework which starts with an organization's business process models and buildsthe system's CRUD matrix. This matrix provides all possible relationships between entities and functions of the system in order to capture all possible requirements of the system. The generated relationships between entities and functions provide analysts with the required prompts to ask potential users/stakeholders during interviews to ensure encompassing all questions. The new framework is demonstrated using a real case study; the Cancer Care and Registration in Jordan.

Key words: Requirements elicitation, essential business entities, requirements completeness, missing requirements, CRUD matrix, business process models.

INTRODUCTION

Requirements elicitation represents the set of activities involved in discovering what the system requirements are, including: The identification of all stakeholders of the system, analysis of the problem application domain, the system's operating environment and of the customers' organizational and business environment (Damian, 2000). Improper capturing of system requirements is the major factor in the failure of most software projects.

Requirements are most commonly written in natural language or represented in semi-formal modelling representations. However, a written natural language requirement is error prone and vague leading to

inherent imprecision, such as ambiguities, incompleteness and inaccuracy (Kamalrudin et al., 2011). Completeness of requirements is of major significance in software engineering and can have a major impact on testing, formal verification, robustness and safety of software. Missing requirements are reported as one of the major causes of software failure (Alrajeh et al., 2012; Cheng et al., 2007); they often result from the lack of anticipation of all possible relations between elements of the system-to-be. In this paper, we propose a new requirements elicitation framework which helps anticipate all possible relations between entities and functions for the system starting

*Corresponding author. E-mail: rana.yousef@ju.edu.jo

Author(s) agree that this article remain permanently open access under the terms of the [Creative Commons Attribution License 4.0 International License](https://creativecommons.org/licenses/by/4.0/)

with BPMs and building the system's CRUD matrix. This matrix provides all possible relationships between entities and functions in order to capture all possible requirements of the system. So, the ultimate goal is to capture all possible user requirements of a system and reduce the number of missing requirements as much as possible.

The new framework is proposed after providing an overview of the current requirements elicitation techniques and discussing challenges facing requirements engineers during the requirements elicitation phase. The framework is then demonstrated using a real case; the Cancer Care and Registration in Jordan (CCR).

REQUIREMENTS ELICITATION TECHNIQUES

Here, we provide a brief description of the most popular requirements elicitation techniques; these are interviews, questionnaires, observations, brainstorming, focus group, prototyping, document analysis, joint application development and interface analysis (Kotonya et al., 1998; Hickey et al, 2002; Hickey et al, 2003; Parvianen et al, 2007; Wieggers, 2013; Dyba, 2013; Iqbal, 2014).

Interviews

Interviews are the commonly used and most popular method for requirements elicitation (Kotonya et al., 1998; Sampaio et al., 1996). First you identify the range of people who are involved as potential users/stakeholders of the system. This extends the range of viewpoints that will eventually feed into your proposed system. However, the completeness and correctness of the elicited requirements relies to a great extent to the way the interviewer asks the questions. Our proposed framework helps the interviewer in phrasing questions through identifying all possible relationships between business entities and functions, and hence allows him/her to include all possible questions within the system boundaries and reduce the chance of missing any question and consequently reducing the chance of missing any requirement.

Questionnaires

Questionnaires can be used to get responses from a larger group of people than can be handled in interviews. Eliciting requirements through questionnaires involves asking individuals to respond to a fixed set of questions, often by indicating their preferred option from a list of alternative responses to each question. The correctness and completeness of requirements elicited through questionnaires also depend on the way questions are presented.

Observation

Observation is the approach used to collect requirements by observing how people do their work. This method is used to collect requirements when users are too busy to be involved in interviews. Observation helps elicit implicit requirements that interviews can not reveal. However, it is time consuming and may not work if the current process involves intellectual work or work that is not easily observable.

Brainstorming

Brainstorming is a technique used to generate new ideas and find the solution to a specific issue (Pfleeger et al., 1998; Robertson et al., 1998; Wilson, 2006). This is conducted as a conference with 6 to 10 members from different departments and domain experts.

Brainstorming includes choosing a topic or a problem, and then drawing up and discussing different solutions. This method helps the group members to share their experiences and creativity to find the best solution. It's also helpful in eliciting requirements in a relatively short time period, but depends on the member's creativity.

Focus group

Focus group is getting a group of people together to discuss a problem area, preferably with some sort of 'stimulus material' relating to the topic under consideration. The focus group technique can be traditional where members gather in the same physical room or can be an online focus group which allows members to be located remotely while participating.

This technique is effective for learning people's attitudes, experiences and desires; it also gives the ability to ask questions and creates an environment where participants can consider their personal view in relation to other perspectives. However, this method suffers from several problems; if the group is too homogenous the group's responses may not represent the complete set of requirements. It is also difficult to schedule the group for the same date and time. Finally, the collected data (what people say) may not be consistent with how people actually behave.

Prototyping

A Prototype is the representation or visualization of the actual system parts (Andriole, 1994; Kotonya et al., 1998; Robertson et al., 1998). Prototyping aims to uncover and visualize interface requirements before designing or developing the application. But the process of gathering requirements is limited because it is difficult to discover

the users/stakeholders expectations without providing some model. A prototype may lead users to set unrealistic expectations of the delivered system's performance, reliability and usability characteristics.

Document analysis (requirements reuse)

Document analysis is used to gather details of the "As Is" environment such as existing business rules, entities, and attributes that need to be included in a new system or need to be updated for the current system. It is used to elicit requirements of an existing system by reviewing the documentation of such system. Reviewing the documentation helps understand the current situation and begin to formulate the questions to ask stakeholders to gather additional requirements. Documents analysis is the first step to prepare for interviews or other interaction based elicitation techniques. One of the drawbacks of this method is that documentation can be outdated so it is important to confirm that what you are reviewing is the most current. Reviewing existing documentation can also be a time consuming process. Accordingly, our proposed framework, helps reduce the time required to analyse the existing system through a semi-automated approach of analysing the set of business process models available for the as-is system.

Requirements workshop (joint application development, JAD)

A requirements workshop, also referred to as joint application development (JAD), is a structured requirements elicitation method (Maiden et al., 1996). It is similar to brainstorming, except that stakeholders are the ones who are involved in the discussion of the proposed system. Key stakeholders should be carefully selected for a short, intensive period of time (typically one or few days) (Maiden et al., 1996). Involving too many participants can slow down the workshop process thus negatively impacting the schedule. Conversely, collecting input from few participants can lead to overlooking requirements that are important to users, or to specifying requirements that do not represent the needs of majority of the users.

Interface analysis

An interface is a connection between two components. Most systems require one or more interfaces with external parties, systems or devices. Interface analysis is initiated by project managers and analysts to reach agreement with the stakeholders on what interfaces are needed. Interface analysis helps to clarify the boundaries of the system. Scenarios are generally used to clarify

what the stakeholders need in each interaction. Use cases are the basic guidelines for the scenario models. A thorough interface analysis will describe the purpose of each interface involved and elicit high-level details about it, and outline its content. This type of elicitation is essential for software solutions which involve applications interacting with one another and/or users interacting with applications. The disadvantage of this technique is that it does not provide an understanding of the business process since this technique only exposes the inputs, outputs and key data elements related to the interfaces.

CHALLENGES FACING REQUIREMENTS ENGINEERS DURING THE ELICITATION PROCESS

As described previously, the requirements elicitation phase is characterized by a close interaction with customers, system users and others involved with or affected by the system. Accordingly, the gathered requirements are commonly in a form of written natural language so that this form of human-centric representation is understood by both customers and developers. However, a written natural language requirement is commonly error prone and vague (Kamalrudin et al., 2011).

A number of requirements elicitation problems are the problems of scope, the boundary of the system is ill-defined, unnecessary design information may be given, problems of understanding, users have incomplete understanding of their needs, users have poor understanding of computer capabilities and limitations, analysts have poor knowledge of problem domain, user and analyst speak different languages, ease of omitting "obvious" information, and conflicting views of different users (Christel et al., 1993; Sutcliffe et al., 2013).

A number of methods have been developed to address the above mentioned problems and improve the elicitation of requirements. For example, the ORDIT methodology (Blyth et al., 1993) emphasizes the definition of organizational requirements as part of the elicitation process. Accordingly, system designers can reason about organizational goals, policies and structures. The authors also developed a language with which to discuss human requirements of socio-technical systems, and to demonstrate how these are linked to the technical features of the system design.

The AMORE project (Christel et al., 1993) is concerned with ways in which large amounts of multimedia information can be visualized, stored and retrieved. AMORE is a system that provides ways to capture and organize requirements generated in many formats to facilitate navigation and browsing of large quantities of material.

Other methodologies that were developed and/or proposed to improve requirements elicitation include concepts like maps and repertory grids (Shaw et al., 1996)

and ethnographic techniques (Sommerville et al., 1994).

Amore recent method to improve requirements elicitation is available in Konaté et al (2014) and Sakamoto et al. (2014) where the authors have proposed a method to elicit functional requirements based on existing products that are available in repositories. They accordingly used mobile applications available on the mobile app stores to elicit requirements under an interaction perspective. Other recent work included proposing and validating a framework to help requirements engineers select the most adequate elicitation techniques at any time (Carrizo et al., 2014), while in Konaté et al. (2014) the authors adapted the separation of concerns method to focus on collaborative aspects of requirements elicitation. They separated engineering aspects from collaboration aspects in order to study both of these aspects and then integrate them.

Jin et al. (2003) presented an approach to support the elicitation process which combines various techniques for requirements elicitation including model-based concept acquisition, goal-driven structured interview and concept reuse, their goal was to support the automation of interaction with customers and to automate the construction of requirements models.

Our proposed framework for improving requirements elicitation addresses the problems of scope and completeness, were we aim to help analysts to be able to bound the system scope and ensure collecting the system requirements even those which are implicit and could be missed out.

Accordingly, our framework is expected to resolve many of the current requirements elicitation problems, such as identifying the system boundaries and including all relevant questions during interviews and questionnaires.

THE NEW REQUIREMENTS ELICITATION FRAMEWORK

Figure 1 represents our proposed framework process model, to elicit requirements starting from existing business process models.

Identifying the system's units of work (UoW) and functions to build the CRUD matrix

As we mentioned earlier, documents analysis is the first step to prepare for interviews or other interaction based elicitation techniques. The requirements of the new system can be easily elicited after understanding the existing system. In this phase we deploy a straightforward algorithm to extract essential business entities as well as functions for the as-is system from the system's business process model.

An Essential Business Entity (EBE) is part of the subject matter of the organization (Ould, 2005). So, EBEs

are there because of the business the organization is in, and they can be identified using a brainstorming exercise with the key person in an organization to answer questions concerned with what the organization makes and what product lines and/or service lines it has, what things the organization can be differentiated from other organizations in the same industry, what events in the outside world it needs to respond to, what business entities are listed in the organization's data model and what things do the organization's information systems keep information on. Recognizing who the organization's external and internal customers, can also help in identifying EBEs.

EBEs can be filtered by putting the word 'a' or 'the' in front of each suggestion. If it is not familiar, it should be excluded. Designed entities, which are there because of the way the organization chooses to do its business rather than because these entities characterize its business fundamentally, should also be excluded. For example, an "invoice" is not an EBE for a car manufacturer organization because it is not in the business of invoices. However, for the invoice handling department, which is in the business of handling invoices, an "invoice" is an EBE.

In this content we are concerned with EBEs which are considered as units of work (UoW). A Unit of Work is an EBE which has a lifetime handled by members of the organization. For example, an EBE that does not have a lifetime of interest to the organisation, or that is part of another EBE. We will not also consider EBE that are only roles that play part in the processes (Ould, 2005).

In our framework, we adapt the method used to automatically identify EBE from the set of business process models presented in (Yousef, 2014) and extend it to identify functions and units of work and then build the CRUD matrix, Algorithms I and II.

Algorithm 1: Building the CRUD matrix from BPMs

Input: the set of business process models, $BP = \{bp_1, bp_2, \dots, bp_i, \dots, bp_n\}$, $0 \leq i \leq n$

Output: The CRUD matrix

Begin

1. For each business process model bp_i in BP do
2. Find irredundant tasks of bp_i , $T = \{t_1, t_2, \dots, t_k, \dots, t_x\}$, $0 \leq k \leq x$
3. Call *Extract_EBEs_from_BPMs* (bp_i), return $E = \{e_1, e_2, \dots, e_b, \dots, e_y\}$, $0 \leq i \leq y$
4. Classify EBEs that have a lifetime which is handled by, or are the responsibly of, members of the organisation as Units of Work (UoWs)
5. End for
6. Build the CRUD matrix as follows:
7. for each task t_k in T do
8. set the t_k as the row of the CRUD matrix
9. for each unit of work, u_i in UOW do:
10. set u_i as the column of the CRUD matrix

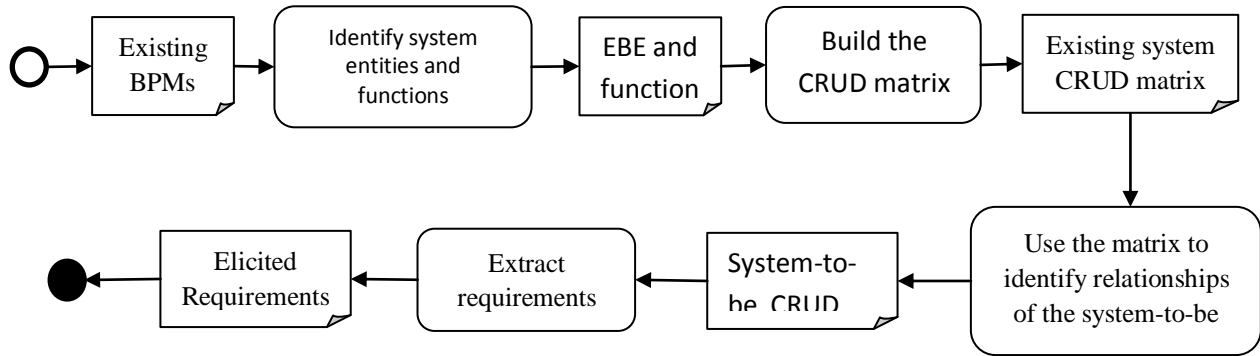


Figure 1. The proposed framework process model.

11. Set the matrix cell as the relationship between t_k and u_l which is one of the CRUD functions (Create, Read, Update or Delete)
 12. end for
 13. end for
End

Algorithm II: Extract_EBEs_from_BPMs

Input: A BPMN business process model, bpm_i , where it is a collection of roles, $R = \{r_0, \dots, r_m\}$ consisting of a number of tasks $T = \{t_0, \dots, t_j\}$

Output: A set of essential business entities present in the bpm_i , $EBE = \{ebe_0, \dots, ebe_{x1}\}$

Begin

Add the bpm_i process name to EBE;

Identify the set of rolls in the business process model, bpm_i ; $R = \{r_0, r_1, \dots, r_j, \dots, r_n\}$, $0 \leq j \leq n$;

For each role r_j in R do the following

Add r_j role name to EBE

Identify the set of tasks in r_j , $T = \{t_0, t_1, \dots, t_k, \dots, t_m\}$, $0 \leq k \leq m$;

For each task t_k in T do the following

If not already in EBE

Add the subject name of the task t_k to

EBE

Add the object name of the task t_k to EBE

End if

End for each task

Exclude designed business entities

End for each role

End

Algorithm I analyses each business process model to extract existing tasks and entities, where each irredundant task is placed in one row and each unit of work, identified as an essential business entity which has a lifetime managed by the organization (Ould, 2005), is placed in a column to build the CRUD matrix's headers. Algorithm II (Yousef, 2014) assumes that the business

processes are modelled using BPMN, however, this algorithm can be generalized for any business process model using model translators such as Yousef et al. (2009).

As we have explained in previously, brainstorming is conducted with members from different departments and domain experts to improve thinking by helping to answer specific questions. So, a brainstorming session is required at this stage to set the matrix's cell relationships for the as-is system.

Building a potential CRUD matrix and eliciting requirements

Having the CRUD matrix built, the system boundaries are set and all potential questions can be extracted from this matrix. According, the analyst can use this matrix to set new relationships for the new system, and generate a new one for the new system. Both matrices will help the analyst phrase questions with a minimum chance of missing a functionality of the system. This would lead to an improvement of the requirements' completeness to a great extent. Following we demonstrate our framework using a real case study form the healthcare domain.

DEMONSTRATING THE PROPOSED FRAMEWORK USING A CASE STUDY

Here, we demonstrate the new framework using a real case study from the healthcare domain; the Cancer Care and Registration in Jordan (CCR).

The Jordan's Cancer Care and Registration (CCR) processes' case study (AbuRub, 2006) is a real case study that has been validated and improved. It provides a number of business process models (BPM)s which we have used to demonstrate the new requirements elicitation framework.

Figure 2 shows one of the CCR processes; the patient reception process. This process was originally obtained

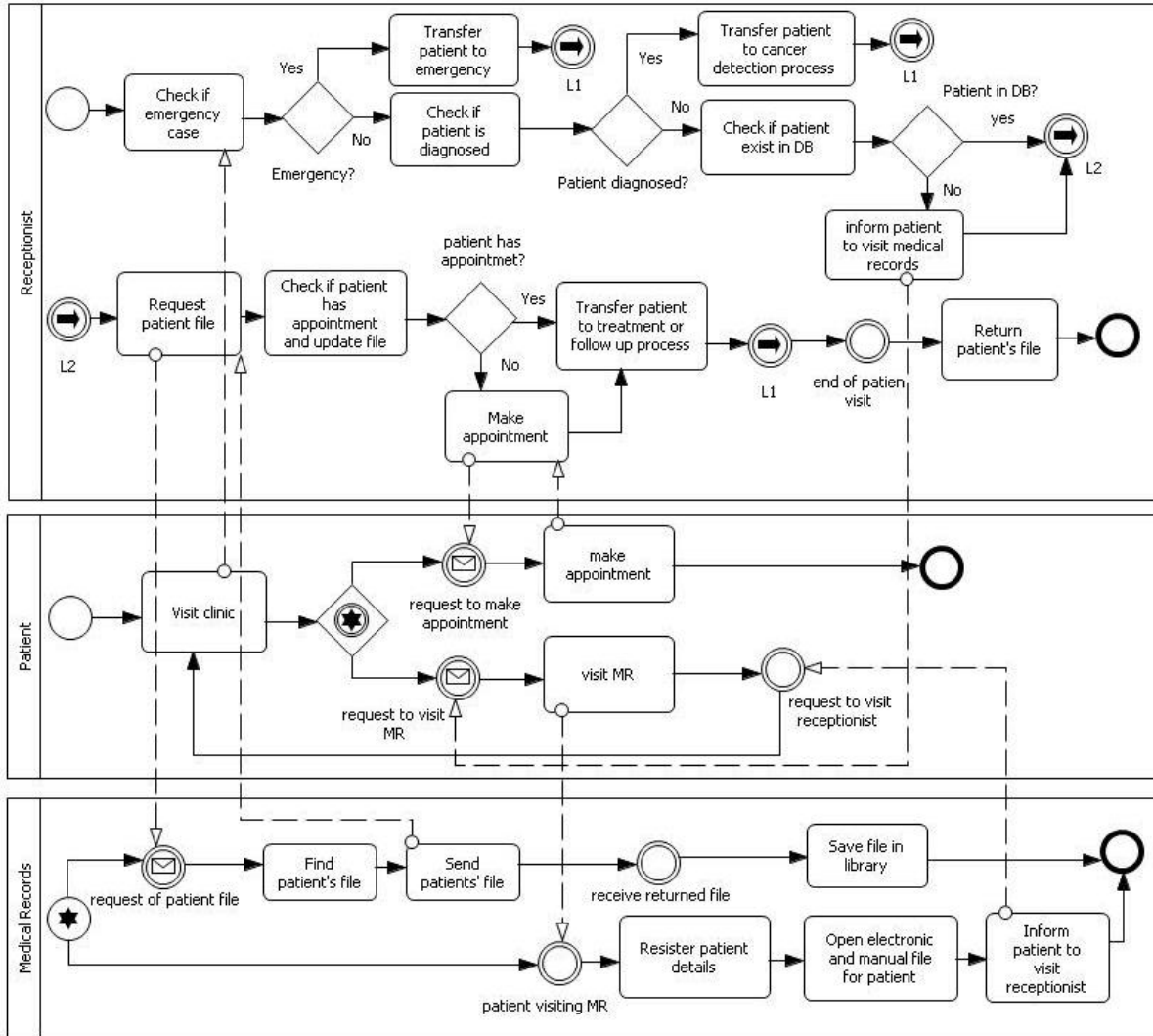


Figure 2. The Patient's Reception Process of the CCR case Study (AbuRub, 2006; Yousef et al., 2009).

from (AbuRub, 2006) and then translated from Role Activity Diagram to Business process model notation diagram by Yousef et al. (2009).

Identifying EBE

A set of EBE were identified using Algorithm II, these are listed in Table 1. Then the set is filtered to produce the Units of Work (those in boldface).

Identifying tasks and building the CRUD matrix

The set of tasks were extracted from the business process model, shown in Table 2, and the CRUD matrix was built using Algorithm I, Figure 3 shows part of the generated CRUD matrix.

The generated CRUD matrix provides an inspiring source of information to the analyst to phrase interview questions and questionnaires, to brainstorm potential relationships between entities and functions, etc. for example, the analyst can identify new functional requirements established from new compensations between entities and functions such as "informing patient" and "appointment", where no relation had been established in this context.

Accordingly, this matrix which was generated from analyzing existing documents, more specifically, the business process models, helps determine the system's boundaries and improves requirements' completeness to a great extent.

CONCLUSIONS

Traditional requirements elicitation techniques include

Table 1. The set of identified EBEs for the CCR case study; the patient's reception process (UoW in boldface).

Reception	Receptionist
Patient	Medical record
Emergency case	Emergency department
Diagnosis	Cancer detection
Database	Electronic File
Appointment	Treatment
Follow-up	Manual file

Table 2. The set of identified functions (tasks).

Check if emergency case	Transfer patient to follow-up
Transfer to emergency	Return patient file
Check if patient diagnosed	Visit clinic
Transfer patient to cancer detection	Visit mr
Check patient Id in DB	Find patient file
Inform patient to visit medical records	Send patient file
Request patient file	Register patient details
Check if patient has appointment	Save file
Update file	Open electronic file
Make appointment	Open manual file
Transfer patient to treatment	Inform patient to visit receptionist

functions	Entities	Medical record	Emergency case	Emergency department	Diagnosis	Electronic File	Appointment	Treatment	Follow-up	Manual
check if emergency case		R	R	R						
transfer to emergency		U	C	C						
check if patient diagnosed		R			R	R				R
transfer patient to cancer detection		U				U	C			U
check id patient in DB		R								
inform patient to visit medical records										
request patient file						R				R
check if patient has appointment		R					R			
update file		U				U				U
make appointment		U				C	U			U
transfer patient to treatment		U				U	C			U

Figure 3. Part of the CRUD matrix for the as-is system.

interviews, questionnaires observation, brainstorming, focus group, prototyping, document analysis, requirements workshop (joint application development, JAD) and interface analysis. Many requirements elicitation problems has been identified in literature, these include the problems of scope, system boundary, understand-ability, poor knowledge of problem domain different terminologies between users and analysts and conflict views of different users. In this paper we were concerned about improving requirements elicitation to help analysts understand the boundaries and identify all entities and functions of the as-is system, so that better questions are phrased during interviews and questionnaires. The proposed framework starts with an inception phase to establish the business case, where in almost all successful applications' developments; this phase constitutes a major part of the development process. Then the framework builds the CRUD matrix of the as-is system, from which requirements of the new system can be inspired. As a consequence, the proposed framework improves requirements' completeness to a great extent. The framework was demonstrated using a real case study from the healthcare domain, the cancer care and registration in Jordan. The next phase in our framework development is to implement a tool to automatically generate questions from the CRUD matrix formulated from the entities identified in the first phase of our framework. So, an analyst can have a well-structured interview with the set of questions that include all entities and functions within the system boundaries.

Conflict of Interest

The authors have not declared any conflict of interest.

REFERENCES

- Alrajeh D, Kramer J, Lamsweerde AV, Russo A, chitel S (2012). Generating obstacle conditions for requirements completeness. In Proceedings of the 2012 International Conference on Software Engineering. IEEE Press. pp. 705-715.
- Andrioles J (1994). Fast cheap requirements prototype: or else. Drexel University, IEEE, 11(2):85-87.
- Blyth AJ, Chudge J, Dobson JE, Strens M (1993). ORDIT: A new methodology to assist in the process of eliciting and modelling organizational requirements. In Proceedings of the conference on Organizational computing systems. ACM. pp. 216-227.
- Carrizo D, Dieste O, Juristo N (2014). Systematizing requirements elicitation technique selection. *Inf. Software Technol.* 56(6):644-669.
- Cheng B, Atlee J (2007). Research directions in requirements engineering. IEEE publications, pp. 285-303.
- Christel MG, Wood DP, Stevens SM (1993). AMORE: The Advanced Multimedia Organizer for Requirements Elicitation (No. CMU/SEI-93-TR-12). Carnegie-mellon univ pittsburgh pa software Engineering inst.
- Damian DEH (2000). Challenges in requirements engineering. Computer Science Technical Report.
- Hickey AM, Davis AM (2002). The Role of Requirements Elicitation Techniques in Achieving Software Quality, Proceedings of the Eighth International Workshop of Requirements Engineering: Foundation for Software Quality, Essen, Germany.
- Hickey AM, Davis AM (2003). Elicitation Technique Selection: How Do Experts Do It?, Proceedings of the Eleventh IEEE International Requirements Engineering Conference, Monterey Bay, CA. pp. 169-178.
- Jin Z, Bell DA, Wilkie FG, Leahy DG (2003). Automated requirements elicitation: Combining a model-driven approach with concept reuse. *Int. J. Software Eng. Knowledge Eng.* 13(1):53-82.
- Kamalrudin M, Hosking J, Grundy J (2011) Improving requirements quality using essential use case interaction patterns. In Proceedings of the 33rd International Conference on Software Engineering. ACM. pp. 531-540.
- Konaté J, Sahraoui AEK, Kolschoten GL (2014). Collaborative Requirements Elicitation: A Process-Centred Approach. *Group Decision and Negotiation*, 23(4):847-877.
- Kotonya G, Somerville I (1998). Requirements engineering process and techniques. John Wiley, 8th Edition.
- Sampaio do Prado Leite J C, Ana Paula PinhoGilvaz (1996). "Requirements elicitation driven by interviews: the use of viewpoints", Proceedings of the 8th International Workshop on Software Specification and Design. IEEE. pp. 85-94.
- Maiden N, Rugg G (1996). ACRE: Selecting Methods for Requirement Acquisition. *IET. Software Eng. J.* 11(3):183-192.
- Ould M (2005). Business Process Management: A rigorous Approach, BCS, UK.
- Parviainen P, Maarit T (2007). A survey of existing requirements engineering technologies and their coverage. *Int. J. Software Eng. Knowledge Eng.* 17(6):827-850.
- Pfleeger SL, Joanne MA (1998). Software engineering: theory and practice. Pearson Education India.
- Robertson S, Robertson J (1998). Mastering the requirements process. Addison Wesley.
- Sakamoto SG, de Miranda LC (2014). M4REMAIP: Method for Requirements Elicitation Based on Mobile Applications under an Interaction Perspective. In Design, User Experience, and Usability. Theories, Methods, and Tools for Designing the User Experience. Springer International Publishing. pp. 74-85.
- Shaw ML, Gaines BR (1996). Requirements acquisition. *Software Eng. J.* 11(3):149-165.
- Sommerville I, Bentley R, Rodden T, Sawyer P (1994). Cooperative Systems Design. *Comput. J.* 37(5):357-366.
- Wilson C (2006). Brainstorming Pitfalls and Best Practices. *ACM.* 13(5):50-63.
- Yousef RM (2014) BPAOntoSOA: Extracting Essential Business Entities from BPMN Models. *Int. J. Comput. Sci.* 11(4):131-134.
- Wieggers K, Joy B (2013). Software requirements. Pearson Education.
- Dyba T, Daniela SC (2013). Process research in requirements elicitation. Empirical Requirements Engineering (EmpiRE), IEEE Third International Workshop on. IEEE.
- Iqbal T (2014). Requirement Elicitation Technique:-A Review Paper. *Int. J. Comput. Math. Sci.* 3:9.
- Sutcliffe A, Pete S (2013). Requirements elicitation: Towards the unknown unknowns. 21st IEEE International Conference for Requirements Engineering (RE).
- Yousef R, Odeh M, Coward D, Sharieh A (2009) Translating RAD Business Process Models into BPMN. In: Applications of Digital Information and Web Technologies. ICADIWT '09, pp. 75-83.
- AbuRub F (2006). Process Modelling for Business Process Improvement, with particular reference to healthcare processes, PhD thesis, University of the West of England.