UNIVERSIDAD DE ANTIOQUIA

DREQUS: an approach for the Discovery of REQuirements Using Scenarios

Aldrin Fredy Jaramillo Franco

Universidad de Antioquia
Facultad de Ingeniería, Doctorado en Ingeniería Electrónica y de Computación
Medellín, Colombia
2020

DREQUS: an approach for the Discovery of REQuirements Using Scenarios

Aldrin Fredy Jaramillo Franco

Tesis presentada como requisito parcial para optar al título de:
Doctor en Ingeniería Electrónica y de Computación

Asesor:
Germán Arturo Urrego Giraldo Ph. D.

Línea de Investigación: Ingeniería de Requisitos

Grupo de Investigación:
Ingeniería y Tecnologías de las Organizaciones y la Sociedad (ITOS)

Universidad de Antioquia
Facultad de Ingeniería, Doctorado en Ingeniería Electrónica y de Computación
Medellín, Colombia
2020

To Franco

ACKNOWLEDGEMENTS

I express my sincere thanks to:

My director Professor Germán Urrego (University of Antioquia), for his wisdom, guidance, and unconditional support. I would also like to thank Professor Colette Rolland (University of Paris I Panthéon, Sorbonne, CRI, Paris, France) for her effective suggestions, valuable feedback, and constant support of this research.

My great friend, Professor Oscar Ortega (University of Antioquia) for the immense trust placed in me and my work. I also want to thank his precise advice which allowed me to improve the research outcomes.

Professor Saïd Assar (Institute of Mines-Télécom Business School, Evry, France Paris, France) for his valuable contributions to our work in the CAREs domain.

Professors Gloria Lucía Giraldo (National University of Colombia), Jaime Echeverri (University of Medellín), and Fabio Vargas (Technologic of Antioquia) for their important reviews performed during the research process.

To the members of the jury: Professors Raúl Mazo (ENSTA Bretagne), Liliana González (University of Medellín), and Bell Manrique (University of Medellín) for dedicating their time to read about my research and to be part of this inquiry.

My dear friends Raúl, and Marielle for allowing me to be part of their family in Paris.

Finally, I want to give very special thanks to my wife Milena, and my mother Elisa for their understanding and unconditional support.

CONTENTS

CHAPTER 1

Introduction

Requirements engineering is recognized as a complex cognitive problem-solving process that takes place in an unstructured and poorly-understood problem context [1]. Requirements elicitation is the activity generally regarded as the most crucial step in the requirements engineering process. The term "elicitation" is preferred to "capture", to avoid the suggestion that requirements are out there to be collected. Information gathered during requirements elicitation often has to be interpreted, analyzed, modeled, and validated before the requirements engineer can feel confident that a complete set of requirements of a system have been obtained [2], [4]. Requirements elicitation comprises the set of activities that enable discovering, understanding, and documenting the goals and motives for building a proposed software system. It also involves identifying the requirements that the resulting system must satisfy in to achieve these goals. The requirements to be elicited may range from modifications to well-understood problems and systems (i.e. software upgrades), to hazy understandings of new problems being automated, to relatively unconstrained requirements that are open to innovation (e.g. mass-market software) [3].

Requirements elicitation remains problematic; missing or mistaken requirements still delay projects and cause cost overruns. No firm definition has matured for requirements elicitation in comparison to other areas of requirements engineering [5]. This research is aimed to improve the results of the requirements elicitation process directly impacting the quality of the software products derived from them.

1.1. Research Method

This research belongs to the field of Design Science Research (DSR). "Design science research creates and evaluates IT artifacts intended to solve identified organizational problems. It involves a rigorous process to design artifacts to solve observed problems, to make research contributions, to evaluate the designs, and to communicate the results to appropriate audiences. Such artifacts may include constructs, models, methods, and instantiations. They might also include social innovations or new properties of technical, social, and/or informational resources" [6]. DSR must produce an "artifact created to address a problem". Its utility, quality, and efficacy must be rigorously evaluated. The research should represent a verifiable contribution and rigor must be applied in both the development of the artifact and its evaluation. The development of the artifact should be a search process that draws from existing theories and knowledge to come up with a solution to a defined problem. Finally, the research must be effectively communicated to appropriate audiences. Peffers et al. propose a DSR methodology consisting of six activities [7]:

Activity 1. Problem identification and motivation.

Define the specific research problem and justify the value of a solution. Since the problem definition will be used to develop an artifact that can effectively provide a solution, it may be useful to atomize the problem conceptually so that the solution can capture its complexity. Justifying the value of a solution accomplishes two things: it motivates the researcher and the audience of the research to pursue the solution and to accept the results and it helps to understand the reasoning associated with the researcher's understanding of the problem. Resources required for this activity include knowledge of the state of the problem and the importance of its solution.

¿How the problem identification and motivation were performed in this research?

To identify the research problems, we followed the next activities:

a. We conducted a Systematic Literature Review (SLR) to understand the progress that has been made in the Requirements Elicitation field during the last years and identify the most promising trends [18]. As a result of this work, we identified the Creative-based Approaches for Requirements Elicitation (CAREs) as one of the most recent and promising trends of proposals for the discovery of Requirements.

b. Considering the relevance of CAREs [18], we made a SLR to identify the methods, benefits, gaps, and drawbacks of these approaches [19]. This work allowed us to identify the problems that are the object of this research.

Both SLRs were performed taking into account the guidelines of Horkoff et al. [9]. We also have considered the works on Evidence-Based Software Engineering and SLRs proposed by Kitchenham et al. [10-11]. The threats to validity were mitigated following considerations of Feldt and Magazinius [12].

Activity 2. Define the objectives for a solution.

Infer the objectives of a solution from the problem definition and knowledge of what is possible and feasible. The objectives can be quantitative, e.g., terms in which a desirable solution would be better than current ones, or qualitative, e.g., a description of how a new artifact is expected to support solutions to problems not hitherto addressed. The objectives should be inferred rationally from the problem specification. Resources required for this include knowledge of the state of problems and current solutions, if any, and their efficacy.

¿How the objectives for a solution were defined?

The objectives for the solution were established considering the problems identified in the current state of the art of CAREs and they are aimed at contributing to mitigate these issues.

Activity 3. Design and development.

Create the artifact. Such artifacts are potentially constructs, models, methods, or instantiations (each defined broadly) [7] or "new properties of technical, social, and/or informational resources" [8]. Conceptually, a design research artifact can be any designed object in which a research contribution is embedded in the design. This activity includes determining the artifact's desired functionality and its architecture and then creating the actual artifact. Resources required moving from objectives to design and development include knowledge of theory that can be brought to bear in a solution.

¿How the solution was designed and developed?

The proposed approach was named Discovery of REQuirements Using Scenarios (DREQUS), and was designed in the following manner: considering previous works, we identified useful resources and components which allowed us to design the required new ones. In this sense, the characterization of existing proposals, obtained in the SLRs [18-19], in terms of their purpose, methods, inputs, outputs, forms of representation of knowledge and resources, gave us an important background to explore the re-use of this knowledge and propose new solutions.

We used Case studies to evaluate the performance of each version of DREQUS; from these assessments, we obtained valuable information that allowed us to make incremental improvements until we reached a satisfactory version of the proposed approach. Figure 1 illustrates the DSR process followed in this research.

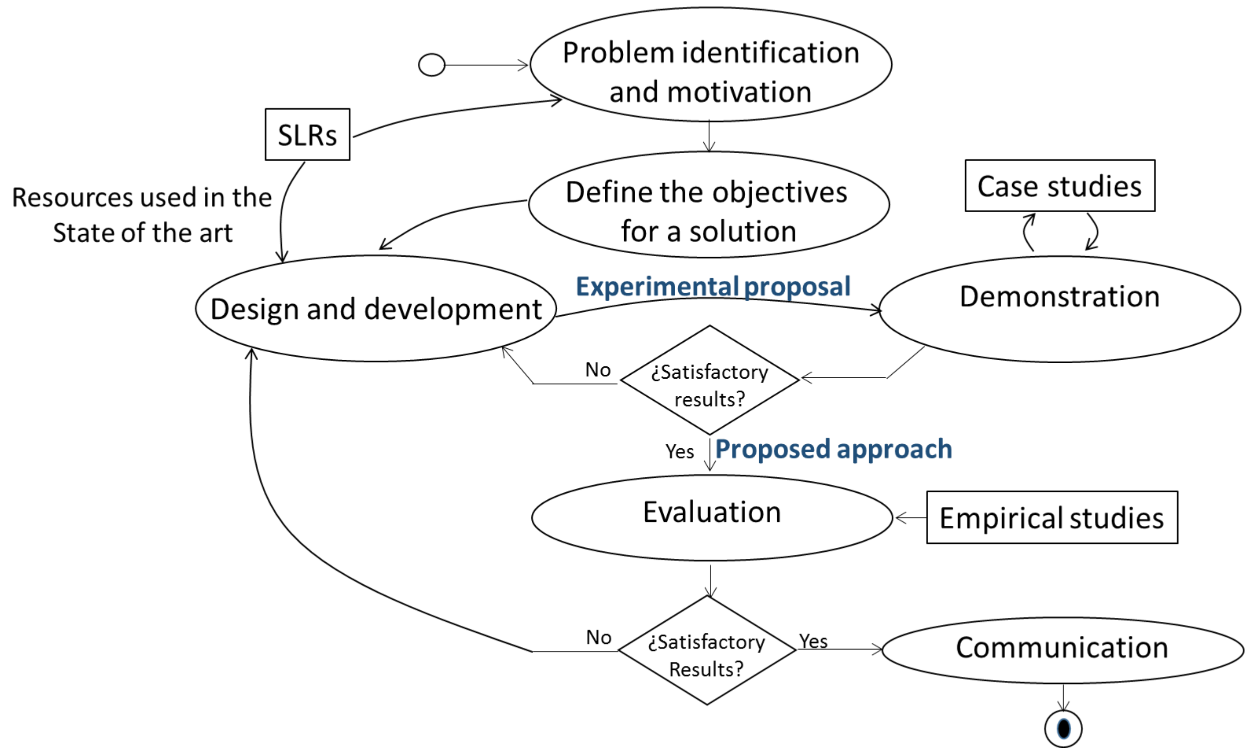


Figure 1. DSR process followed in the research.

Activity 4. Demonstration.

Demonstrate the use of the artifact to solve one or more instances of the problem. This could involve its use in experimentation, simulation, case study, proof, or other appropriate activity. Resources required for the demonstration include effective knowledge of how to use the artifact to solve the problem.

¿How the proposed solution was demonstrated?

DREQUS was demonstrated and preliminary assessed using Case studies. We followed a continuous improvement process where in case of not satisfactory results, we returned to activity 3 repeating the process until having positive outcomes. The case studies were configured and executed according to the guidelines suggested by Lee and Rine [13].

Activity 5. Evaluation.

Observe and measure how well the artifact supports a solution to the problem. This activity involves comparing the objectives of a solution to actual observed results from the use of the artifact in the demonstration. It requires knowledge of relevant metrics and analysis techniques. Depending on the nature of the problem venue and the artifact, the evaluation could take many forms. It could include such items as a comparison of the artifact's functionality with the solution objectives from activity two above, objective quantitative performance measures, such as budgets or items produced, the results of satisfaction surveys, client feedback, or simulations. It could include quantifiable measures of system performance, such as response time or availability. Conceptually, such evaluation could include any appropriate empirical evidence or logical proof. At the end of this activity, the researchers can decide whether to iterate back to step three to try to improve the effectiveness of the artifact or to continue to communicate and leave further improvement to subsequent projects. The nature of the research venue may dictate whether such iteration is feasible or not.

¿How the proposed solution was evaluated?

DREQUS was evaluated employing two Empirical Studies (ES):

ES1: this preliminary assessment was performed to validate the efficacy of DREQUS in comparison to a non-assisted process.

ES2: this final evaluation was conducted to validate the efficacy of DREQUS in comparison to Brainstorming. We selected Brainstorming due that is the most well-known and widely-used creativity fostering technique and is probably the only such technique used in most software developing companies [23-24].

In both ES, the approaches' effectiveness was measured in terms of completeness (recall), precision, and over-specification of the discovered requirements [15]. These metrics are widely accepted by the Requirements Engineering community as relevant quality indicators of a solution [25].

The best practices in Evidence-Based Software Engineering were considered [10]. The threats to the validity of the ES were mitigated according to the principles suggested by Feldt and Magazinius [12].

N.B.: for the sake of the research scope, we did not compare the approaches in terms of the level of innovation (novelty) of the obtained requirements. This task is part of our future works.

Activity 6. Communication.

Communicate the problem and its importance, the artifact, its utility and novelty, the rigor of its design, and its effectiveness to researchers and other relevant audiences, such as practicing professionals, when appropriate. In scholarly research publications, researchers might use the structure of this process to structure the paper, just as the nominal structure of an empirical research process (problem definition, literature review, hypothesis development, data collection, analysis, results, discussion, and conclusion) is a common structure for empirical research papers. Communication requires knowledge of the disciplinary culture.

¿How the proposed solution was communicated?

The research results were published in venues widely accepted by the software engineering community. To ensure the quality of the articles, we followed the guidelines suggested by Alley [16], and Shaw [17].

1.2. Problem Statement

The Requirements Elicitation (RE) phase is one of the most important stages in the development of an information system. One of the main RE challenges is to ensure that the system requirements are consistent with the needs of the organization where it will be used. Consequently, much effort has been devoted to developing approaches and tools to assist the requirements engineers in this critical task of the development process. However, due to the complexity of the process, there are still challenges that remain as priorities for the RE researchers [18].

A recent understanding describes RE process as inherently creative, involving cycles of an incremental building followed by insight-driven re-conceptualization of the problem space. Moreover, in the last decade, a line of Creative-based Approaches for Requirements Elicitation (CAREs) has recognized the importance of creativity in the requirements definition process. These approaches develop a vision in which requirements should be imagined and invented by stakeholders instead of being simply "gathered" from them [19].

CAREs is one of the most remarkable recent trends for requirements discovery with promising advantages. Nevertheless, some drawbacks of these proposals have also been identified; in particular, it is recognized that there is a lack of proposals aimed to guide the systematic exploration of the entire Solution Ideas Space (SIS) avoiding one to wander aimlessly, over-visiting some parts of the space and under-visiting other parts of the space [19-22]. In this sense, this research is motivated by the following issues identified in the CAREs state of the art:

a. There is a lack of mechanisms aimed to ensure the alignment of the discovered requirements with the system purpose.

b. The current proposals rely on the Requirements Engineer´s knowledge, which in many cases is intuitive, and based on the experience using a particular technique(s). Therefore, there is a need to bring support and guidance to the stakeholders in the systematic exploration of the entire Solution Ideas Space (SIS) and the consequent discovery of requirements.

c. The current solutions do not facilitate the transition from the discovered requirements towards the future system specification.

1.3. Research Questions

Considering the identified limitations of CAREs, this thesis is intended to address the following research questions:

Main RQ: ¿How the Requirements Engineer can be effectively assisted in the systematic exploration of the Solution Ideas Space and the consequent discovery of requirements for a future system?

The specific questions to be solved are:

RQ1: ¿How the alignment of the discovered requirements with the system purpose can be ensured?

RQ2.: ¿How effective guidance for the systematic exploration of the Solution Ideas Space and requirements discovery can be provided?

RQ3: ¿How the transition from the discovered requirements towards the future system specification can be facilitated?

1.4. Research Hypotheses

In order to validate the DREQUS efficacy concerning the research questions, we defined the next hypotheses, means of validation, and metrics:

Table 1. Research hypotheses, means of validation, and metrics.

Hypothesis	Means of validation or demonstration	Metric used to validate the hypothesis	Metric definition
H1: DREQUS contributes to the alignment of the discovered requirements with the system purpose.	Empirical studies	Precision	Precision reflects the accuracy of the approach (i.e. how correct are the requirements discovered with the approach) [15].
		Over-specification	Over-specification measures how much extra requirements discovered using the approach

			are not found in the answer key [15].
H2: DREQUS provides effective guidance for the systematic exploration of the Solutions Ideas Space and requirements discovery.	Empirical studies	Recall	Recall reflects the completeness of the results produced by the approach [15].
H3: DREQUS approach facilitates the transition from the discovered requirements towards the future system specification.	It was demonstrated through a case study.	It does not apply.	It does not apply.

1.5. Contributions

The main contributions of DREQUS are:

a. An approach for the discovery of requirements (stage 1 of DREQUS)

This approach allows to tackle the next issues:

- There is a lack of mechanisms aimed to ensure the alignment of the discovered requirements with the system purpose.

- The current proposals rely on the Requirements Engineer´s knowledge, which in many cases is intuitive, and based on the experience using a particular technique(s). Therefore, there is a need to bring support and guidance to the stakeholders in the systematic exploration of the entire Solution Ideas Space (SIS) and the consequent discovery of requirements.

This solution uses a set of proposed *Probing questions* to facilitate the discovery of requirements. These questions are dynamically instantiated from *The System promise, Internal actions of a Future System Scenario*, and a set of proposed *Functional and Cognitive verbs*. The proposed *Probing questions* and their elements can also be used by other approaches oriented to requirements elicitation.

b. An approach for the requirements specification (stage 2 of DREQUS):

This process contributes to solving the following CAREs drawback:

- The current solutions do not facilitate the transition from the discovered requirements towards the future system specification.

This component of DREQUS facilitates the requirements specification through of Use cases. This task is performed using algorithms for requirements similarity calculation and clustering, and a proposed set of rules, and guidelines. The proposed approach can also be used by other requirements elicitation techniques to specify their requirements.

c. A proposed set of Functional and Cognitive verbs

Functional verbs denote processes that are currently performed by software in the Business Information Systems (BIS) domain. Meanwhile, *Cognitive verbs* denote mental processes performed by humans. As evidenced by recent advances in Artificial Intelligence (AI), this type of processes will increasingly be present in future systems. These verbs were identified from previous works, and are used as the core of probing questions which lead the discovery of requirements of an under-construction system.

The proposed set of Functional and Cognitive verbs is useful not only for other Requirements Elicitation approaches but also for Business activities related to Knowledge Management.

d. A set of rules and guidelines to discover Use cases from other existing ones

Given a set of Use cases, DREQUS facilitates the discovery of new Use cases utilizing rules and guidelines aimed to find *Alternative*, *Complementary*, and *Exceptional* scenarios. These heuristics are also useful for other Requirements Elicitation approaches based on Use case models.

e. A repository of information about requirements elicitation approaches

In order to understand the progress that has been made in the Requirements Elicitation field during the last 30 years, we performed a Systematic Literature Review. As a result of this work, we have built a repository with 505 proposals. For each of them, this resource stores valuable information about aspects like *Number of citations, Source*, *Target, Purpose, Represented knowledge, Form of representation of knowledge*, *Methods*, *Resources*, *and Tools* among others. This information is indeed a useful source of knowledge for researchers of the Requirements Elicitation community. It can be freely accessed from [18].

1.6. Thesis organization

This thesis is organized as follows:

Chapter 2 presents the state of the art. In the first place, a SLR in the Requirements Elicitation domain is presented; then is introduced a more specific SLR in the promising field of Creative-based Approaches for Requirements Elicitation (CAREs) is introduced.

Chapter 3 illustrates the proposed solution. This section describes the mechanisms, and methods proposed by DREQUS to tackle the aforementioned issues of CAREs. The proposed solution is demonstrated through a case study.

Chapter 4 provides two Empirical Studies (ES) conducted to evaluate the proposed approach.

Finally, **CHAPTER 5** presents the research discussion, conclusions and future research.

References

[1] Nguyen L. and Swatman P. A. Promoting and Supporting the Creative and Insight-Driven RE Process Using Design Rationale. Working Paper, 15, UNISA, Adelaide, 2006.

[2] Ben Achour C. and Rolland C. Introducing genericity and modularity of textual scenario interpretation in the context of requirements engineering. Centre de Recherche en Informatique. Université de Paris I – Sorbonne. 1997.

[3] Cheng B. H. C. and Atlee J. M. Research Directions in Requirements Engineering. Future of Software Engineering. FOSE´07. 2007.

[4] Nuseibeh B. and Easterbrook S. "Requirements Engineering: A Roadmap". Proceedings of the Conference the Future of Sofware Engineering. ICSE 00. Limerick Ireland 2000.

[5] Sutcliffe A. and Sawyer P. Requirements elicitation: towards the unknown unknowns. Proceedings of the RE Conference 2013. Rio de Janeiro, Brazil, 2013.

[6] Hevner, A.R., March, S.T., and Park, J. Design Research in Information Systems Research. MIS Quarterly, 28, 1, 75-105. 2004.

[7] Peffers, K., Tuunanen T., Chatterjee M.A, Rothenberger S. A design science research methodology for information systems research. Journal of Management Information Systems 24, n° 3 pp. 45--77. 2007.

[8] Järvinen, P. Action Research is Similar to Design Science. *Quality & Quantity*, 41, 1, 37-54. 2007.

[9] Horkoff J., Li T., F., Salnitri M., Cardoso E., Giorgini P., Mylopoulos J.and Pimentel J. Taking Goal Models Downstream: A Systematic Roadmap. Research Challenges in Information Science (RCIS), IEEE Eighth International Conference on. Page(s): 1 – 12. 2014.

[10] Kitchenham B., Dyba T., and Jorgensen M. "Evidence-based software engineering," in Proc. of the 26th International Conference on Software Engineering (ICSE04), pp. 273–281. 2004.

[11] Kitchenham, R. Pretorius, D. Budgen, O. P. Brereton, M. Turner, M. Niazi, and S. Linkman, "Systematic literature reviews in software engineering – a tertiary study," Information and Software Technology, vol. 52, no. 8, pp. 792 – 805, 2010.

[12] Feldt R. and Magazinius A.. Validity Threats in Empirical Software Engineering Research-An Initial Survey. In Proceedings of the 22nd International Conference on Software Engineering and Knowledge Engineering, pp. 374-379. 2010.

[13] Lee S. W. and Rine D. C. Case Study Methodology Designed Research in Software Engineering Methodology Validation. In Proceedings of the Sixteenth International Conference on Software Engineering and Knowledge Engineering (SEKE'04), pp. 117-122, Banff Centre, Banff, Alberta, Canada, June 20 to June 24, 2004.

[14] Sakhnini V., Mich L., Berry D. M. Group versus individual use of power-only EPMcreate as a creativity enhancement technique for requirements elicitation. Springer Science+Business Media New York. 2017.

[15] Harmain, H., Gaizauskas, R.: CM-Builder. An automated NL-Based CASE Tool. Proceedings of the Fifteenth IEEE International Conference on Automated Software Engineering (ASE´00). Grenoble. 2000.

[16] Alley M. The craft of scientific writing. Springer-Verlag New York. ISBN 978-1-4419-8288-9. 2018.

[17] Shaw M. Writing Good Software Engineering Research Papers. In Proceedings of the 25th International Conference on Software Engineering (ICSE03). 2003.

[18] Jaramillo A. Requirements elicitation approaches: a systematic review. Research Challenges in Information Science (RCIS), IEEE 9th International Conference on. Athens. Greece. 2015.

[19] Jaramillo, A., Assar, S. Leveraging creativity techniques in requirements elicitation: a literature review. Requirements Engineering Magazine. International Requirements Engineering Board. Issue 2016-02. 2016.

[20] Berry D., Mich L. and Sakhnini V. The effectiveness of an optimized EPMcreate as a creativity enhancement technique for web site requirements elicitation. Requirements Engineering Journal. 2012.

[21] Sakhnini V., Mich L., Berry D. M. Group versus individual use of power-only EPMcreate as a creativity enhancement technique for requirements elicitation. Springer Science+Business Media New York. 2017.

[22] Herrmann, A., Mich, L., and Berry, D.M. Creativity Techniques for Requirements Elicitation: Comparing Four-Step EPMcreate-Based Processes, Seventh Workshop on Empirical Requirements Engineering (EmpiRE), Banff, AB, Canada, August 2018.

[23] Berry, D., Mich, L., Anesi, C. 2005. Applying a pragmatics-based creativity-fostering technique to requirements elicitation. Requirements Engineering Journal. 2005.

[24] Sakhnini V., Mich L., Berry D. M. Group versus individual use of power-only EPMcreate as a creativity enhancement technique for requirements elicitation. Springer Science+Business Media New York. 2017.

[25] Pastor, Ó., González, A., Condori-Fernández, N., España, S. An empirical comparative evaluation of requirements engineering methods. Brazilian Computer Society. 2010.

CHAPTER 2

State of the art

The first section of this chapter presents a Systematic Literature Review (SLR) conducted to identify and characterize the proposals performed in recent years to tackle the requirements elicitation problem. As one of the main results of this research, we discovered that Creativity-based Approaches for Requirements Elicitation (CAREs) appear as one of the most promising trends.

Considering the importance of CAREs, we performed a second and more specific SLR in this domain; this time we were interested in identifying and characterizing the existing proposals of CAREs to analyze their strengths and weaknesses as a resource to identify in a precise manner the main challenges that these approaches faced in the next years. This last SLR can be found in the second section of this chapter.

2.1. Requirements Elicitation Approaches: A Systematic Review

2.1.1. Introduction

The Requirements Elicitation (RE) phase is one of the most important stages in the development of an information system. One of the main challenges of the RE is to ensure that the system requirements are consistent with the needs of the organization where it will be used [1]. Consequently, much effort has been devoted to developing approaches and tools to assist the requirements engineers in this critical task of the development process [2], [3]. Nevertheless, due to the complexity of the process, there are still challenges that remain as priorities for the RE researchers.

This chapter aims at understanding the progress that has been made in the RE field during the last 30 years. More specifically we are interested in answer the research question: ¿what approaches exist that supports RE in software development processes? We detail this overarching question by investigating relevant characteristics of the identified works. To accomplish this purpose, taking into account the work of Kitchenham et al. [4], [5], we have conducted a Systematic Literature Review (SLR), which results in 505 publications.

This chapter presents contributions for researchers and practitioners who can have a better understanding of the RE evolution during the last years. Besides, the obtained results provide insights into the relevant issues and perspectives that should be considered in future proposals. The resulting repository of RE approaches is available online; we hope that this resource will facilitate the development of future works aimed to contribute to the RE field.

This chapter is organized as follows: Section 2.1.2 introduces the research questions and methodology; Section 2.1.3 presents the obtained results; Section 2.1.4 analyzes the threats to the validity and the actions taken in order to mitigate them; Section 2.1.5 reviews related works and Section 2.1.6 considers the SLR conclusions.

2.1.2. Systematic Literature Review Methodology

In order to answer the overarching question "RQ0: ¿what approaches exist which support requirements elicitation in software development processes?" we have defined a set of inclusion/exclusion criteria, which delimit the scope of the Systematic Literature Review (SLR). These criteria include the search for comprehensive approaches aimed to elicit Functional Requirements (FRs) and Non-functional requirements (NFRs) in the software engineering domain. The search was limited to the last 30 years (since 1989 to 2019); to ensure the quality of reviewed proposals, we selected publications in conferences, journals or books available at recognized scientific databases. A summary of the inclusion/exclusion criteria is presented in Table I.

Aimed at characterizing the proposed approaches, we formulated more detailed questions as showed in Table II.

TABLE I. CRITERIA FOR INCLUSION AND EXCLUSION OF PUBLICATIONS

Inclusion criteria	**Exclusion criteria**
The proposal elicits functional requirements (FRs) or non-functional requirements (NFRs) or both and	The proposal elicits knowledge outside of the software engineering domain, or
Describes comprehensive approaches, and	Describes requirements elicitation techniques e.g. interviews, brainstorming, ethnography, etc., or
Published in 1989 or after, and	Published before 1989, or
In conference, journal, or in/is a book.	In workshops, regional conferences, and theses.

TABLE II. RESEARCH QUESTIONS

RQ0	¿What approaches exist which allow the requirements elicitation in software development processes?
RQ1	¿What sources of requirements are required by the approach?
RQ2	¿What is the purpose of the approach? ¿What problem is solved? ¿What targets are produced by the approach?
RQ3	¿What knowledge is represented in the approach?, ¿How is represented?
RQ4	¿How the requirements are discovered?, ¿What methods, resources and tools are used by the approach?

Our SLR is based on the process proposed by Horkoff et al. [6]; we followed a two-steps process aimed to increase the proposals coverage. The first step made a systematic search in several scientific databases: IEEE, SPRINGER, ACM, DBLP, and SCOPUS; considering our research questions and scope we defined the following search string:

("elicitation" OR "gathering" OR "acquisition" OR "discovery") AND ("requirements" OR "functional requirements" OR "non-functional requirements") AND ("approach" OR "proposal" OR "method")

We used the selected scientific engines and configured the search dates to only obtain proposals since 1989. Using the inclusion/exclusion criteria, we filtered the works by reading the title and venue, the abstract and, if necessary, the introduction or conclusions. Step 2 (snowballing process) took each publication identified in the previous step and scanned their references to select the relevant ones. We assessed the candidate papers in the same way we evaluated the articles in step 1. To ensure that the snowballing stops, we limited the search of references to a depth of two. Fig. 1 summarizes the SLR process.

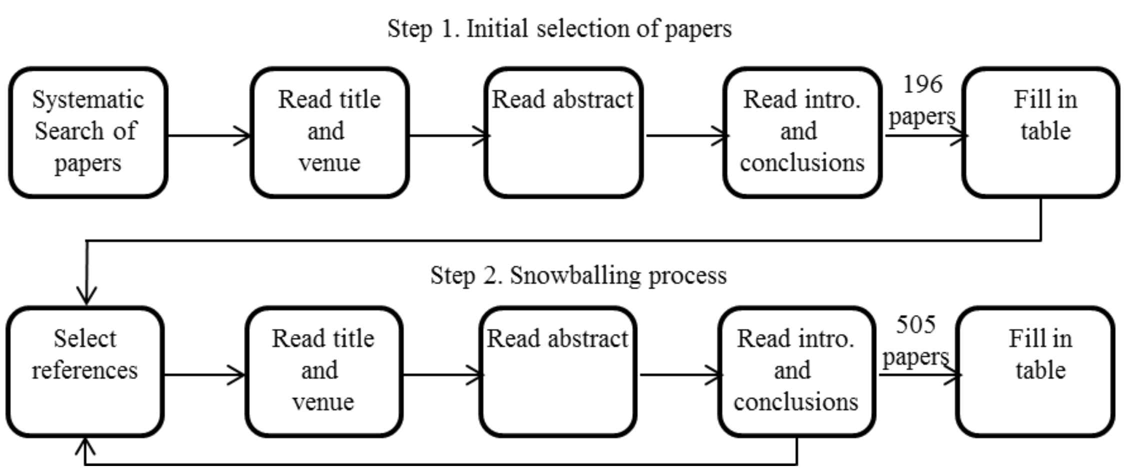


Fig. 1. SLR process

As the outcome of the initial search (step 1), we obtained 196 articles; after submitting these papers to step 2, we finally ended up with 505 approaches. Each proposal was analyzed and tagged with the following fields*: identifier*, *title*, *authors, publication venue, publication year, database, journal/conference/book, # citations, source, target, purpose, knowledge represented, form of representation of knowledge, methods, resources, and tools*. We consolidated the results in a document that is available online.[1] This repository of tagged proposals facilitated the answer to the research questions as is presented in the next section.

2.1.3. Results

As indicated in the previous section, through our SLR, we have answered the overarching question: RQ0. ¿What approaches exist which allow the requirements elicitation in software development processes? The identified publications were found in the following scientific databases: IEEE provided 56% (283 publications), SPRINGER 22.9% (116 p.), ACM 9.1% (46 p.), SCOPUS 5.6% (30 p.) and other databases 5.6% (30 p.). Fig. 2 shows the publications per database.

We noted that most of the publications (56%) were found in the IEEE database being preferred by the RE community to date. Fig. 3 shows that 68.3% (345 p.) corresponds to conferences; followed by journals 26.7% (135 p.); 3.96% (20 p.) are book chapters and 0.9% (5 p.) corresponds to books. These results show how conferences are by far the main mean of knowledge socialization for the RE community.

In order to establish the evolution of the RE field, we have defined two periods of comparison: period 1, from 1989 to 2003, and period 2, from 2004 to 2019. Fig. 4 presents the number of publications in each of these periods. We observe a remarkable increase of proposals (367 p.) during the last period in comparison with the first one (138 p.); this puts in evidence the progressive interest and effort dedicated by researchers and practitioners in the search for solutions to the challenges posed by RE. This also is indicative of the growing relevance of RE issues not only for academia but also for industry.

[1] http://goo.gl/P0XMvy

In the next paragraphs, we are going deeper into the approaches aiming to answer the detailed questions (RQ1-RQ4).

RQ1. Sources of requirements. Table III shows that the main sources in both periods are: Domain knowledge, Initial requirements, and Stakeholders goals. We note that in period 1 these sources are similarly relevant for researchers; nevertheless, this changes in period 2 where Stakeholders´ goals are given priority (64 p.) over Initial requirements (40 p.) and Domain knowledge (37 p.). We can also observe how some important sources in period 1 (ERP and business processes, Situations, arguments, selected strategies and options, and Use cases) have lost relevance in the second period. Likewise, in the last period, we note the emergence of sources like Business process models, Security goals, privacy goals and attacks, and Legal texts.

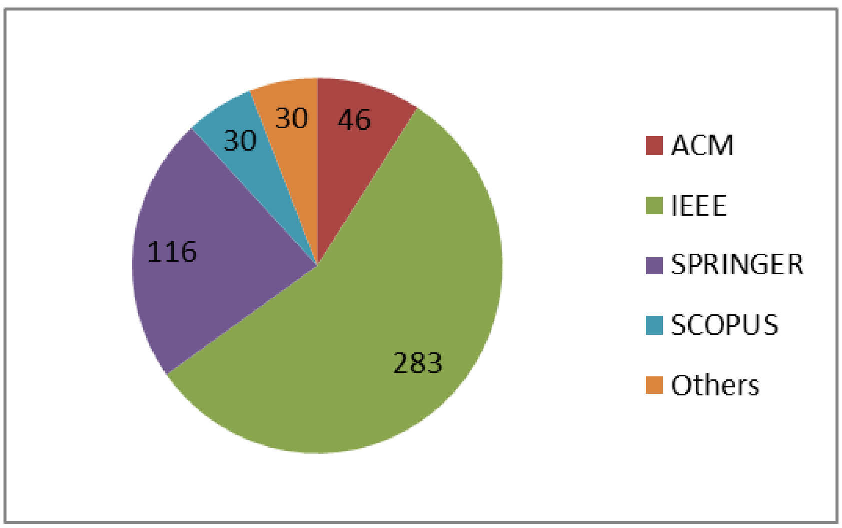


Fig. 2. Publications per database.

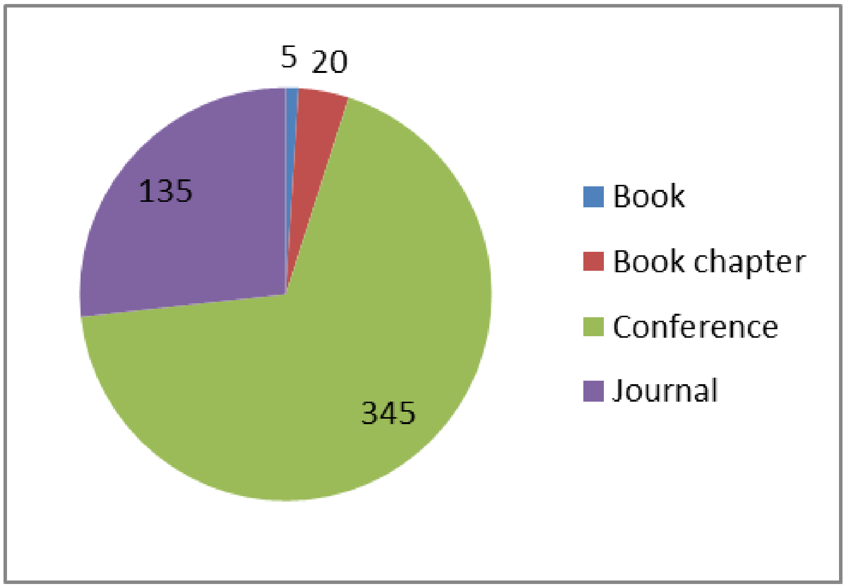


Fig. 3. Types of publication venues.

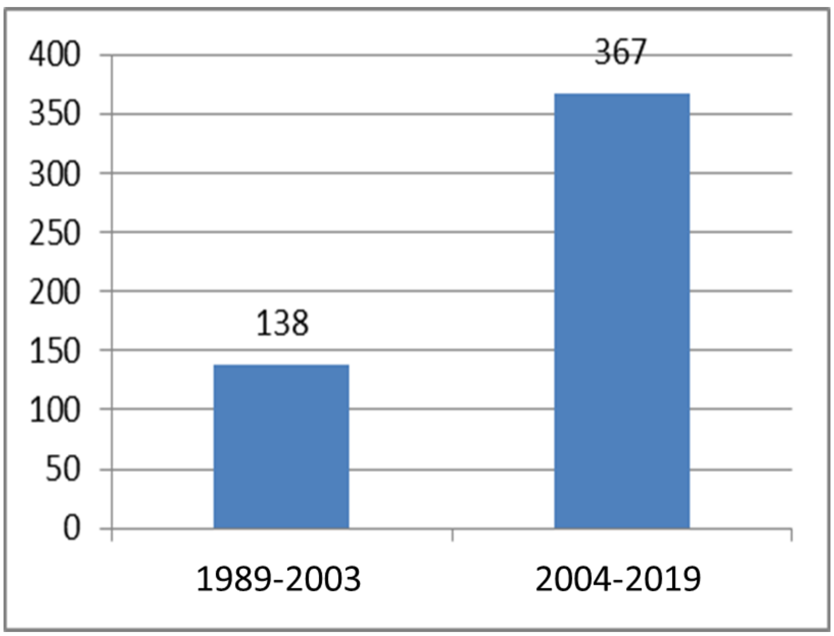


Fig. 4. Publications per period.

TABLE III. THE 10 MOST USED SOURCES[2]

1989-2003		**2004-2019**	
Source	**#**	**Source**	**#**
Initial requirements	17	Stakeholders goals	64

[2] Convention to interpret the tables: elements highlighted (yellow color) in the first period (1989-2003) correspond to elements that do not appear in the second period (2004-2019). Otherwise, elements highlighted (blue color) in the second period correspond to elements that do not appear in the first period.

Domain knowledge	16	Initial requirements	40
Stakeholders goals	16	Domain knowledge	37
Scenarios	10	Business process models	19
Business models	7	Goal models	18
Problem situation	6	Problem situation	16
ERP and business processes	4	Security goals, privacy goals and attacks	14
Goal models	4	Scenarios	13
Situations, arguments, selected strategies and options	4	Business models	11
Use cases	4	Legal texts	9

RQ2. Purpose and target. Both periods present proposals aimed at eliciting FRs, NFRs, FRs, and NFRs or other targets (Fig. 5).

The first period shows a dominant line of works in FRs (61 p.); meanwhile, proposals on NFRs appear in an incipient form (7 p.). The elicitation of FRs and NFRs also appears in an important manner being the second research stream (48 p.). Other works (i. e. groundwork) constitute the third priority for researchers (22 p.). On the other hand, important changes occurred in the second period in comparison to the first one: the main line of research corresponds to FRs and NFRs (147 p.), the second priority is the line of FRs (119 p.), in third place we find research on NFRs (70 p.) and the fourth line of work corresponds to other proposals (31 p.).

RQ3. knowledge and representation used. The main forms of knowledge representation used in both periods are Scenarios and Goal models. In the first period, researchers use Scenarios and Goal models in similar proportions (22 p. and 20 p. respectively). While, in the last period, there is a wide preference for Goal models (78 p.) over Scenarios (32 p.). We also observe that Viewpoint models, Conceptual meta-models, and Goal obstacles are diminishing in their use during the last period. Besides, we note the emergence of Security models, Use cases, and Ontologies. These results are condensed in Table IV.

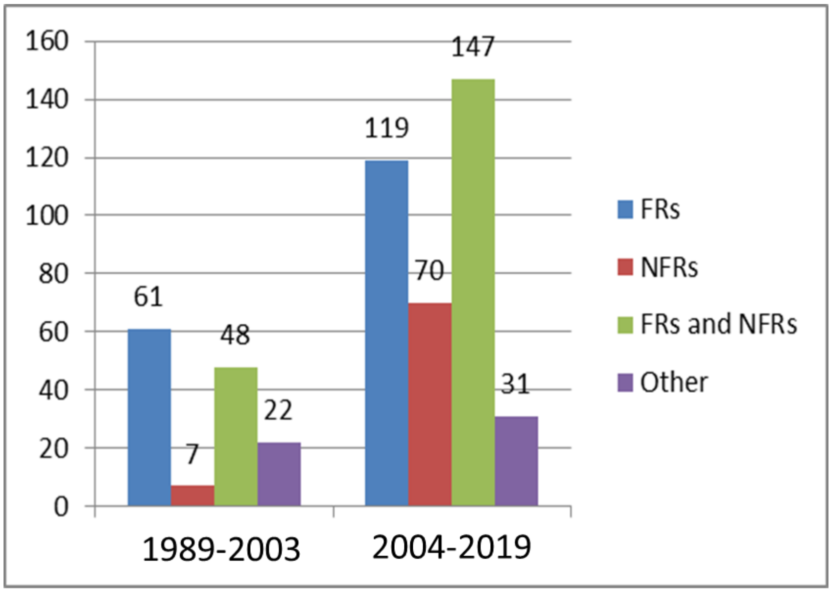


Fig. 5. Proposals target.

TABLE IV. THE 10 MOST USED FORMS OF KNOWLEDGE REPRESENTATION

1989-2003		2004-2019	
Form of knowledge representation	**#**	**Form of knowledge representation**	**#**
Scenarios	22	Goal models	78
Goal models	20	Scenarios	32
Viewpoints models	7	Security models	22
Patterns	6	Use cases	14
Alignment models	4	Natural language	12
Change models	4	Patterns	12
Conceptual meta-models	4	Alignment models	10
Goal obstacles	4	Ontologies	10
Business process models	2	Business process models	9
Natural language	2	Change models	8

RQ4. Methods, resources, and tools. Considering methods, we have grouped the approaches into categories that extend those proposed by van Lamsweerde [7] and Wieringa and Daneva [8]. Table V summarizes these results. We observe that, in the first period, Scenario-based elicitation and validation approaches (32 p.) are more used than Goal-based reasoning approaches (22 p.); nevertheless, this changes drastically in the second period where Goal-based reasoning approaches (77 p.) are preferred over Scenario-based elicitation and validation approaches (38

p.). In contrast with the first period, in the last years, we perceive an increasing interest in methods like Creative-based Approaches for Requirements Elicitation (CAREs), Pattern-based approaches, Quality-model-based approaches, Quality-verification-based approaches, and Requirements reuse among others. Particularly, in the last years, the amount of Creative-based Approaches for Requirements Elicitation (CAREs) (88 p.) evidences an increasing and predominant interest of the RE community for this promising stream of proposals.

Regarding the used resources (see Table VI), we note that Natural language processing techniques and resources, I* framework, and the MAP formalism are important resources in both periods; this could be indicative of their relevance in future proposals. On the other hand, we observe important changes in recent years compared to the first period. Maybe the most outstanding change is the positioning of I* framework as the most used resource by the RE community. We also highlight the raising of other resources like Creativity techniques, KAOS framework, Security models, Analytic Hierarchy Process, and NFR taxonomies.

TABLE V. USED METHODS

1989-2003		**2004-2019**	
Methods	**#**	**Methods**	**#**
Reference-model-based approaches	34	Creative-based Approaches for Requirements Elicitation (CAREs)	88
Scenario-based elicitation and validation	32	Goal-based reasoning	77
Goal-based reasoning	22	Scenario-based elicitation and validation	38
Viewpoints, facets and conflicts	10	Reference-model-based approaches	25
Fitness, alignment or change-oriented approaches	7	Pattern-based approaches	24
Groundwork	7	Quality-model-based approaches	17
Agents-based approaches	4	Fitness, alignment or change-oriented approaches	16
Quality-model-based approaches	4	Process-mining-based methods	13
Ethnography-based approaches	3	Agents-based approaches	12
Pattern-based approaches	3	Commitments, privileges, regulations and rights oriented approaches	12
Process-mining-based methods	3	Reactive-systems-based approaches	8
Responsibilities-based approaches	3	Inquiry-based approaches	7

Commitments, privileges, regulations and rights oriented approaches	1	Ethnography-based approaches	6
Constructionist and organization-theory-based approaches	1	Groundwork	6
Creative-based Approaches for Requirements Elicitation (CAREs)	1	Viewpoints, facets and conflicts	5
Inquiry-based approaches	1	Quality-verification-based approaches	4
Product-line oriented approaches	1	Requirements reuse	4
Reactive-systems-based approaches	1	Responsibilities-based approaches	3
		Product-line oriented approaches	2

TABLE VI. THE 10 MOST USED RESOURCES

1989-2003		**2004-2019**	
Resources	**#**	**Resources**	**#**
Goal decomposition techniques	9	I* framework	39
Natural language processing techniques and resources	7	Natural language processing techniques and resources	14
Heuristics for goal identification	6	Domain knowledge	11
Questionnaires	5	MAP formalism	10
Consistency rules	4	KAOS framework	8
I* framework	4	Problem frames	8
MAP formalism	4	Creativity techniques	7
Categories of goals	4	Security model	7
Guidelines to identify requirements	3	Analytic Hierarchy Process	5
Heuristics to derive requirements from business process models	3	NFR taxonomy	5

Concerning to the tools that support the approaches, we observe an unexpected result; as shown in Fig. 6, the proportion of proposals supported by tools has diminished in the last period (31%) in comparison to the first period (42%). This could be due to different causes and more studies are needed to validate this preliminary result and have a better understanding of this situation.

From Table VII, we can note that only two of the tools are used in both periods: CREWS-SAVRE and L´ECRITOIRE. In the second period, an important number of new tools (8 tools) emerge in comparison to the first period. We also note that several of the most used tools are authored by the same leading researchers: ART-SCENE, CREWS-SAVRE, and Mobile Scenario Presenter correspond to Maiden et al.; NFR-classifier and Collaborative recommender are developed by Cleland-Huang et al.; Gaius T and T-Tool are authored by Mylopoulos et al. The presence of these authors providing different tools is remarkable and could be indicative of the maturity level of some of their works. Other outstanding tools in the last period are L´ECRITOIRE, Rolland et al., and ReqSimile, Natt och Dag et al.

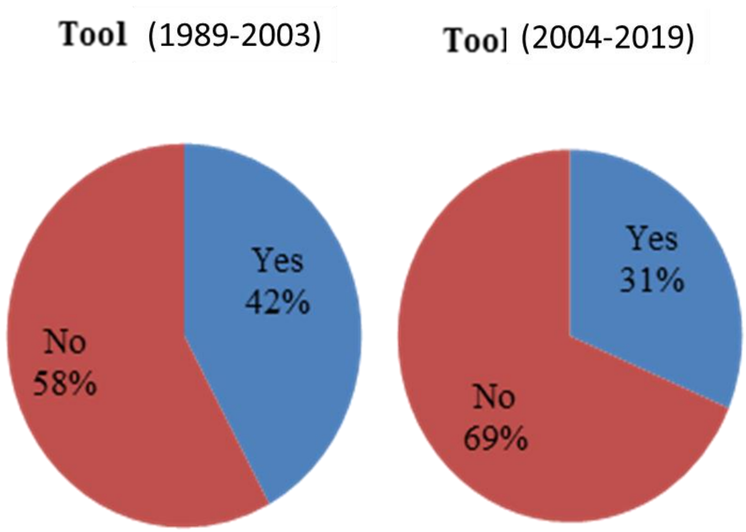


Fig. 6. Tools per period.

TABLE VII. THE 10 MOST USED TOOLS

1989-2003		2004-2019	
Tools	**#**	**Tools**	**#**
L'ECRITOIRE	4	ART-SCENE	6
GRAIL	3	CREWS-SAVRE	4
ACME	2	T-tool	4
AMORE	2	Mobile Scenario Presenter	3
CREWS-PRIME	2	NFR classifier	3
CREWS-SAVRE	2	Cerno	2
OICSI	2	Collaborative recommender	2
SCR tool set	2	GaiusT	2
Ventana GroupSystems	2	ReqSimile	2
Visal basic Prolog tool	2	L'ECRITOIRE	2

Finally, to identify the most influential proposals, we have consulted and consolidated the number of citations of each publication in IEEE, ACM, SCOPUS and Google scholar databases; the main of these results appear in Table VIII. Considering these outcomes, it is important to highlight the influence of the works of van Lamsweerde et al. who has 4 publications in the top ten most cited papers; Mylopoulos et al. 3 p., Scheer 1 p., Yu 1 p. and Rolland et al. 1p. We also note that these publications are mainly related to Agents-based approaches (REA-466, 269, 272), Reference-model-based approaches (REA-423), Groundwork (REA-460), Goal-based reasoning (REA-500, 462), Quality-model-based approaches (REA-461, 278) and Scenario-based approaches (REA-362).

This puts in evidence the noteworthy interest of the RE community for these types of solutions.

TABLE VIII. THE 10 MOST CITED PAPERS

Id.	Proposal	Authors	Citations
REA-466	"Agent-Based Tactics for Goal-Oriented Requirements Elaboration".	van Lamsweerde A. and Letier E.	3201
REA-269	Tropos: An agent-oriented software development methodology.	Mylopolous J, Bresciani P, Giorgini P, Giunchiglia F, and Perini A.	2526
REA-460	Requiremens engineering: From craft to discipline.	van Lamsweerde A.	2451
REA-500	Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering.	Yu E.	2032
REA-423	Aris – Business Process Modeling.	Scheer A. W.	1836
REA-272	Towards Requirements-Driven Software Development Methodology: The Tropos Project'.	Mylopoulos J., Castro J. and Kolp M.	1055
REA-461	Reasoning about confidentiality at requirements engineering time.	van Lamsweerde A. and De Landtsheer R.	1048
REA-278	Representing and Using Non-Functional Requirements: A Process-Oriented Approach.	Mylopoulos J., Chung L., and Nixon B.	1042
REA-462	"GRAIL/KAOS: An Environment for Goal-driven Requirements Engineering".	van Lamsweerde A., Darimont R., Delor E. and Massonet P.	1040
REA-362	Guiding Goal Modeling Using Scenarios.	Rolland C., Souveyet C., and Ben Achour C.	950

2.1.4. Threats to Validity

Considering possible threats to the validity of our SLR we have defined methodological aspects to mitigate them [9]. The construct validity refers to how well the studied parameters and their outcomes are adequate to the research questions addressed. Aimed at obtain relevant approaches to answer the research questions, we used a set of leading scientific research engines (IEEE, SPRINGER, ACM, DBLP, and SCOPUS) and a search query designed to retrieve proposals aligned with our research purpose. We also use inclusion/exclusion criteria (Table I) aimed at minimizing the subjectivity when deciding the publications inclusion or exclusion. Nevertheless, given the fact that the assessment of the proposals was made only by one person, this process is error-prone; this could be tackled in the future considering a multiple evaluation perspective and a conflict resolution process when individuals disagree.

External validity refers to the ability to generalize the research findings obtained to other domain under different settings [9]. In this sense, our SLR provides generalization over the established research scope; the obtained results can´t be generalized to other scopes or domains.

2.1.5. Related Work

We have taken into account works on evidence-based software engineering and SLR in software engineering proposed by Kitchenham et al. [4], [5]. Likewise, threats to validity of our SLR were mitigated following considerations of Feldt and Magazinius [9]. On the other hand, we present remarkable studies aimed at establishing the state of the art of RE for software development: a systematic mapping on creativity in requirements engineering is described by Lemos et al. [10]. Reviews on automated RE and tools are carried out by Nicolás and Toval [11], Sawyer et al. [12] and Maeche et al. [13]. Recent surveys on NFRs and quality are authored by Ullah et al. [14], Pastor et al. [15] and Loucopoulos et al. [16]. Regarding RE in Goal-Oriented Requirements Engineering (GORE) approaches, we can find valuable insights in studies of van Lamsweerde [17], Loucopoulos and Kavakli [18], Lapouchnian [19], Anwer et al. [20] and Horkoff et al [6]. The industrial practice of RE approaches is visited by Yu et al. [21], [22] and Glinz et al. [23]. Finally, overviews of some of the main challenges currently faced in RE are introduced by

Easterbrook and Nuseibeh [2], Cheng and Atlee [3], van Lamsweerde [7], Wieringa and Daneva [8], Singh and Kaur [24], Salinesi et al. [25] and Sutcliffe and Sawyer [26].

2.1.6. Conclusions

In order to understand the progress that has been made in the RE field during the last 30 years, we have conducted a SLR. As a result, we have identified a set of 505 publications, available online, which allowed us to answer our research questions. In the sequel, we present our main findings.

Concerning the **Sources** used by the proposals (RQ1), we found that in the last period, Stakeholders goals are the most used input. Furthermore, we observe the emergence of Business process models, Security goals, privacy goals, and attacks and Legal texts as relevant entries for the approaches.

Regarding the **Purpose and target** of the approaches (RQ2), in recent years (second period) we observe an important increase of proposals in all lines of research: FRs, NFRs, FRs and NFRs, and Groundwork. We also note a growing interest especially in the lines of "FRs and NFRs" and NFRs elicitation; this could be an important indicator of the growing relevance of RE issues not only for academia but also for industry.

As to the **Knowledge and representation used** (RQ3), we found that Goal models are by far the preferred form of knowledge representation in last years, followed by Scenarios, Security goals, Use cases and Ontologies.

Regarding the **Methods, resources, and tools** used by the proposals (RQ4), in recent years we observe a significant growth on the use of Goal-based reasoning approaches; meanwhile, Scenario-based approaches have lost terrain but still are protagonists in the scene. On the other hand, the most remarkable fact is the emergence of new trends like Creative-based Approaches for Requirements Elicitation (CAREs), Pattern-based approaches, Quality-model-based approaches, Quality-verification-based approaches, and Requirements reuse.

During the periods in consideration, Natural language processing techniques and resources, I* framework and MAP formalism have been valuable resources for researchers; this could be indicative of their importance for future works. On the other hand, I* framework has been positioned as the main resource in the last years. We also must remark the emergence of resources like Creativity techniques, Security models, Analytical Hierarchy Process and NFR taxonomies.

Concerning the used tools, we observe an unexpected result: the proportion of proposals supported by tools has decreased in the last period (31%) compared to the first period (42%). This could be due to different reasons and more studies are required to validate this preliminary result and understand its causes. After a revision of the tools used in the first period, we observe that only CREWS-SAVRE and L´ECRITOIRE maintain their presence in the second period. We also observe that several of the most used tools were developed by teams under the leadership of the same researcher: ART–SCENE, CREWS-SAVRE, and Mobile scenario correspond to Maiden et al., NFR-classifier and Collaborative recommender are authored by Cleland-Huang et al. and GaiusT and Tool-T are developed by Mylopoulos et al. This is a possible indicator of the maturity of their works.

Thanks to this SLR we have identified the Creative-based Approaches for Requirements Elicitation (CAREs) as the stream of proposals which is currently of the most interest for the RE community. Aimed at understanding the strengths and limitations of CAREs, we conducted other SLR in this field; this work is presented in the next section.

REFERENCES

[1] O. Pastor, I. Villanueva and J. Sánchez. Elicitación de requisitos en sistemas de gestión orientados a procesos. Proceedings of the 11th Internationa Workshop on Requirements Engineering (WER´ 05). Pp. 38-49. Porto, Portugal. 2005.

[2] S. Easterbrook and B. Nuseibeh Bashar. "Requirements Engineering: A Roadmap". Proceedings of the Conference the Future of Sofware Engineering. ICSE 00. Limerick Ireland 2000.

[3] B. Cheng and J. Atlee Joann. Research Directions in Requirements Engineering. Future of Software Engineering (FOSE'07). 2007.

[4] B. Kitchenham, T. Dyba, and M. Jorgensen, "Evidence-based software engineering," in Proc. of the 26th International Conference on Software Engineering (ICSE04), 2004, pp. 273–281. [Online]. Available: http://dl.acm.org/citation.cfm?id=998675.999432

[5] B. Kitchenham, R. Pretorius, D. Budgen, O. P. Brereton, M. Turner, M. Niazi, and S. Linkman, "Systematic literature reviews in software engineering – a tertiary study," Information and Software Technology, vol. 52, no. 8, pp. 792 – 805, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950584910000467

[6] J. Horkoff, T., F. Li, M. Salnitri, E. Cardoso, P. Giorgini, J. Mylopoulos and J. Pimentel. Taking Goal Models Downstream: A Systematic Roadmap. Research Challenges in Information Science (RCIS), IEEE Eighth International Conference on. Page(s): 1 – 12. 2014.

[7] A. van Lamsweerde. Requirement Engineering in the Year 00: A Research Perspective. Proc. 22nd Int. Conf. on the Software Engineering, University of Limerick, Ireland. 2000.

[8] R. Wieringa and M. Daneva. Requirements Engineering for Enterprise Systems:WhatWe Know and What We Don't Know?. S. Nurcan et al. (eds.), Intentional Perspectives on Information Systems Engineering, 2010.

[9] R. Feldt and A. Magazinius. Validity Threats in Empirical Software Engineering Research-An Initial Survey. In Proceedings of the 22nd International Conference on Software Engineering and Knowledge Engineering, pp. 374-379. 2010.

[10] J. Lemos, C. Alves, L. Duboc and G. Rodrigues. 'A systematic mapping study on creativity in requirements engineering', in Proceedings of the 27th Annual ACM Symposium on Applied Computing, pp. 1083–1088. 2012.

[11] J. Nicolás and A. Toval. On the generation of requirements specifications from software engineering models: A systematic literature review. Information and Software Technology 51(9), 1291–1307. 2009.

[12] P. Sawyer, D. Berry, R. Gacitua and S.F. Tjong. The case for dumb requirements engineering tools, in: Requirements Engineering: Foundation for Software Quality, Springer, Berlin/Heidelberg. pp. 211–217. 2012.

[13] A. Maedche, M. Brhel and H. Meth. The state of the art in automated requirements elicitation. Information and Software Technology 55. 695–1709. 2013

[14] S. Ullah, M. Iqbal and A. Khan. M. A Survey on Issues in Non-Functional Requirements Elicitation. Computer Networks and Information Technology (ICCNIT), 2011 International Conference on. Page(s): 333 – 340. 2011.

[15] Ó. Pastor, Y. Ormeño and J. Panach. Usability requirements elicitation: an overview of a mapping study. INTERACCION '12: Proceedings of the 13th International Conference on Interacción Persona-Ordenador. 2012.

[16] P. Loucopoulos, H. Mahmudul and M. Nikolaidou M. Classification and Qualitative Analysis of Non-Functional Requirements Approaches. Lecture Notes in Business Information ProcessingVolume 175, pp 348-362. 2014.

[17] A. van Lamsweerde. Goal-Oriented Requirements Engineering: A Roundtrip from Research to Practice. Proceedings of the 12th IEEE International Requirements Engineering Conference (RE'04). 2004.

[18] P. Loucopoulos and E. Kavakli. Goal Driven Requirements Engineering: Analysis and Critique of Current Methods, in Information Modeling Methods and Methodologies (Adv. topics of Database Research), John Krogstie, Terry Halpin and Keng Siau (eds), IDEA Group, pp 102 – 124. 2004.

[19] A. Lapouchnian. Goal-Oriented Requirements Engineering: An Overview of the Current Research. Depth Report, University of Toronto, 2005.

[20] S. Anwer, M. Jinnah and N. Ikram. "Goal Oriented Requirement Engineering: A Critical Study of Techniques". In Proceedings of the 13th Asia Pacific Software Engineering Conference, APSEC 2006.

[21] (Yu et al., 2011) Yu Eric, Elahi Golnaz, Li Tong and Liu Lin. Security Requirements Engineering in the Wild: A Survey of Common Practices. 35th IEEE Annual Computer Software and Applications Conference. 2011.

[22] (Yu et al., 2013) Yu E., Amyot D., Mussbacher G., Franch X. and Castro Jaelson: Practical Applications of i* in Industry: The State of the Art. RE 2013.

[23] (Glinz et al., 2013) Todoran Irina, Seyff Norbert, and Glinz Martin. How Cloud Providers Elicit Consumer Requirements: An Exploratory Study of Nineteen Companies. Requirements Engineering Conference (RE) 2013, 21st IEEE International. 2013.

[24] (Singh and Kaur, 2010)Singh Tajinder and Kaur Ranjeet. Analysis and Need of Requirements Engineering. International Journal of Computer Applications (0975 – 8887). Volume 7– No.14, October. 2010.

[25] (Salinesi et al., 2010) Salinesi C., Nurcan S., Souveyet C. and Ralyté J. Intentional perspectives on information systems engineering, Springer, 2010.

[26] (Sutcliffe and Sawyer, 2013) Sutcliffe Alistair, Sawyer Pete. Requirements elicitation: towards the unknown unknowns. RE 2013. Rio de Janeiro. 2013.

2.2. Leveraging Creativity Techniques in Requirements Elicitation: A Literature Review

2.2.1. Introduction

Requirements Engineering (RE) is recognized as a complex cognitive problem-solving process which takes place in an unstructured and poorly-understood problem context [1]. Requirements elicitation is the activity generally regarded as the most crucial step in the RE process. The term "elicitation" is preferred to "capture", to avoid the suggestion that requirements are out there to be collected simply by asking the right questions. Information gathered during requirements elicitation often has to be interpreted, analyzed, modeled and validated before the requirements engineer can feel confident that a complete set of requirements of a system have been obtained [2], [4]. Requirements elicitation comprises the set of activities that enable discovering, understanding and documenting of the goals and motives for building a proposed software system. It also involves identifying the requirements that the resulting system must satisfy in order to achieve these goals. The requirements to be elicited may range from modifications to well-understood problems and systems (i.e. software upgrades), to hazy understandings of new problems being automated, to relatively unconstrained requirements that are open to innovation (e.g. mass-market software) [3].

Elicitation still remains problematic; missing or mistaken requirements still delay projects and cause cost overruns. No firm definition has matured for requirements elicitation in comparison to other areas of RE [5]. A recent understanding describes the RE process as inherently creative, involving cycles of an incremental building followed by insight-driven re-conceptualization of the problem space [1]. Moreover, in the last decade, a line of academic works has recognized the importance of creativity in the requirements definition process. These approaches develop a vision in which requirements should be imagined and invented by stakeholders, instead of being simply "gathered" from them [2], [4]. The relevance of these works, in a context where innovative solutions represent a competitive advantage for companies, is noteworthy.

This chapter explores recent advances in Creativity-based Approaches for Requirements Elicitation (CAREs), examining their contributions to the requirements elicitation problem and the challenges to be faced in the future. In previous work, aimed at understanding the progress that has been achieved in the requirements elicitation domain, a Systematic Literature Review

(SLR) of proposals resulted in 505 publications [6]. As an important result, we identified a promising set of proposals in the field of CAREs. Other studies related to creativity in the RE domain have been carried out i.e. [7], [8]; our research is complementary to them and its main difference consists in the focus on the requirements elicitation domain and creativity techniques. The main contribution of this work is the identification and characterization of an important set of creativity techniques in order to support the work of practitioners and stimulate their adoption.

This chapter is organized as follows: section 2.2.2. introduces creativity in requirements elicitation, section 2.2.3. presents the method we have used in this chapter, section 2.2.4. introduces the results, section 2.2.5 discusses the obtained results and section 2.2.6. considers the research conclusions.

2.2.2. Creativity and Requirements Elicitation

From a cognitive psychology perspective, creativity is "the ability to produce work that is both novel (i.e. original, unexpected) and appropriate (i.e. useful, adaptive concerning task and constraints)" [10]. An idea is creative when it brings a new insight into a given situation [20]. The process of creativity includes the ability to change one's approach to a problem, to produce ideas that are both relevant and unusual, to see beyond the immediate situation, and to redefine the problem or some aspect of it. Creativity as a multidisciplinary research field has been investigated from the perspective of design, arts, psychology, literature, among other areas. In the field of requirements engineering, several authors have emphasized the importance of treating requirements elicitation as a creative problem-solving process [11]. Indeed, while requirements were traditionally considered to exist in an implicit manner in the mind of stakeholders and the analyst´s job is to capture them, this view is now considered to drastically reduce the scope of the requirements engineering phase. Instead, the invention is claimed to be an essential part of the requirements engineering activity, and "requirements analysts are ideally placed to innovate" [21].

Maiden and Robertson in [12] noted a lack of creativity theories and models in current requirements elicitation research and practice. In this chapter, we adopted and were inspired by the framework developed by Nguyen and Shanks [9]. This framework provides a structured means for understanding the role and potential of creativity in requirements engineering. For the purpose of our study, we focus on the product and process perspectives [32]. From a product perspective, three characteristics are essential: novelty (i.e. new, original), value (i.e. helpful, useful) and “surprisingness” (i.e. unusual, unexpected). From a process perspective, Nguyen and Shanks adopt the analysis of Boden in [14] and Shneiderman in [15]. They describe the creative process as an internal process of exploration and transformation of conceptual spaces in the individual mind. They consider three views of a creative process, inspirationalist, structuralist, and situationist:

Inspirationalist view tends to study how insight, the magical ‘‘Aha!” moment, occurs and emphasize an individual’s creative cognitive processes. Wallas’s model [16] is the dominant inspirationalist creative process model. Wallas describes his creative process model as consisting of four stages: preparation, incubation, illumination (insight), and the verification and expression of insight. Creativity involves conscious and unconscious mental processes and insight is seen as a breakthrough of unconscious ideas.

Structuralist view is influenced by an alternative theory to problem-solving which emphasizes a rational, systematic and structured search for information and the evaluation and selection of alternative solutions. The core of structuralist creative processes lies in the deliberate generation and evaluation of ideas. Therefore, a structured, guided process of divergent and convergent thinking exists in various structuralist process models.

Situationalist view emphasizes the role of the human and social environment and professional domain in the creative collaborative process. Overall, the situationalist view incorporates the communication of creative ideas within teams and thus has the potential to be aligned within core requirements elicitation activities including requirements communication, negotiation, and agreement [4].

2.2.3. Method

This chapter focuses on techniques used in creativity-based approaches for requirements elicitation. The term technique is used here to denote any specific way of handling, conducting or managing the requirements elicitation task. In the first place, we have taken the set of CAREs identified in a previously published systematic review of the requirements elicitation literature [6]. Using the snowball technique to identify and select relevant studies in the literature, each of the articles was scanned aiming to find new CAREs in their references. Once a candidate paper was found we read the abstract to validate the criteria: (1) the paper must deal with a proposal for CAREs and (2) the paper must be published in a recognized scientific database i.e. IEEE, ACM, Springer among others. Moreover, we compared the list of papers we obtained with those included in a similar literature review publication [7]. At the end of this process, we obtained the final set of papers dealing with CAREs (see Appendix 1). Analyzing carefully each paper, we extracted and identified the creativity technique that is proposed and used in the paper. The techniques are classified according to two dimensions, product (i.e. representation) vs. process. A technique is *representation-oriented* if it seeks to stimulate creativity by introducing a specific manner for describing the result of the requirements elicitation task. It is *process-oriented* if it defines a specific manner for handling the requirements elicitation task. Three subcategories are defined: *organizational* if the technique prescribes how the elicitation process is to be conducted and organized in order to be creative, *psychological* if the technique prescribes a way to psychologically stimulate the participant's ability to be creative; and *cognitive* if the technique prescribes a way to stimulate the participant's cognitive capacities in being creative.

2.2.4. Results

We have selected 23 papers that present creativity-based approaches for requirements elicitation. They are presented in Appendix 1 and are numbered A1, A2, etc. From this set of papers, we have identified 30 different creativity techniques that are explicitly defined and used in these studies. The techniques are presented here according to the two perspectives: product and process. For each technique, we provide a brief definition, and when available, a short example that illustrates the applicability of the technique. This information is synthesized from the corresponding paper in Appendix 1.

2.2.4.1. Representation-oriented Techniques

These techniques focus on the means by which requirements are represented. Creativity is expected to be stimulated because models help in capturing requirements in original and innovative ways.

Creativity technique: Topic maps	
Definition: a topic map consists of a set of nodes, linked by associations. A node may fill a specific role in an association. The representation mechanism also supports the subclass relation and the instance-of relation [A11], [22].	**Usage example:** in [A11], the knowledge associated to a "selling goods" problem is represented as topic maps which are used to support the reasoning and generation of new ideas from a proposed set of heuristics. Other examples can be found in [22].
Creativity technique: Goal modeling	
Definition: consists in the use of goal models during RE activities as part of a creativity methodology guided by tool-support [A7]. Many goal-oriented notations for eliciting requirements have been proposed in the last decade, interested readers can see literature reviews on this subject (e.g. [17]).	**Usage example:** in [A7], the authors present a running example of train transport from London airports, specifically, the purchase of tickets, which offers the possibility of many different train services at different prices and routes, and can be confusing to visitors. Using the goal-oriented notation i*, models of the problem are built and used as entry to creativity techniques which allow the reasoning on these models and the idea generation.
Creativity technique: Scenarios	
Definition: as the name suggests, scenarios are narrative and specific descriptions of current and future processes including actions and interactions between the users and the system [18], [23]. NB: we have classified the Scenarios also as a	**Usage example:** in [A3], the authors foster creativity exploring the different uses of scenarios on requirements discovery using results from requirements processes in two projects. The first specified requirements on a new aircraft

cognitive technique because writing a concrete scenario helps stakeholders in better grasping the functional and non-functional requirements for future systems.	management system at a regional UK airport to reduce its environmental impact. The second specified new work-based learning tools to be adopted by a consortium of organizations. Other examples can be found in [23].

Creativity technique: Storyboard	
Storyboard is an extended version of the Scenarios technique. Beyond textual descriptions for a scenario, it combines visual representations with graphics and text to describe system behaviors in a concrete form directly observable by stakeholders [A4], [24].	**Usage example:** in [A4] storyboarding is used to generate requirements for a security access system scenario. Other examples can be found in [24].

2.2.4.2. Process-oriented Techniques

We present the process-oriented techniques according to three subcategories: organizational, cognitive and psychological.

2.2.4.2.1. Organizational Perspective

Techniques in this category concern the way the requirements elicitation process is to be organized. We find here well-known techniques such as creativity workshops and brainstorming sessions; other more specific techniques seek to take stakeholders out of the usual brainstorming protocol and create new settings for collectively exploring the requirements elicitation problem. These techniques are generally combined with other techniques from the representation perspective (e.g. goal modeling) or the cognitive process perspective (e.g. analogies).

Creativity technique: Creativity Workshops	
Definition: a Workshop is a generic term given to a number of different types of group meetings where	**Usage example:** Within health care domain, a group of diabetes patients, doctors and nurses

the emphasis is on collectively developing and discovering requirements for a software system [18].	created visions about the technology and how they could be helped in their daily management of the disease. They collaboratively constructed a conceptual text-based scenario landscape relating to the participants' common situations and problems [A19].
Creativity technique: Brainstorming	
Definition: Brainstorming is a process where participants from different stakeholder groups engage in informal discussion to rapidly generate as many ideas as possible without focusing on any one in particular. It is important when conducting this type of group work to avoid exploring or critiquing ideas in great detail [18].	**Usage example:** In [A7], using the London Airport Trains system, the authors illustrate the use of brainstorming in conjunction with goal models in order to help users to build and fill in the details of i* models.
Creativity technique: Roles playing	
Definition: Consists in the use of different roles (e.g. explorer, artist, judge, and warrior) to focus the participants on ideas generation from diverse and unexpected perspectives [A1].	**Usage example:** In a software project in the Air traffic management domain, workshop participants were encouraged to play each other's controller, pilot, and manager roles to generate ideas from unencumbered perspectives [A1].
Creativity technique: Walt Disney	
Definition: This technique decomposes the creative process into three different steps called Dreamer, Critique, and Realist, respectively. Each of these steps would usually lead to prolonged sessions, which could easily need several hours. While the basic goals of these steps are well defined, their detailed performance is hardly defined and no detailed guidance on their performance is	

given [A10].	
Creativity technique: Game mechanics	
Definition: This technique provides levels and goals, which can be in the form of awards, credits and acknowledgements, in order to motivate and engage participants in the creative problem-solving process [A5].	**Usage example:** COLLAGE is an EU-funded Integrated Project, to inform and enable the design of effective Web 2.0 social creativity and learning technologies and services. Game mechanics are employed as a means to set intermediate goals in the overall search space that will both guide and engage problem solvers in further creative activities. Just as a game has levels that one tries to achieve, so should each creative search activity be informed by specific goals; game mechanics are used to provide these goals. Each subspace reveals a new goal that compels the problem solver to continue their creative search activity [A5].
Creativity technique: Positive space design	
Definition: Space design refers to the context in which creativity takes place, including its environment, place, situation and climate. It can also refer to the environment the person is in, the product that is produced or the process that takes place, and explains the interaction between the person and situation that can promote (positive) or inhibit creativity [A6].	**Usage example:** In [A6], the authors illustrate the design of a positive space which included colorful, round-shaped furniture at different heights, a bed and vivid cushions that could support people standing, sitting in different positions and lying. The intention was to create a feeling of being at home. Other features of the positive space included hanging handmade lanterns to decrease the ceiling height, colorful pictures with positive themes such as food, nature, happiness, excitement, and people, pot plants, and windows to provide views of nature
Creativity technique: Hall of fame	
Definition: This creativity technique helps the	**Usage example:** In [A23], the authors illustrate and

participant to take a step away from the most obvious and reasonable perspectives by consulting the world great minds. The participants used these characters to force connections and generate new requirements for their projects by consulting the famous people [A23], [25].	evaluate the Hall of fame and Idea box techniques. In total, 34 creativity workshops were conducted with 90 students from two universities, and 86 industrial practitioners from six companies. The results indicate that Hall of Fame was the technique that led to the greatest number of requirements that were included in future releases of the products. Other examples can be found in [25].

Creativity technique: Creative spaces for conversations	
Definition: Consists in the use of spaces that embrace creative conversations. The spaces bring together many project stakeholders and the creative team to address the social challenge in creative and innovative ways [A18], [26].	**Usage example:** The company Uscreates (UK) supports public-sector behavior change programs. During the design stage, Uscreates designs spaces that embrace creative conversations. The spaces bring together many project stakeholders including local authorities, community groups, and members of the target audience, and the creative team to address the social challenge in creative and innovative ways. Other examples can be found in [26].

2.2.4.2.2. Cognitive Perspective

This category includes the largest number of techniques. This is easily explained by the fact that creativity is essentially a cognitive ability [13]. Some of these techniques are renown in other fields of research, e.g. constraint removal in innovation management; other techniques have been specifically designed for requirements elicitation.

Creativity technique: Constructivist learning	
Definition: Constructivist learning refers to two knowledge building mechanisms: assimilation and	**Usage example:** Two experiential digital simulations (simulation of interviews system and

accommodation. Through assimilation, the learner interprets and incorporates new knowledge into an existing conceptual framework representing his or her knowledge of a topic area. Accommodation occurs when the learner could not fit the new learning into his or her existing framework, as a result he or she reframes (restructures) the existing conceptual framework [A13].	simulation of requirements analysis system) are used as a proof of concept. Learning from these case studies suggests that both systems analyst and business users can be stimulated to be active learners in their discovery of problem, creative ideas and problem solutions in requirements elicitation and discovery.
Creativity technique: Analogies	
Definition: Analysts use previous experience in similar domains as a discussion template for facilitating group work and conducting interviews. Analogies and abstractions of existing problem domains can be used as baselines to acquire specific and detailed information, identify and describe possible solution systems [18].	**Usage example:** In the air traffic management (ATM) domain, in order to explore new ideas for conflict resolution, the authors invited a textile expert to discuss Indian textile design and a musician to discuss modern music composition. Participants were encouraged to find analogies between these domains and ATM, then to generate new ideas about conflict resolution using those analogical elements [A1].
Creativity technique: Presenting solution space knowledge	
Definition: In this technique people change the solution space in a way that things that were considered impossible are now possible [A1].	**Usage example:** The authors worked with Eurocontrol, the organization overseeing European air space. This organization has a complex sociotechnical system named CORA-2. Air traffic controllers will resolve aircraft conflicts using resolutions and advice from the CORA-2 software. The authors wanted CORA-2 requirements to specify how controllers should work and interact with the software system as well as how the software system should function—for example, how to increase automated support for controllers without deskilling them. In this context, the

	facilitators encouraged participants to change the CORA-2 solution space to make possible ideas that participants once considered impossible [A1].
Creativity technique: Constraints removal	
Definition: This technique identifies and challenges the current constraints of the system in order to eliminate them in a new solution [A1].	**Usage example:** In [A6], the authors encouraged two design groups to identify multiple constraints on a supermarket car park service, select constraints to eliminate, diminish or reinterpret, and then generate new ideas in the less-constrained ideas spaces.
Creativity technique: Combining ideas	
Definition: This technique creates new ideas from a combination and synthesis of existing ideas [A1].	**Usage example:** Combinatorial creative thinking is encouraged by means of different strategies e.g. randomly introducing unexpected items into the scenarios. The facilitators encouraged participants to investigate pairs of existing requirements and ideas to create new ones from unforeseen combinations [A1].
Creativity technique: Walkthroughs	
Definition: The requirements engineer leads stakeholders through a segment of documentation and the participants ask questions and make comments about possible errors, violations, omissions and other problems. The big idea behind walkthroughs is very simple: people are better at recognition than recall [A4].	**Usage example:** Each analyst walked through the scenario to discover and document requirements for the security access system. Next, the experimenter seeded the software tool combin-Formation with queries. The analyst continued the walkthrough with the tool to discover and document requirements [A4]. Other examples can be found in [27].
Creativity technique: Viewpoints	

Definition: This technique aims to model the domain from different perspectives in order to develop a complete and consistent description of the target system [18].	**Usage example:** Two experiments in [A8] illustrate the proposal. The first one corresponds to the development of the Corsi Online system (a Web application to help manage on-line courses for a university) and the second one was conducted at the software company that had developed Civilia to support community services for citizens [A8].
Creativity technique: Deconstruction	
Definition: The basic idea of this technique is to start with the usual perception the developers had of their products and step-by-step remove certain constituents. Then the participants would need to replace them with something else [A10].	**Usage example:** In [A10], the authors illustrate the use of the deconstruction, questions list and Walt Disney techniques with a case study corresponding to the development of an automatic web content creation system.
Creativity technique: Questions list	
Definition: Consists in the use of questions as a means to support divergent thinking [A10].	**See usage example for technique** "Deconstruction".
Creativity technique: Heuristics	
Definition: A heuristic technique is a rule of thumb, strategy, trick, simplification, or any other kind of device which drastically limits search for solutions in large problem spaces. Heuristics do not guarantee optimal solutions; in fact, they do not guarantee any solution at all; all that can be said for a useful heuristic is that it offers solutions which are good enough most of the time [19].	**Usage example:** The use of heuristics is exemplified in [A11] where the authors propose a set of heuristics which support the reasoning and generation of new ideas from a problem knowledge ("selling goods") represented as topic maps. Other examples can be found in [22].
Creativity technique: Why why why?	
Definition: This technique urges the users to	**Usage example:** The use of this technique is

constantly question the motivation for each element of a system design. This, obviously, can help the user to move up the model, adding higher level intentions until the why question is no longer sensible [A7].	showed by means of the train transport from London airports [A7]. I* models of the problem are built and used in conjunction with why questions urging users to constantly question the motivation for each element of the goal models.
Creativity technique: Idea box	
Definition: The Idea Box technique starts by stating a challenge, followed by selecting the parameters of the stated challenge. Then, a list of options for each parameter is created, and finally, the participant should try different combinations to find new concepts and requirements [A23], [28].	**See usage example for technique** "Hall of Fame" in section 4.2.1. Other examples can be found in [28].
Creativity technique: Design rationale	
Design rationale, in simple words, is information which represents and explains the reasoning behind the requirements engineering process [A12].	
Creativity technique: Physualization	
Definition: Physualization is the physical manipulation of visualization entities – it is not just visualization for the sake of communicating or creating a record. Physualization actively promotes physical manipulation to help participants explore possibilities in the requirement and design spaces by engaging more of their sensory and cognitive processes – possibly leading to improvements in the requirements process and resulting artifacts [A15].	**Usage example:** The author presents examples taken from work performed in gathering requirements for video games. The focus in the work sessions was on capturing the intended user experience in general, and the intended emotional experience in particular [A15].
Creativity technique: Speed modeling	

Definition: It is a 3D form of brainstorming. Rather than exploring a subject area or question through a typical, written brainstorming, a facilitator poses a number of questions that participants have to answer by modeling with Plasticine within 30 seconds to three minutes. The time limit encourages quick thought and prevents participants from feeling conscious of the quality of their creativity and ability to express it [A18], [26].	**See usage example for technique** "Creative spaces for conversations" in section 4.2.1 Other examples can be found in [26].
Creativity technique: Picture stimulation	
Definition: In this technique a particular problem or idea can be reinvented using different angles motivated from pictures [A19]	**Usage example:** Picture stimulation and Cultural probing techniques were used in projects MAGNET (2004-2005) and MAGNET Beyond (2006-2008), part of the IST EU program, to express wishes and needs in the requirements identification [A19].
Creativity technique: Cultural probing	
Definition: Cultural probing is a technique which allows the user to carry around a probe designed to provoke inspirational responses in different circumstances [A19].	**See usage example for technique** "Picture stimulation"

2.2.4.2.3. Psychological perspective

The idea behind these techniques is to stimulate the individual's state of mind (e.g. humor, emotions, etc.) in order to enhance the cognitive abilities and encourage creativity. We identified only one technique in this category.

Creativity technique: Influence positive emotions	
Definition: Positive emotions technique seeks to adapt the way people think and act such that creativity during idea generation is augmented. Interactive systems can be designed to stimulate and influence participants' emotions and get more out of their own creative capabilities [A22].	**Usage example:** de Rooij et al. developed an interactive system in order to validate whether this system can be used to hack into the function of cognitive appraisal processes in emotion, positive emotions in particular, and that this can be used to augment creative ideation. Their findings show that effectively, an interactive system can be used to augment creative ideation [A22].

2.2.5. Discussion

The techniques we have presented browse a large spectrum of ideas for stimulating creativity and helping requirements engineering in finding ways to take stakeholders out of conventional settings. For maximum efficacy, these techniques need to be combined together. Indeed, most of the studies combine many techniques; only 5 papers rely on one single technique. On the other hand, a small subset of techniques (e.g. scenarios and brainstorming) is well-known in the requirements elicitation domain and these techniques are applied in many studies. However, a large set of techniques can be considered as "outsiders", i.e. they are at the stage of academic proposals, and have been applied in the context of only one single study (e.g. Topic maps, Cultural probing, etc.). Nevertheless, the profusion of creativity techniques illustrates their growing importance for the requirement engineering community.

2.2.5.1. Advantages and Limitations of CAREs

With no doubt, the main advantage of CAREs is the number of requirements generated in a short time. Additional benefits include: the diversity of generated ideas from different stakeholders which typically differ from those generated with "standard" requirements elicitation techniques; shaking people out of tried and tested ways of thinking about requirements; Improvement of communication between stakeholders and Improvement of organizational climate which facilitates the creative processes, among others.

On the other hand, one of the main drawbacks of these techniques is the difficulty to implement in industrial settings due to the fact that a lot of resources and effort are required, which results expensive for companies. Indeed, participants of the experiments often reported having difficulty with using some these techniques. Furthermore, some authors argued that due to time-to-market constraints, there was insufficient time for creative thinking during requirements elicitation activities. Besides, other remarkable issues identified in the CAREs state of the art are [A9], [29], [30], [31]:

a. There is a lack of mechanisms aimed to ensure the alignment of the discovered requirements with the system purpose.

b. The current proposals rely on the Requirements Engineer´s knowledge, which in many cases is intuitive, and experience using a particular technique(s). This leads to a process which is inefficient and prone to errors. Therefore, there is a need for the systematic support and guidance of the Requirements Engineer in the requirements discovery process.

c. The current solutions do not facilitate the transition from the discovered requirements towards the future system specification.

These advantages and limitations should be considered by researchers and practitioners when selecting CAREs. Likewise, techniques limitations introduce important challenges to be faced by researchers and practitioners in the future.

2.2.6. Conclusions

In order to explore recent advances in Creativity-based Approaches for Requirements Elicitation (CAREs), we have conducted a literature review and identified a set of 30 techniques to support creativity in requirements elicitation. Inspired by the framework of Nguyen and Shanks [9], these techniques were characterized according to two perspectives: product and process. The process perspective is specialized further to capture organizational, cognitive and psychological facts of the requirements elicitation process. Some of these techniques are well-known for requirements elicitation; however, most of them are new and illustrate the growing interest of the research community. Although some techniques have been applied on real world projects and in an industrial context, for most of these techniques, the available evidence is insufficient to demonstrate their feasibility and added value.

The next chapter presents a proposed solution to some of the problems of CAREs (issues a, b, and c) identified in this SLR.

References

[1] Nguyen L. and Swatman P. A. Promoting and Supporting the Creative and Insight-Driven RE Process Using Design Rationale. Working Paper, 15, UNISA, Adelaide, 2006.

[2] Ben Achour C. and Rolland C. Introducing genericity and modularity of textual scenario interpretation in the context of requirements engineering. Centre de Recherche en Informatique. Université de Paris I – Sorbonne. 1997.

[3] Cheng B. H. C. and Atlee J. M. Research Directions in Requirements Engineering. Future of Software Engineering. FOSE´07. 2007.

[4] Nuseibeh B. and Easterbrook S. "Requirements Engineering: A Roadmap". Proceedings of the Conference the Future of Sofware Engineering. ICSE 00. Limerick Ireland 2000.

[5] Sutcliffe A. and Sawyer P. Requirements elicitation: towards the unknown unknowns. Proceedings of the RE Conference 2013. Rio de Janeiro, Brazil, 2013.

[6] Jaramillo A. Requirements elicitation approaches: a systematic review. Research Challenges in Information Science (RCIS), IEEE 9th International Conference on. Athens. Greece. 2015.

[7] Lemos J., Alves C., Duboc L. and Rodrigues G. N. ‘A systematic mapping study on creativity in requirements engineering’, in Proceedings of the 27th Annual ACM Symposium on Applied Computing, pp. 1083–1088. 2012.

[8] Grube P. and Schmid K. Selecting Creativity Techniques for Innovative Requirements Engineering. MERE 2008.

[9] Nguyen L. and Shanks G. A framework for understanding creativity in requirements engineering. Information and software technology. 655-662. 2009.

[10] Sternberg R. J. and Lubart T. Defying the Crowd: Cultivating Creativity in a Culture of Conformity, Free Press, 1995.

[11] Maiden N., Jones S., Karlsen K., Neill R. and Milne A. Requirements Engineering as Creative Problem -Solving: A Research Agenda for Idea Finding. 18th IEEE International Requirements Engineering Conference (RE), Sydney, 2010. Available at:

https://core.ac.uk/download/files/146/18294869.pdf (retrieved 28/03/2016)

[12] Maiden N. and Robertson S. Integrating creativity into requirements engineering process: experiences with an air traffic management system, in: 13th IEEE International Conference on Requirements Engineering (RE'05), Paris, France, 2005.

[13] Sternberg R. J. The nature of creativity. Creativity Research Journal, Vol. 18, No. 1, 87–98. 2006.

[14] Boden M. A. The Creative Mind: Myths and Mechanisms, Basic Books, Inc., New York, NY, 1991.

[15] Shneiderman B. Creating creativity: user interfaces for supporting innovation, ACM Transactions on Computer–Human Interaction (TOCHI) 7 (1). 114– 138. 2000.

[16] Wallas G. The Art of Thought, Jonathan Cape, London, England, 1926.

[17] van Lamsweerde A. Goal-oriented requirements engineering: a roundtrip from research to practice [engineering read engineering]. In: Proceedings 12th IEEE International Requirements Engineering Conference. pp. 4–7 (2004).

[18] Zowghi D. and Coulin C.: Requirements Elicitation: A Survey of Techniques, Approaches, and Tools. In: Aurum, A. and Wohlin, C. (eds.), Engineering and Managing Software Requirements. pp. 19 46. Springer Berlin Heidelberg. 2005. Available at:https://opus.lib.uts.edu.au/bitstream/10453/11626/1/2005003295.pdf (retrieved 28/03/2016)

[19] Feigenbaum E. A. and Feldman. Computers and Thought. J. (Eds.) New York: McGraw-Hill, 1963.

[20] Kneller G. F. The Art of Science and Creativity. New York, NY: Holt, Rinehart & Winston, Inc. 2005.

[21] Robertson J. Requirements analysts must also be inventors, IEEE Software (January, February) 2005.

[22] El-Sharkawy S. and Schmid K. A Heuristic Approach for Supporting Product Innovation in Requirements Engineering: A Controlled Experiment. Available at: http://www.klausschmid.net/wp-content/uploads/paper/El-SharkawySchmid11c.pdf (retrieved 28/03/2016).

[23] Jones S., Maiden N. and Karlsen K. Creativity in the specification of large-scale socio-technical systems. Available at:http://openaccess.city.ac.uk/2803/1/JonesMaidenPaper.pdf (retrieved 28/03/2016).

[24] Pennell L. and Maiden N. Creating Requirements – Techniques and Experiences in the Policing Domain. Available at:http://crinfo.univ-paris1.fr/REFSQ/03/papers/P02-Pennel.pdf (retrieved 28/03/2016).

[25] Hollis B. and Maiden N. Extending Agile Processes with Creativity Techniques. Available at:http://openaccess.city.ac.uk/4025/1/Extending%20Agile%20Processes.pdf (retrieved 28/03/2016)

[26] Cook M. R. The emergence and practice of co-design as a method for social sustainability under new labour. Available at:https://core.ac.uk/download/files/96/16424120.pdf (retrieved 28/03/2016)

[27] Mavin A. and Maiden N. Determining socio-technical systems requirements: experiences with generating and Walking through scenarios. Available at: https://www.city.ac.uk/__data/assets/pdf_file/0006/81438/RE03_Camera-Ready_Paper.pdf (retrieved 28/03/2016)

[28] Sprague T. L. Adopt a tool: morphological box.

http://creativesprague.weebly.com/uploads/6/0/9/1/60910867/_morphologicalbox.pdf (retrieved 28/03/2016)

[29] Jaramillo, A., Assar, S. Leveraging creativity techniques in requirements elicitation: a literature review. Requirements Engineering Magazine. International Requirements Engineering Board. Issue 2016-02. 2016.

[30] Sakhnini V., Mich L., Berry D. M. Group versus individual use of power-only EPMcreate as a creativity enhancement technique for requirements elicitation. Springer Science+Business Media New York. 2017.

[31] Herrmann, A., Mich, L., and Berry, D.M. Creativity Techniques for Requirements Elicitation: Comparing Four-Step EPMcreate-Based Processes, Seventh Workshop on Empirical Requirements Engineering (EmpiRE), Banff, AB, Canada, August 2018.

[32] Rolland C. ¿Are Methods Really Useful? Proceedings of the 20th International Conference on Software Engineering ICSE'98, Kyoto, Japan, April 1998.

Appendix 1. Analyzed CAREs aiming to answer the research questions.

[A1] Maiden N., Gizikis A. and Robertson S. Provoking Creativity: Imagine What Your Requirements Could be Like. IEEE Software, vol. 21, issue 5, 2004. Available at:
http://www.volere.co.uk/pdf%20files/IEEE_Software_Creativity_paper.pdf (retrieved 28/03/2016)

[A2] Maiden N. and Zachos K. Inventing Requirements from Software: An Empirical Investigation with Web Services. RE 2008.

[A3] Maiden N., Seyff N., I. Karlsen K., Lockerbie J., Grünbacher P., Graf F. and Ncube C. Exploring how to use scenarios to discover requirements. Requirements Engineering Journal. 2009.

[A4] Maiden N.A.M., Karlsen K. and Kerne A. Inventing Requirements with Creativity Support Tools. REFSQ Requirements Engineering Foundation for Software Quality. 2009.

[A5] Maiden N., Zachos K. and Webster S. An Emerging Computational Model of Flow Spaces to Support Social Creativity. Proceedings of the Fourth International Conference on Computational Creativity. 2013. Available at:http://www.computationalcreativity.net/iccc2013/download/iccc2013-webster-zachos-maiden.pdf (retrieved 28/03/2016)

[A6] JMaiden N. and Nouri M. Exploring Associations between the Work Environment and Creative Design Processes. Andrzej M.J. Skulimowski (ed.): Proceedings of KICSS'2013, pp. 449-460. Progress & Business Publishers, Kraków, 2013. Available at:http://da.xmachina.gr/Proceedings/KICSS2013/files/papers/paper11.pdf (retrieved 28/03/2016)

[A7] Maiden N. and Horkoff J. Supporting Creative RE with i*. Proceedings of the Eighth International i* Workshop (istar 2015), CEUR Vol-978. 2015. Available at: http://ceur-ws.org/Vol-1402/paper1.pdf (retrieved 28/03/2016)

[A8] Berry D., Mich L. and Anesi C. Applying a pragmatics-based creativity-fostering technique to requirements elicitation. Requirements Engineering Journal. 2005. Available at: https://cs.uwaterloo.ca/~dberry/FTP_SITE/reprints.journals.conferences/mich.anesi.berry.rej.pdf (retrieved 28/03/2016)

[A9] Berry D., Mich L. and Sakhnini V. The effectiveness of an optimized EPMcreate as a creativity enhancement technique for web site requirements elicitation. Requirements Engineering Journal. 2012.

[A10] Schmid K. A Study on Creativity in Requirements Engineering. Softwaretechnik-Trends, 26(1), 2006. Available at:http://pi.informatik.uni-siegen.de/stt/26_1/01_Fachgruppenberichte/RE/09_schmid.pdf (retrieved 28/03/2016)

[A11] Schmid K. Reasoning on Requirements Knowledge to Support Creativity. MARK 2009.

[A12] Nguyen L. and Swatman P. A. Promoting and Supporting the Creative and Insight-Driven RE Process Using Design Rationale. Working Paper, 15, UNISA, Adelaide, 2006. Available at: https://www.researchgate.net/publication/240814600_Promoting_and_Supporting_the_Creative_and_Insight-Driven_RE_Process_Using_Design_Rationale (retrieved 28/03/2016)

[A13] Nguyen L. and Cybulski J. Into the Future: Inspiring and Stimulating User´s Creativity. PACIS 2008. Available at: http://www.pacis-net.org/file/2008/PACIS2008_Camera-Ready_Paper_203.pdf (retrieved 28/03/2016)

[A14] Saeki M., Ohshiro K. and Watahiki K. Integrating an Idea Generation Method into a Goal-Oriented Analysis Method for Requirements Elicitation. APSEC Asia Pacific Software Engineering Conference. 2005.

[A15] Callele D. Physualization: Going Beyond Paper Prototyping. REV 2010. Available at: http://experiencefirstdesign.com/wp/wp-content/uploads/2012/12/05625660.pdf (retrieved 28/03/2016)

[A16] Kerkow, D., Adam, S., Riegel, N. and Uenalan, O. A Creativity Method For Business Information Systems. Kaiserslautern VII, 2010.

[A17] Millard, N., Lynch, P. and Tracey, K. Child´s Play: Using Techniques Developed to Elicit Requirements from Children with Adults. RE 1998.

[A18]Cook M. R. Creative Requirements Conversations. IEEE Software, Vol 27, Issue 2, 2010.

[A19] Sorensen L. and Olesen H. Creativity in Requirement Identification. WWRF 2008. Available at:http://vbn.aau.dk/files/14850327/creativity_ottawa_feb_2008_final.pdf (retrieved 28/03/2016)

[A20] Bhowmik T., Niu N., Mahmoud A. and Savolainen J. Automated Support for Combinational Creativity in Requirements Engineering. RE 2014, Karlskrona, Sweden, 2014.

[A21] Rayasam S. and Niu N. Using i∗ for Transformational Creativity in Requirements Engineering. Proceedings of the Eighth International i* Workshop (istar 2015), CEUR Vol-978, 2015.

[A22] de Rooij A., Corr P. J. and Jones S. Emotion and Creativity: Hacking into Cognitive Appraisal Processes to Augment Creative Ideation. ACM SIGCHI Conference on Creativity and Cognition. Glasgow, Scotland. 2015.

[A23] Svensson R. B. and Taghavianfar M. Selecting Creativity Techniques for Creative Requirements: An Evaluation of Four Techniques using Creativity Workshops. RE 2015, Ottawa, ON, Canada, 2015.

Appendix 2. Detailed analysis of the selected CAREs

ID																														
	Topic maps	Influence positive emotions	Goal models	Constructivist learning	Workshops	Brainstorming	Analogies	Presenting solution space knowledge	Removing constraints	Combining ideas	Roles	Scenarios	Walkthroughs	Viewpoints	Deconstruction	Questions list	Walt Disney	Heuristics	Game mechanics	Positive space design	Why why why?	Hall of fame	Idea box	Design rationale	Physualization	Storyboard	Creative spaces for conversations	Speed modeling	Picture stimulation	Cultural probing
A1					X	X	X	X	X	X	X	X																		
A2					X		X					X	X																	
A3					X				X			X	X																	
A4												X	X																	
A5																			X											
A6					X	X			X				X							X										
A7			X			X		X		X											X									
A8						X								X																
A9						X								X																
A10															X	X	X													
A11	X							X										X												
A12																								X						
A13				X																										
A14			X			X																								
A15																									X					
A16					X	X										X										X				
A17						X					X	X														X				
A18																											X	X		
A19					X	X						X																	X	X
A20										X																				
A21			X																											
A22		X																												
A23						X			X													X	X							

CHAPTER 3

Proposed solution: DREQUS: an approach for the Discovery of REQuirements Using Scenarios

3.1. Introduction

Requirements Engineering is the Software Engineering (SE) field whose purpose is to achieve the satisfaction of the needs of customers and users of a future system. During Requirements Engineering, needs are elicited, negotiated, validated, and specified in requirements documents.

Requirements Elicitation (RE) involves discovering the requirements that the resulting system must satisfy in order to achieve the stakeholders´ goals. The requirements to be elicited may range from modifications to well-understood problems and systems (e.g. software upgrades), to hazy understandings of new problems being automated, to relatively unconstrained requirements that are open to innovation (e.g. mass-market software) [1]. The term “elicitation” is preferred to “capture”, to avoid the suggestion that requirements are out there to be collected.

To develop a system of quality is necessary to count with requirements of quality: complete and consistent requirements that correspond to the stakeholders´ needs. Therefore, RE is a fundamental stage in the software development process, and underestimate its importance can lead projects to failure [2-3].

RE remains problematic; missing or mistaken requirements still delay projects and cause cost overruns. No firm definition has matured for requirements elicitation in comparison to other areas of Requirements Engineering [3-6].

In chapter 2 (section 2.1), aimed at discerning the progress that has been achieved in the requirements elicitation domain, we conducted a Systematic Literature Review (SLR) [10]. As an important outcome, we identified a promising set of proposals in the field of Creativity-based Approaches for Requirements Elicitation (CAREs). Motivated by the increasing relevance of

CAREs, we performed a second SLR in this domain; the results of this research appear in section 2.2. of chapter 2.

CAREs conceive the requirements elicitation process as inherently creative, involving cycles of an incremental building followed by insight-driven re-conceptualization of the problem space [21]. Moreover, in the last decade, a line of academic works has recognized the importance of creativity in the requirements definition process. These approaches develop a vision in which requirements should be imagined and invented by stakeholders, instead of being simply "gathered" from them [4], [22]. The relevance of these works may be remarkably high in a context where innovative solutions represent a competitive advantage for companies.

CAREs browse a large spectrum of ideas for stimulating creativity and helping requirements elicitation in finding ways to take stakeholders out of conventional settings. A small subset of these techniques (e.g. scenarios and brainstorming) is well-known in the RE domain and a large set of techniques can be considered as "outsiders". The profusion of creativity techniques illustrates their growing importance for the requirement engineering community. With no doubt, the main advantage of these techniques is the number of requirements generated in a short time. Additional benefits include: the diversity of generated ideas from different stakeholders which typically differ from those generated with "standard" requirements elicitation techniques; shaking people out of tried and tested ways of thinking about requirements; improvement of communication between stakeholders and improvement of organizational climate which facilitates the creativity-based processes.

On the other hand, some drawbacks of these proposals have also been identified; in particular, it is recognized that there is a lack of proposals aimed to guide the systematic exploration of the entire Solution Ideas Space (SIS) avoiding one to wander aimlessly (a), over-visiting some parts of the space (b) and under-visiting other parts of the space (c) [17], [24-25]. Graphically:

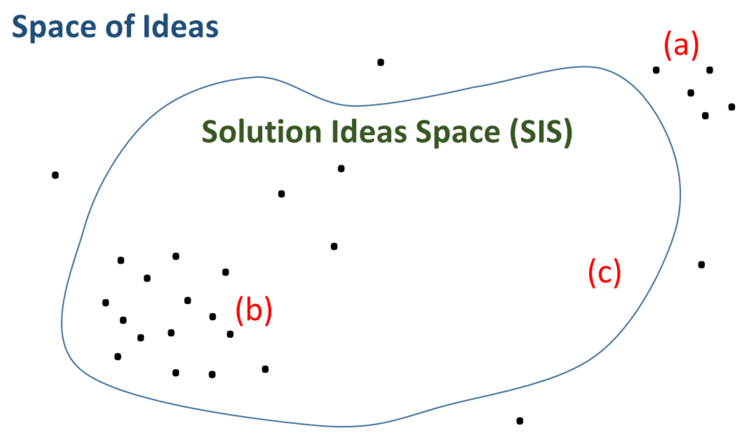


Figure 1. CAREs - issues of interest for this research.

This work is motivated by the following issues identified in the CAREs state of the art (see Figure 1):

- There is a lack of mechanisms aimed to ensure the alignment of the discovered requirements with the system purpose (Figure 1.: a)).
- The current proposals rely on the Requirements Engineer´s knowledge, which in many cases is intuitive, and based on the experience using a particular technique(s). Therefore, there is a need to bring support and guidance to the stakeholders in the systematic exploration of the entire SIS and the consequent discovery of requirements (Figure 1.: b) and c)).
- Besides, the current solutions do not facilitate the transition from the discovered requirements towards the future system specification.

In order to tackle these issues, we proposed an approach named DREQUS (Discovery of REQuirements Using Scenarios). In the sequel, the components and methods of DREQUS are described.

This chapter is organized as follows: Section 3.2.1. presents a set of mechanisms proposed by DREQUS to discover requirements in the first stage of the approach; Section 3.2.2. Identifies the type of approach which DREQUS corresponds to and its strategies; Section 3.2.3. describes Stage 1 of DREQUS corresponding to the initial discovery of requirements; and Section 3.2.4. presents

Stage 2 of DREQUS corresponding to the discovery of Use cases from the requirements discovered in Stage 1.

3.2. The Proposed Solution

We need a process that implements mechanisms contributing to the solution of the identified problems (see the previous section). To this purpose, following the *DSR methodology* and *Evidence-based Software Engineering* principles and guidelines [7-20], we explored the use of different resources that been used by related works in the field of CAREs. Considering these resources, we have designed and developed experimental versions of our proposal which were evaluated by means of case studies. In a process of continuing re-design and improving, we arrived at the first version of the proposed approach named Discovery of REQuirements Using Scenarios (DREQUS). Figure 2. illustrates the process followed to design the proposed solution.

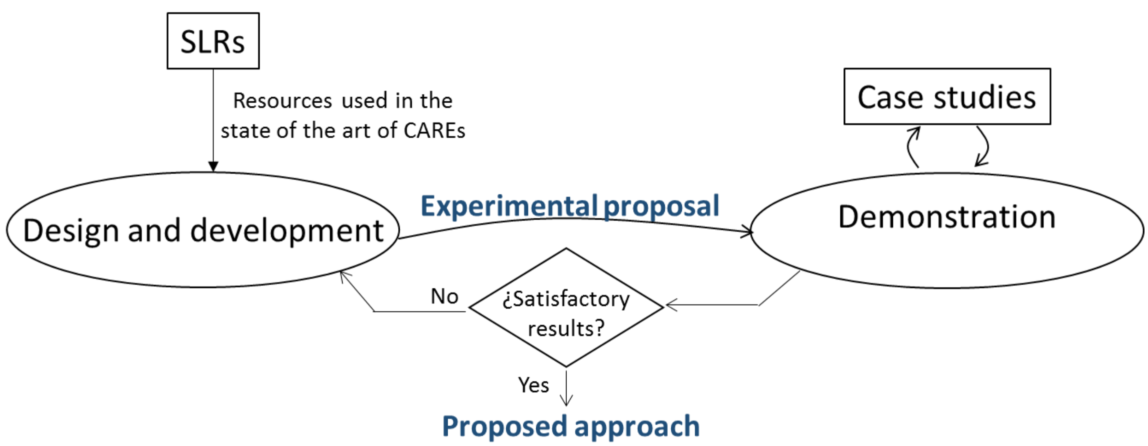


Figure 2. Process followed for the design of the proposed approach.

In the next paragraphs, we will present the components and processes of the DREQUS approach. To this aim, we will use the MAP representation system: a *map* is composed of one or more sections. A section is an aggregation of two kinds of intentions, the source, and target intentions together with a strategy. An *intention* is a goal that can be achieved by the performance of a strategy [26]. Figure 3. provides a map overview of the proposed approach.

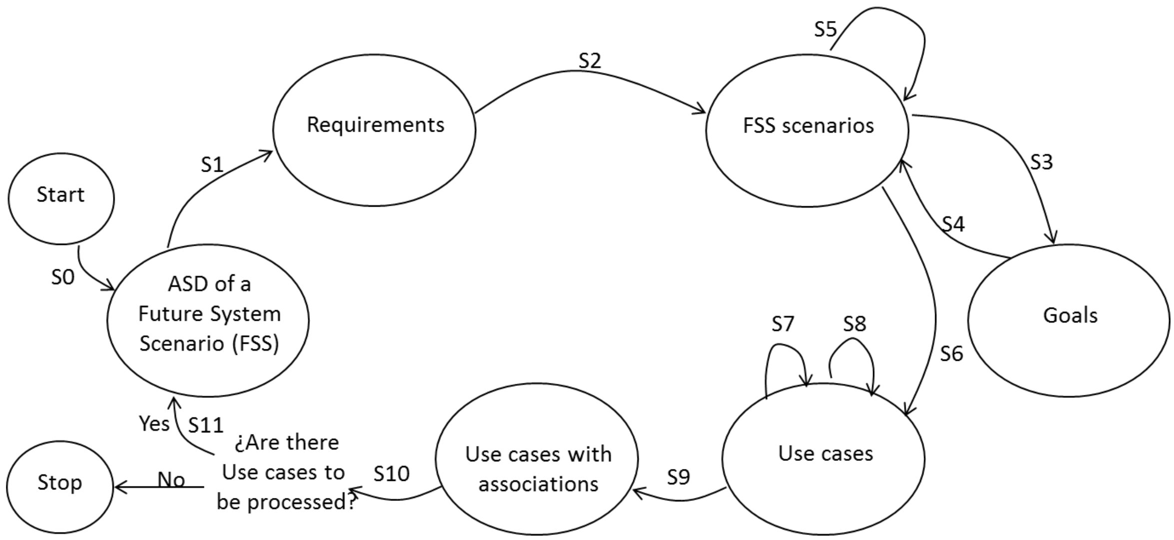


Figure 3. Overview of the DREQUS approach.

In Figure 3, S0 – S11 correspond to the next strategies:

S0: Building the Abstract Sequence Diagram (ASD) of a Future System Scenario (FSS) by envisioning its future behavior.

S1: Discovery of requirements by answering probing questions (What/Why, How and Which) based on the internal actions of the ASD, the Functional and Cognitive verbs, and the System Promise.

S2: Discovery of Future System Scenarios by requirements similarity and assessing whether the requirements occur in normal or alternative scenarios.

S3: Identify the goal related to each scenario by reasoning on the goals of the Primary Agent (using the Cockburn´s test [61]).

S4: Discovery of Future System Scenarios through merging scenarios by goals similarity.

S5: Discovery of Future System Scenarios by reasoning on failure conditions (exceptional scenarios).

S6: Identify Use Cases by establishing a relation between the scenario goal (from the normal and alternatives scenarios) and the required Use cases to meet these goals.

S7: Discovery of new Use Cases, from the discovered Use Cases, by composition: using the producing / consuming principle.

S8: Discovery of new Use Cases, from the discovered Use Cases, by composition: reasoning on missing complementary information or services.

S9: Identify associations between Use Cases by reasoning on the type of relation existing between them.

S10: Select the next Use Cases to be processed by reasoning on Use Cases and their relations.

S11: For the selected Use case: Build the Abstract Sequence Diagram of the normal scenario by envisioning its future behavior.

From Figure 3, we can identify the next two main stages:

Stage 1: this stage is aimed at the discovery of requirements by means of the strategies S0-S1.

Stage 2: it allows the discovery of Use cases applying the strategies S2-S11.

Figure 4. presents an abstract view of the DREQUS process stages.

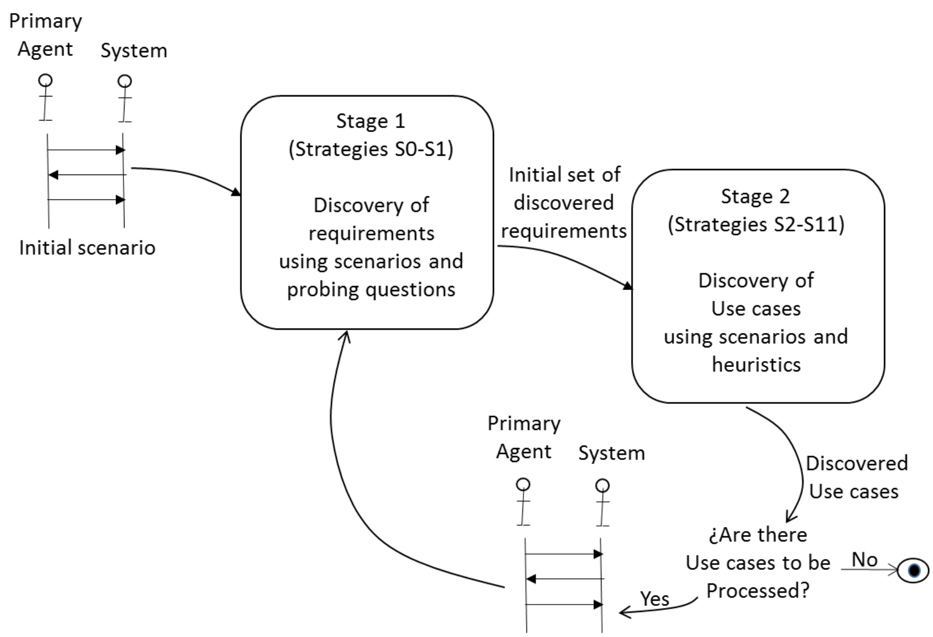


Figure 4. Stages of the DREQUS process

In the next section, we will present the mechanisms proposed for the discovery of requirements in stage 1 of DREQUS, strategies S0, and S1.

3.2.1. Establishing a set of mechanisms for the discovery of requirements in the DREQUS first stage (S0 and S1).

Regarding the first stage of the process, DREQUS proposes the use of the next mechanisms:

a. Assisting the stakeholders not to wander aimlessly

To this aim, we introduce the **"System promise"** concept. A "System promise" is an explicit declaration or assurance made to a system agent with respect to the future, stating that the system will do or refrain from some specified act, or that the system will give or bestow some specified thing [28].

To establish the *System Promise*, the **System owner** (a stakeholder, or a set of stakeholders, who has the authority to decide the functionalities and characteristics of the system in construction), with the Requirements Engineer support, must answer the question: **"*what is the main Promise that the future system must ensure to their agents?*"** The answer to this question requires the next mental and collaborative processes: ***Creation, Assessment, Negotiation,*** **and** ***Decision***.

The *System Promise* acts as a spotlight which, delimiting the SIS, focus the System owner´s reflection towards relevant requirements. A poor definition of the *System Promise* would negatively affect the relevance and quality of the discovered requirements. Figure 5. illustrates the "System promise" effect.

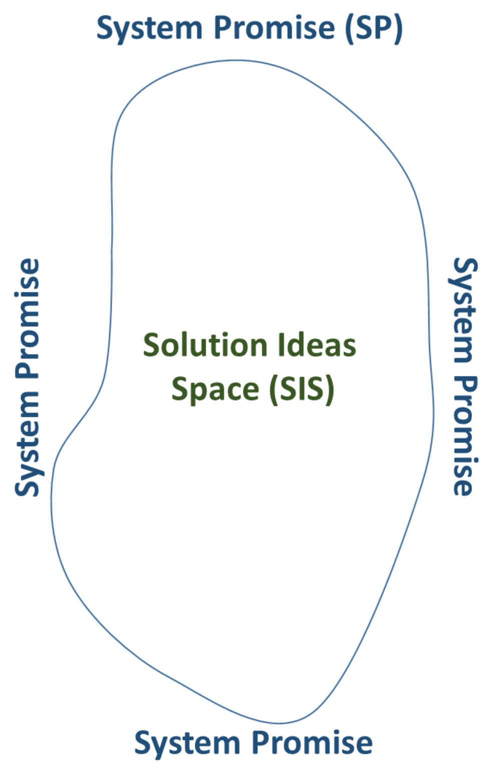


Figure 5. The “System promise” effect delimiting the SIS.

b. Assisting the stakeholders not to over-visiting some parts of the Solution Ideas Space and under-visiting other parts of this space

Based on the General Systems Theory [29], we propose to make a partition of the Solution Ideas Space (SIS) into Input, Evolution and Decision, and Output Ideas:

SIS = (Input ideas) U (Evolution and Decision ideas) U (Output ideas)

As a complement of the SIS partitioning, we propose to use a set of Functional and Cognitive Verbs to discover requirements in each partition. The rationale to use this resource, and its elements, is presented in the next paragraphs:

Remarkable works of authors like Fillmore [30] and Levin [31], among others, recognize the central role of verbs in determining the semantics of any sentence. Thanks to that, there is a wide set of verbs-based proposals for Functional (FRs) and Non-Functional Requirements (NFRs) elicitation [10]. Considering this stream of proposals, we asked ourselves: *¿What verbs, denoting FRs or NFRs, can be found in the Solution Ideas Space?* Trying to answer this question, we evidence that the verbs differ from one domain to another; e. g. verbs like "Hit" or "Kill" can usually be found in the video games domain, but they are unusual in the business domain. Taking this into account, we decided to focus our efforts on the Business Information Systems (**BIS**) domain.

Aimed to identify a set of verbs useful to discover requirements in the BIS domain, we re-formulate our initial question into the next ones: *¿Which are the verbs currently used in software specifications of BIS? and,* as a consequence of the progress of Artificial Intelligence (AI) methods, *¿What verbs will be used in future specifications of software for BIS?:*

- Regarding the question "*¿Which are the verbs currently used in software specifications of BIS?"*: Mendling et al. discuss the use of text and icons for labeling the graphical constructs in a process model [32]. The authors apply the MIT Process Handbook [33] and the Levin classes [31] in the classification of verbs used in the activity labels of the SAP Reference Model [34]. Being the SAP solution a market-leading tool in the enterprise system market, the examination of SAP process models gave us an understanding of the use of these verbs in real-life business contexts. As a result of this work, a set of twenty-five generic verbs for describing activities in business process models was synthesized. We will refer to these verbs as *Functional verbs*.

- In relation to the question "*¿What verbs will be used in future specifications of software for BIS?*": Metzler et al. propose a taxonomy of cognitive functions that supports formal functional modeling of Cognitive Technical Systems (CTSs are systems that possess similar cognitive capabilities as humans) [35]. This taxonomy is based on literature research and consists of a set of cognitive capabilities on three hierarchical levels that present hypernym relations. To the best of our knowledge, this proposal presents a wide and representative set of Cognitive functions useful for Cognitive systems development. We will name these verbs as *Cognitive verbs*.
Figure 6. presents the sets of verbs obtained from the referred works.

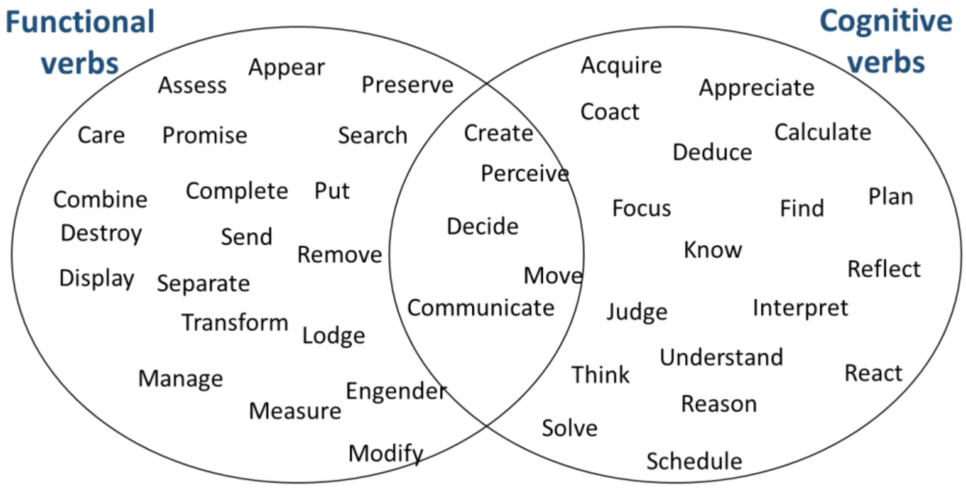


Figure 6. Initial sets of Functional and Cognitive verbs.

The use of all these 43 verbs resulted in unpractical due that each verb introduces complexity. Therefore, we faced the challenge of minimizing the number of verbs to be used without losing the initial expressiveness. The strategy followed to cope with this challenge is condensed in Figure 7.

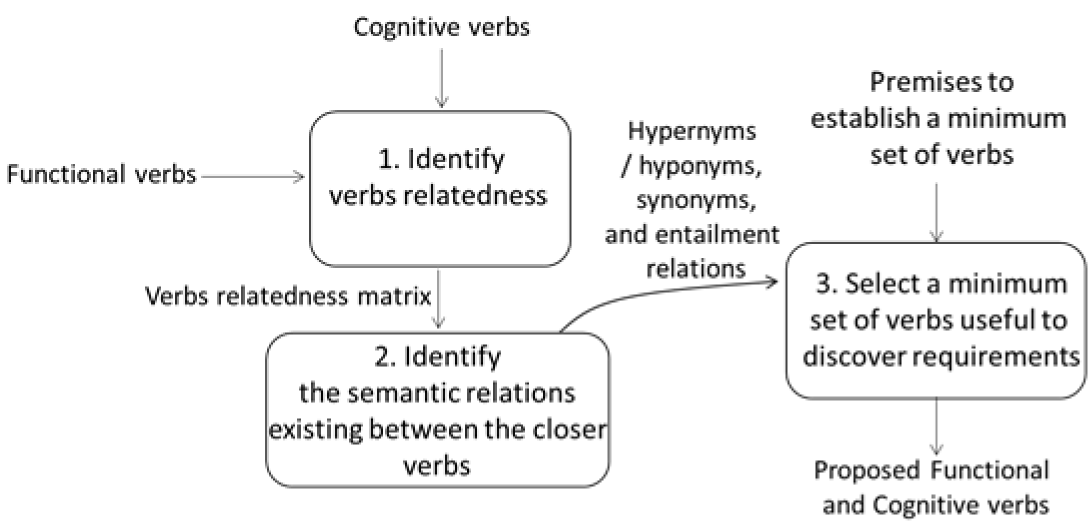


Figure 7. Process to Identify the minimum set of Functional and Cognitive verbs

In the next paragraphs, these processes are described.

1. Identify the verbs relatedness.

Aiming at having confident and objective measures of the verbs relatedness we use WordNet [36] WordNet is an on-line lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, and adjectives are organized into synonym sets (synsets), each representing one underlying lexical concept. Different relations link the synsets [37].

WordNet is a very complex graph of many thousands vertices and arcs, where vertices represent lexical units and sets of lexical units, and arcs represent lexico-semantic relations of several types (e.g. hypernym/hyponym relations) [38]. Figure 8 presents an excerpt from this graph which is visualized using the Wordvis tool [39].

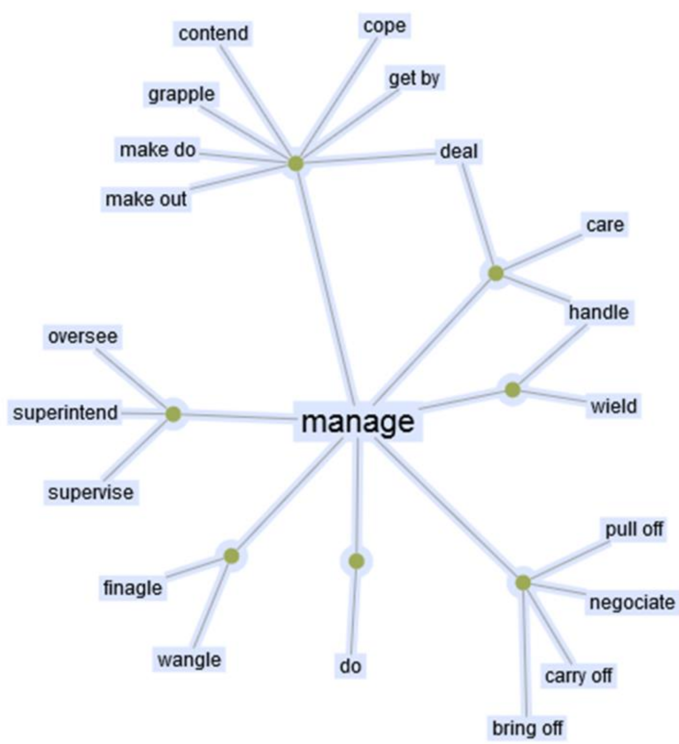


Figure 8. Example of a Wordnet graph corresponding to the “manage” verb.

Considering the WordNet graph of relations, the verbs relatedness can be measured by means of the path length existing between two verbs. The relatedness score is inversely proportional to the number of nodes along the shortest path between the synsets. The shortest possible path occurs when the two synsets are the same, in which case the length is 1. Thus, the maximum relatedness value is 1 [40].

Given the amount and complexity of verbs relations in WordNet, it is difficult to visualize or manually establish the semantic proximity between two verbs (verbs relatedness) and therefore the relations between them e.g. hypernym/hyponym relations. Considering this, we use WordNet::Similarity [41] to calculate a semantic proximity matrix which allows us to prioritize and select the verbs candidate for the search of semantic relations. Figure 9 shows an example of the semantic relatedness calculation between “acquire” and “appear” and Table 1 introduces an excerpt of the resulting verbs relatedness matrix (The complete matrix can be found in Appendices I and II).

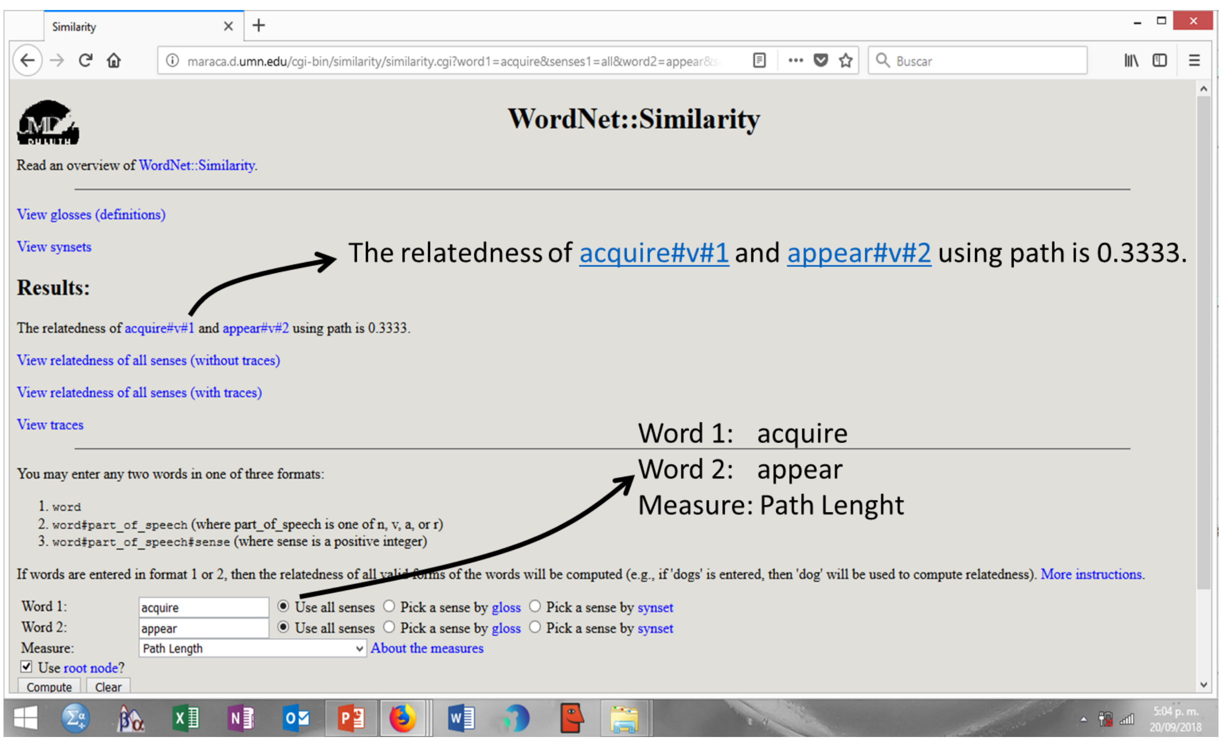


Figure 9. Semantic relatedness between “acquire” and “appear”.

Table 1. Excerpt of the verbs relatedness matrix.

	Acquire	Appear	Appreciate	Assess	Calculate	Care	Coact	Combine	Communicate	Complete	Create	Decide	Deduce
Acquire	1	0,3333	0,25	0,25	0,3333	0,25	0,25	0,25	0,2	0,3333	0,3333	0,3333	0,2
Appear	0,3333	1	0,25	0,25	0,3333	0,25	0,25	0,25	0,2	0,3333	0,3333	0,3333	0,2
Appreciate	0,25	0,25	1	0,25	0,25	0,2	0,2	0,25	0,1667	0,25	0,25	0,25	0,1667
Assess	0,25	0,25	0,25	1	0,3333	0,2	0,2	0,2	0,1667	0,25	0,25	0,25	0,2
Calculate	0,3333	0,3333	0,25	0,3333	1	0,25	0,25	0,25	0,2	0,3333	0,3333	0,3333	0,2
Care	0,25	0,25	0,2	0,2	0,25	1	0,2	0,2	0,1667	0,25	0,25	0,25	0,1667
Coact	0,25	0,25	0,2	0,2	0,25	0,2	1	0,25	0,25	0,25	0,3333	0,25	0,1667
Combine	0,25	0,25	0,25	0,2	0,25	0,2	0,25	1	0,3333	0,25	0,25	0,25	0,1667
Communicate	0,2	0,2	0,1667	0,1667	0,2	0,1667	0,25	0,3333	1	0,2	0,25	0,2	0,1429
Complete	0,3333	0,3333	0,25	0,25	0,3333	0,25	0,25	0,25	0,2	1	0,3333	0,3333	0,2
Create	0,3333	0,3333	0,25	0,25	0,3333	0,25	0,3333	0,25	0,25	0,3333	1	0,3333	0,2
Decide	0,3333	0,3333	0,25	0,25	0,3333	0,25	0,25	0,25	0,2	0,3333	0,3333	1	0,2

2. Identify the semantic relations existing between the closer verbs.

Using the Verbs Relatedness Matrix obtained in the previous step, the strategy followed to minimize the initial set of verbs is as follows:

a. Identify closer verbs: given a pivot verb: identify its closer verbs using the Verbs Relatedness Matrix; e.g. given the verb "Remove" its closer verbs are, Table 2:

Table 2. Verbs semantically closer to "Remove"

	Remove
Acquire	0,3333
Calculate	0,3333
Complete	0,3333
Create	0,3333
Decide	0,3333
Destroy	0,3333
Find	0,3333
Know	0,3333
Manage	0,3333
Modify	0,3333
Move	0,3333
Perceive	0,3333
Reflect	0,3333
Search	0,3333
Separate	0,3333

b. Identify the semantic relations existing between closer verbs: in WordNet, verbs are linked in a hierarchy according to relations like those presented in Table 3:

Table 3. Semantic relations between verbs considered to minimize the verbs sets.

Semantic relation	**Definition**	**Visual representation used in this work**
Synonymy	Two expressions are synonymous in a linguistic context C if the substitution of one for the other in C does not alter the truth value. For example, the substitution of *plank (X)* for *board (Y)* will seldom alter truth values in carpentry contexts, although there are other contexts of *board* where that substitution would be totally inappropriate [37].	X ←*Synonym*→ Y
Co-hyponymy	Verbs that have the same hypernym [42]. Eg. If Y is hypernym of X and Y is hypernym of Z, then X and Z are co-hyponyms in relation to Y.	Y ◁– X, Z
Hypernymy / Hyponymy	[verb Y is a hypernym of the verb X if the activity X is a (kind of) Y (to *perceive* is a hypernym of to *listen*] and hyponym [verb Y is a hyponym of the verb X if the activity Y is doing X in some manner (to *lisp* is a hyponym of to *talk*)] [43].	Y ◁– X
Entailment	In logic, entailment, or strict implication, is properly defined for propositions; a proposition *P* entails a proposition *Q* if and only if there is no conceivable state of affairs that could make *P* true and *Q* false. Entailment is a semantic relation because it involves reference to the states of affairs that *P* and *Q* represent. The term will be generalized here to refer to the relation between two verbs *V1* and *V2* that holds when the sentence *Someone V1* logically entails the sentence *Someone V2*; this use of entailment can be called lexical entailment. Thus, for example, *snore* lexically entails *sleep* because the sentence *He is snoring* entails *He is sleeping*; the second sentence necessarily holds if the the first one does [37].	V1 –*Entails*→ V2

We used Wnconnect [44] to have an objective result of whether exists or not a semantic relation (synonymy, co-hyponymy hypernymy/hyponymy, or entailment) between pairs of verbs. Figure 10 illustrates the result of this process for the verbs "Remove" and "Move". Table 4 summarizes the semantic relations identified for "Remove" and its closer verbs.

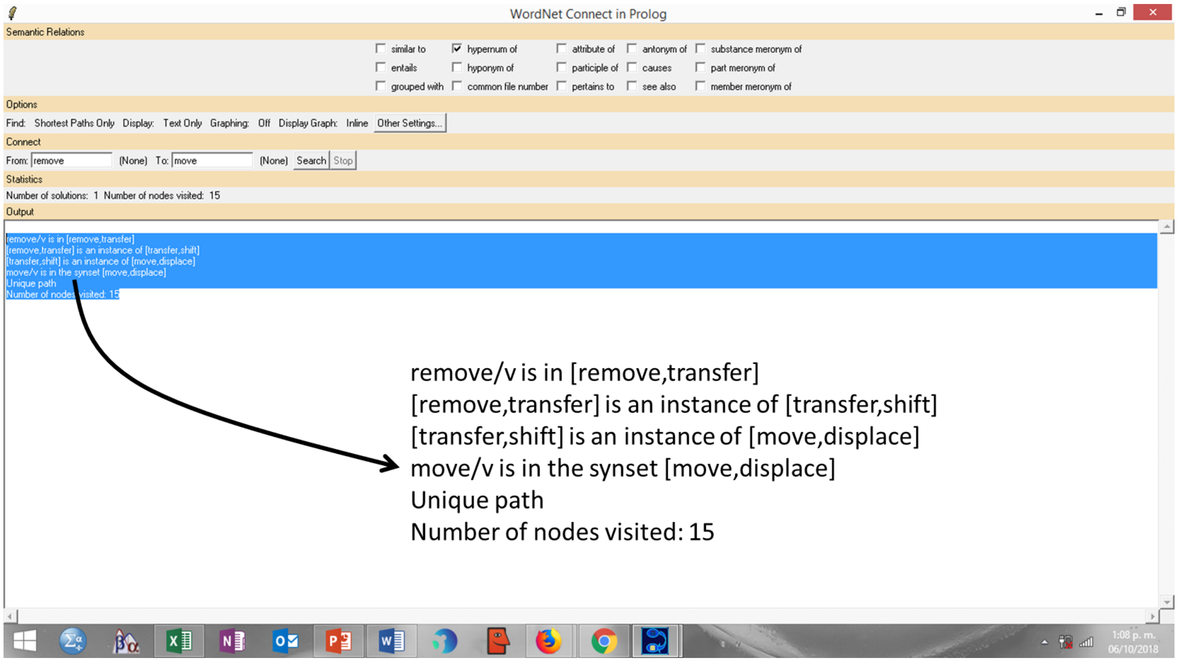


Figure 10. Hypernym/Hyponym relation existing between "Move" and "Remove".

Table 4. Semantic relations identified for "Remove" and its closer verbs.

Pivot verb	**Closer verb**	**Semantic relation**		
		Hypernymy/Hyponymy	**Synonymy**	**Entailment**
Remove	Acquire	No path	No path	No path
	Calculate	No path	No path	No path
	Complete	No path	No path	No path
	Create	No path	No path	No path
	Decide	No path	No path	No path
	Destroy	No path	No path	No path
	Find	No path	No path	No path
	Know	No path	No path	No path
	Manage	No path	No path	No path
	Modify	No path	No path	No path
	Move	**Remove is an instance of Move.**	No path	No path
	Perceive	No path	No path	No path
	Reflect	No path	No path	No path
	Search	No path	No path	No path
	Separate	No path	No path	No path

The semantic relation between “Move” and “Remove” is graphically represented in Figure 11.

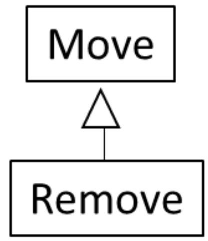


Figure 11. Graphical representation of the hypernym/hyponym realtion between “Remove” (hyponym) and “Move” (hypernym)

The systematic execution of this process leaded us to the next set of semantic relations between the initial set of verbs, Figure 12:

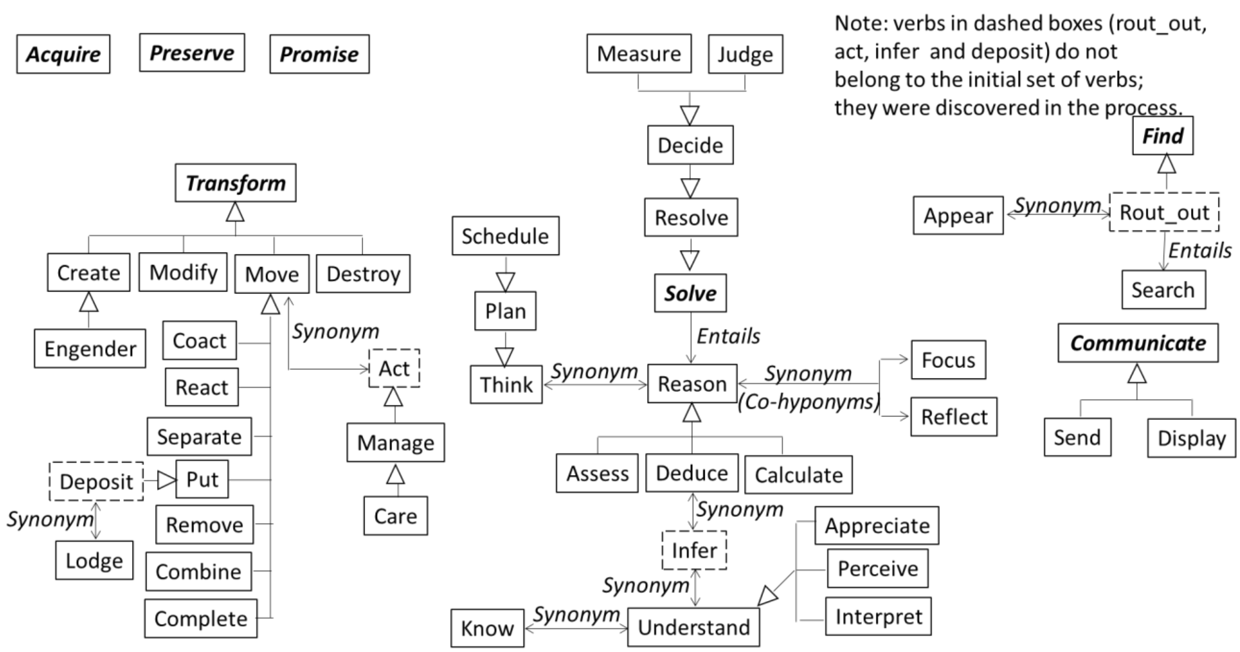


Figure 12. Semantic relations existing between the initial set of Business and Cognitive verbs

3. Select a minimum set of verbs useful to discover requirements.

Considering the semantic relations identified in the previous step, we propose to leave aside some verbs according to the next premises.

Premise 1 (P1). Synonyms: given the *Business context* and two synonyms in this context, by definition, it is possible to leave aside one of these verbs.

Premise 2 (P2). Co-hyponyms: these verbs are candidates to synonyms [42].

Premise 3 (P3). Hypernyms: they allow us to leave aside their hyponyms. This due that hypernyms represent a family of concepts of the same type which expands the possibilities of reasoning on alternatives in comparison to hyponyms which are more precise concepts.

Premise 4 (P4). Entailment: by definition, in this kind of relation the entailed verb can be left aside.

E.g., considering that "Move" is "hypernym of "Remove", using P3, we can leave aside "Remove".

The systematic application of the premises over the obtained verbs relations let us prune the initial set of verbs as indicated in Table 5:

Applying	We conserve	And prescind from	and its related verbs
No premise	Promise	Not apply	Not apply
No premise	Acquire	Not apply	Not apply
No premise	Preserve	Not apply	Not apply
P3	Transform	Create	Engender (by P3)
P3	Transform	Modify	Not apply
P3	Transform	Move	Coact (by P3) React (by P3), Separate (by P3), Put (by P3), Deposit (by P3), and Lodge (by P1), Remove (by P3), Combine (by P3), |Complete (by P3), Act (by P1), Manage (by P3), Care (by P3)
P3	Transform	Destroy	Not apply
P3	Solve	Resolve	Decide (by P3), Measure (by P3), Judge (by P3)
P4	Solve	Reason	Focus and Reflect (by P2 and P1), Think (by P1), Plan (by P3), Schedule (by P3), Assess (by P3), Deduce (by P3), Infer (by P1), Understand (by P1), Know (by P1), Appreciate (by P3), Perceive (by P3), Interpret (by P3), Calculate (by P3)
P3	Find	Rout_out	Appear (by P1), Search (by P4)
P3	Communicate	Send	Not apply
P3	Communicate	Display	Not apply

Table 5. Set of verbs obtained after applying the premises.

The final set of proposed verbs is: Acquire, Promise, Preserve, Transform, Solve, Find and Communicate; we have noticed that these verbs can be grouped in the following categories: Input, Evolution and Decision and Output as is presented in Table 6.

Table 6. Unified set of Functional and Cognitive verbs

Input verb	**Verbs of evolution and decision**	**Output verb**
Acquire	Promise, Transform (create, modify, move and destroy), Solve (resolve, reason), Preserve and Find	Communicate

These verbs constitute a generic and basic set of verbs that are used by DREQUS to discover the requirements for a new business information system. We do not claim that this set of verbs is enough for all domains and systems; instead, we encourage the Requirements Engineers to extend this proposal adding their verbs according to their particular needs. Each of these verbs guides the reflection and imagination of the *System owner* towards different and necessary actions (Input, Evolution and Decision, and output actions) involved in any process.

Table 7. summarizes the mechanisms proposed by DREQUS for the SIS exploration.

Table 7. Mechanisms that facilitate the SIS exploration.

a. "System promise": delimits the SIS.
b. SIS partitions and their Functional and Cognitive verbs: facilitate the exploration of the SIS by means of sub-spaces (input, evolution and decision and output) and actions suggested by the Functional and Cognitive verbs.

The next section introduces the process followed to select the type of approach to be designed and its inputs.

3.2.2. Identifying the type of approach to be designed and its processes (strategies)

To establish the type of approach to be designed, we consider, as alternatives, the types of proposals existing in the CAREs state of the art [11]. After analyzing these works we decided to explore the use of *a Representation-oriented* design. We discarded using psychological and cognitive-oriented approaches due that they are not well known in the RE domain. We also discarded using organizational-oriented approaches considering that they are expensive and difficult to put in practice.

Regarding *Representation-oriented* approaches, we discarded using *Topic maps* due that this type of representation is an outsider in the RE field. Therefore, we experimented with the next well-known and widely used models: Goal models, Storyboard, and Scenarios. As a result of multiple cases studies we concluded:

- Goal models: the goal concept is fuzzy and the discovery of goals is not an easy task (conclusion in coincidence with [45]). Besides, the operations over goals like refinement and composition are complex and led us to models that were hard to manage. For these reasons, we ruled out this alternative.

- Storyboard: the unstructured nature of the stories introduced complexity which was difficult to deal with. We discarded this alternative for this reason.

- Scenarios: scenarios are narrative and specific descriptions of current or future processes including actions and interactions between the users and the system. Scenarios represent paths of possible behavior through a use case [46]. After trying with different forms of scenarios representation, like text, Business Process Models (BPMs), and Sequence Diagrams, we observed that although all these types of models allowed us to describe the systems functionalities with sufficiency, we obtained the better cost-benefit ratio using Sequence Diagrams (the cost-benefit ratio was estimated in terms of the effort to describe the scenario and its easiness to be automated). In fact, we propose to use Abstract Sequence Diagrams (ASD) which resulted in more convenience than Sequence Diagrams.

An ASD establishes the actions between the *Primary Agent* (i.e. an agent who has a goal that requires the assistance of the system [27]) and the system; these actions use resources (input or output data) to accomplish their tasks and are classified as follows:

- System Interactions: system interactions establish the communication between the *Primary Agent* and the system with the aim to accomplish a *Primary agent´s* goal.

- System internal actions: the *system internal level* focuses on what the system needs to perform the interactions selected at the system interaction level. The 'what' is expressed in terms of system internal actions that involve system objects but may require external objects such as other scenarios or systems [45]. Figure 13 describes the ASD elements and their relations and Figure 14 shows the graphical representation of an ASD.

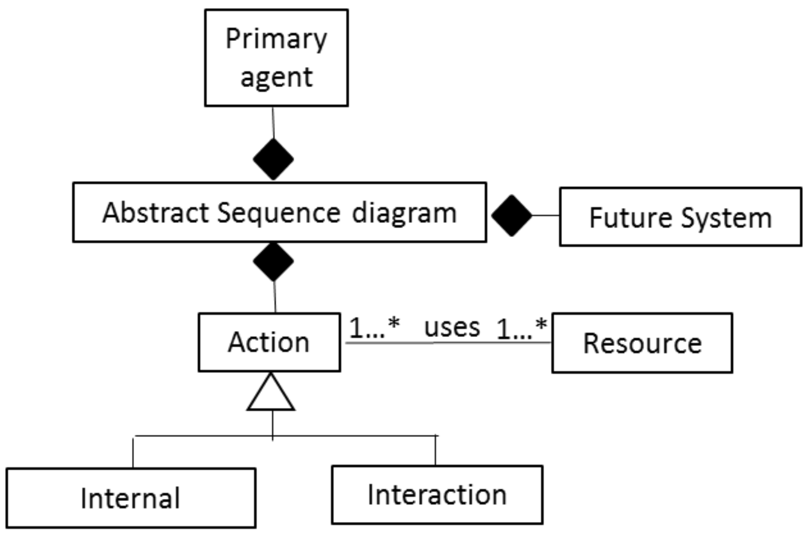


Figure 13. ASD elements and their relations.

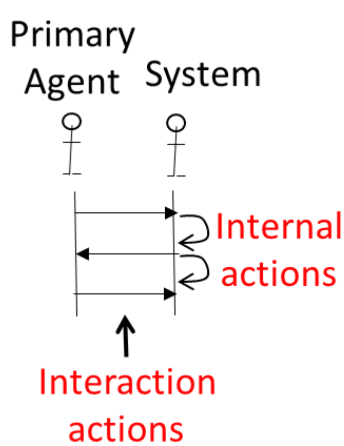


Figure 14. Graphical representation of an ASD.

Using only *internal actions* we obtained similar results than those achieved using both interaction and internal actions (with considerably less effort). This result is in agreement with Rolland and Salinesi: “The system internal level defines the software requirements to meet the system requirements” [45].

Considering the internal actions of ASDs, we experimented with two options of scenarios: Current System Scenarios (CSS) (also known in the RE field as “As-Is” scenarios) and Future System Scenarios (FSS) (also known as “To-Be” scenarios). As a result, we obtained that the CSS tended to facilitate the elicitation of obvious requirements; on the other hand, the FSS tended to allow the discovery of not evident requirements. In consequence, we based our proposal on the use of Future System Scenarios (FSS) represented as Abstract Sequence Diagrams (ASD).

3.2.3. Stage 1 of DREQUS (S0-S1): Initial Discovery of Requirements

This stage is aimed to tackle the next CAREs issues:

- There is a lack of mechanisms aimed to ensure the alignment of the discovered requirements with the system purpose (Figure 1.: a)).

- The current proposals rely on the Requirements Engineer´s knowledge, which in many cases is intuitive, and based on the experience using a particular technique(s). Therefore, there is a need to bring support and guidance to the stakeholders in the systematic exploration of the entire SIS and the consequent discovery of requirements (Figure 1.: b) and c)).

In the previous sections, we have identified and described the mechanisms used by the proposed approach to deal with the identified problems of CAREs. In the next paragraphs, using these resources, we will detail the strategies followed by DREQUS to assist the discovery of requirements of a future system. Figure 15. shows the map corresponding to the first stage of the proposal.

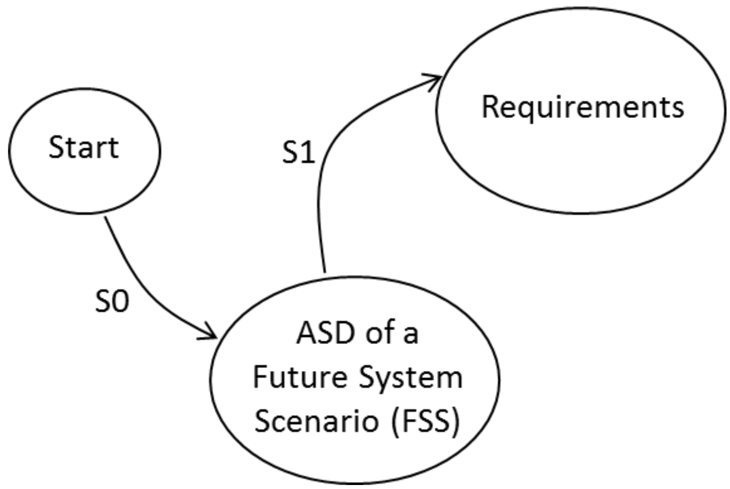


Figure 15. DREQUS – Stage 1.

The strategies S0 and S1 are described in the sequel:

3.2.3.1. S0: Building the Abstract Sequence Diagram (ASD) of a Future System Scenario (FSS) by envisioning its future behavior.

The initial Future System Scenario must be *imagined* by the System owner and it must correspond to his vision of the normal behavior of one of the future system functionalities. This task must be achieved, with the Requirements Engineer assistance, by means of the next steps:

1. Identify the Primary Agent of the scenario.
2. Envision the Primary Agent 's goal to be reached with the scenario assistance.

Considering the normal scenario:

3. Identify the scenario interactions between the Primary Agent and the system.
4. Identify the data required by each interaction in order to be successfully executed.
5. Identify the system internal actions.
6. Identify the data required by each system internal action in order to be successfully executed.
7. Build the Abstract Sequence Diagram taking into account the elements discovered in the previous steps.

8. Execute S1.

Comment: The Requirements Engineer represents the normal scenario of the functionality by means of an ASD with the next characteristics: - The diagram must represent the interaction of only two agents: *The Primary Agent* and the *Future System*; - The diagram must remark the *System Interaction Actions* and the *System Internal Actions*.

This ASD is used as an input for the next strategy as is presented in the next section.

3.2.3.2. S1: Discovery of requirements by answering probing questions (What/Why, How and Which) based on the internal actions of the ASD, the Functional and Cognitive verbs, and the System Promise.

We propose to stimulate the stakeholders´ reflection over the new system requirements through a list of probing questions that are dynamically instantiated from the Internal actions, the Functional and Cognitive verbs, and the System promise. These questions appear in Figure 16.

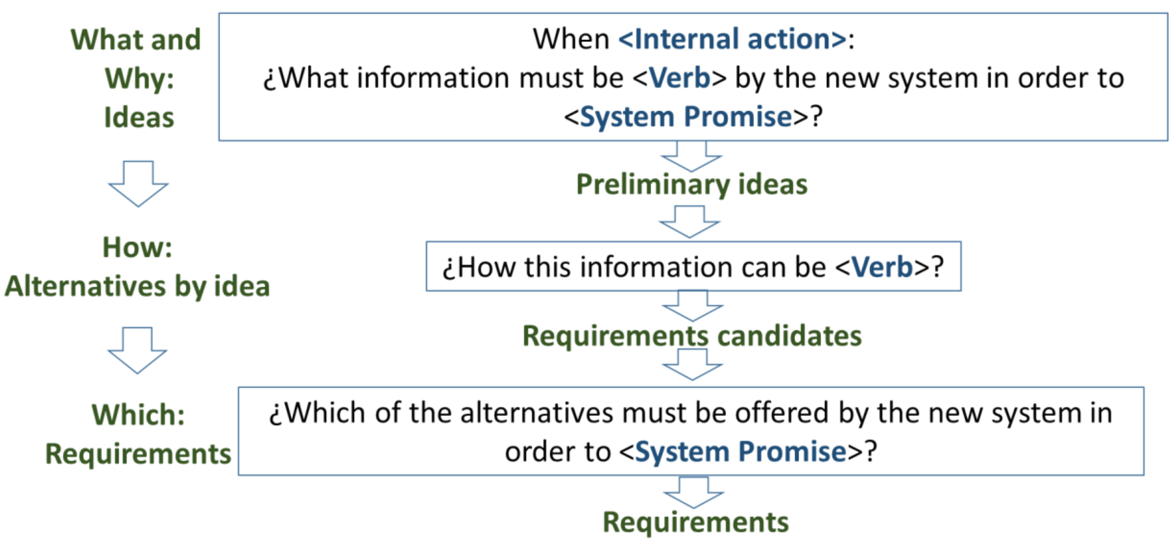


Figure 16. List of probing questions to be instantiated.

- "*What*" questions: this type of question is aimed to *imagine* the required functionalities, problems, challenges to be solved, and the characteristics of the future system. The consideration of the "System promise" provides the alignment of the answers with the system intention (Why).

- "*How*" questions: for each answer given to a "*What*" question, the "*How*" question challenges the *System owner* to imagine the best *alternatives* which allow the functionality accomplishment and its required characteristics.

- "*Which*" questions: The *System owner* has to *Assess, Negotiate, and Decide* about the alternatives to be implemented in the new system. As indicated by Aurum and Wohlin [47], this requires to assess each alternative in aspects like the cost-benefit, alignment with the organizational strategy (*System Promise*), added value, among others.

These *questions* foster the *System owner's* creativity thanks to the combination of the *System internal actions, the Functional and Cognitive verbs, and the System Promise*. The answers to these types of questions involve the next mental and collaborative processes: Create, Assess, Negotiate, and Decide. Figure 17. provides a complete view of the proposal components.

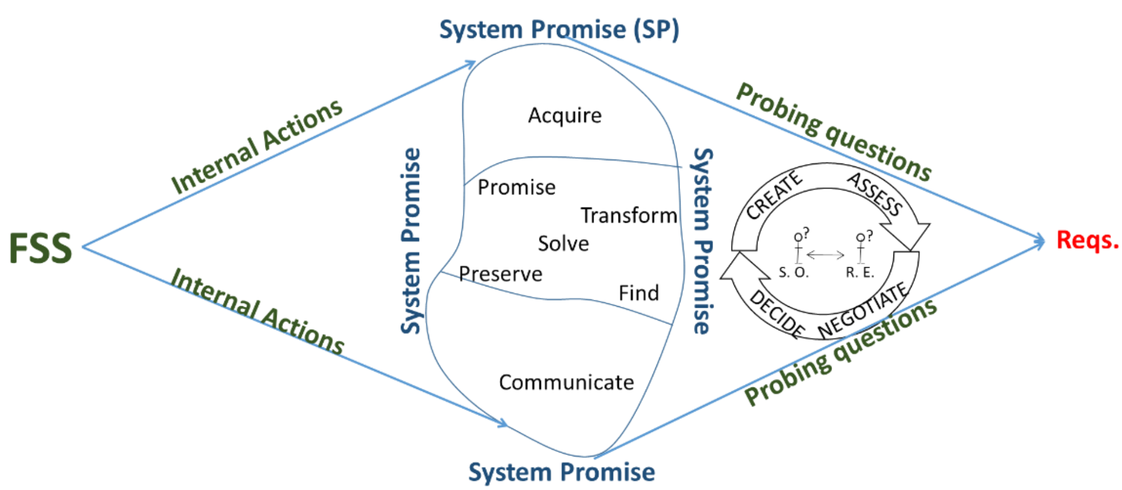


Figure 17. General view of the initial phase of DREQUS.

The *System Owner (SO)* must solve the instantiated questions with the *Requirements Engineer* (RE) assistance (if it is required). The thoughtful answers to the generated questions constitute a set of requirements that are part of the requirements specification for the future system. These requirements will be used in stage 2 as the input to discover new Use Cases and requirements (see Figure 3: strategies S2 – S11).

3.2.3.3. Illustrating the DREQUS Stage 1 (S0-S1) through an application case

Problem statement: several hotels have made an agreement to build one reservation system for all hotels *("Booking rooms system*"). These are their initial recommendations (wishes and needs):

The system shall treat customer requests as automatically as possible. Any request is made by either an already known customer or by a fresh new prospect. The customer is identified by his Id. whereas personal data (name, address, and telephone number) shall be captured for a prospect. The request is set in general terms: the requested hotel category, the number of rooms needed and the period that the person requests. A request relates to one single period but one or several rooms. Obviously, the system shall memorize information about hotels such as name, code, category, rooms, ground & mail address, and telephone number (end of the problem statement).

Hereafter, we will count with the DREQUS assistance, to discover the requirements for the new system, executing the first stage of the process.

S0: Building the Abstract Sequence Diagram (ASD) of a Future System Scenario (FSS) by envisioning its future behavior.

Following the guidelines offered by DREQUS, The *System owner* (investors), arrives at the next elements of the FSS (the Requirements Engineer encourages the System owner to be imaginative and also brings support to these tasks):

1. Identify the Primary Agent of the scenario.
Result: the primary agent of the FSS is the *Customer / Prospect*.

2. Envision the Primary Agent's goal to be reached with the scenario assistance.
Result: the goal of the Customer / Prospect is "Reserve a room".
Considering the normal scenario of "Reserve a room":

3. Identify the scenario interactions between the Primary Agent and the system.

Result: the scenario interactions are: Request for rooms, Inform rooms, Confirm reservation, and Inform reservations.

4. Identify the data required by each interaction in order to be successfully executed.

Result: the data required by each interaction are:

- Request for rooms: period, number of rooms, and maximum cost.
- Inform rooms: rooms, number of beds, amenities, and costs.
- Confirm reservation: Yes or no.
- Inform reservations: reservation

5. Identify the system internal actions.

Result: the system internal actions are: Consult rooms availability, and Reserve rooms.

6. Identify the data required by each system internal action in order to be successfully executed.

Result: the data required by each internal action are:

- Consult rooms availability: Period, Number of rooms, and Maximum cost.
- Reserve rooms: Yes or no, Period, Number of rooms, and Maximum cost.

7. Build the Abstract Sequence Diagram taking into account the elements discovered in the previous steps.

Result: considering the information obtained in the previous steps, the Requirements Engineer builds the next ASD:

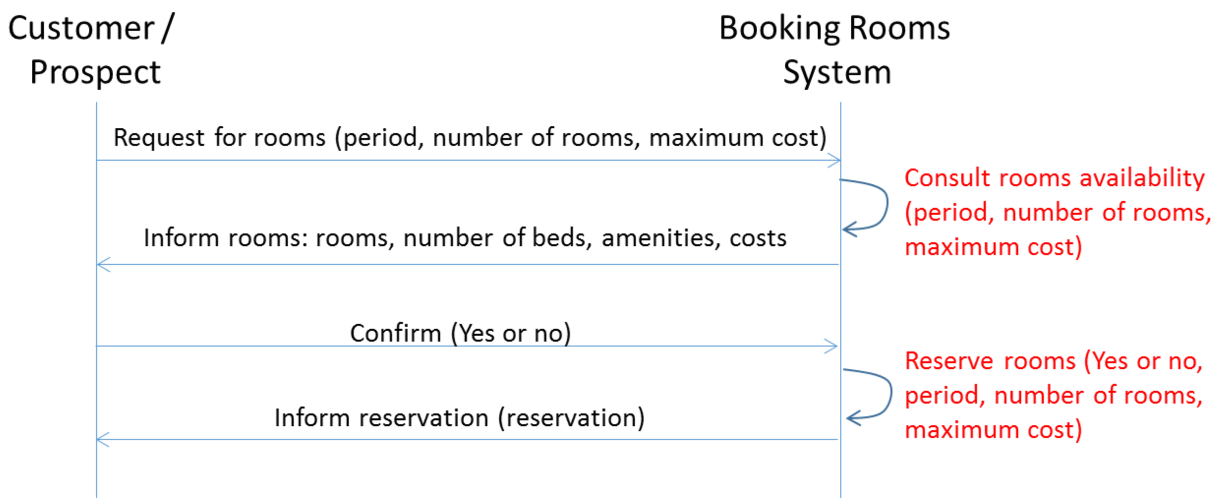


Figure 18. ASD corresponding to the “Reserve rooms” FSS

8. Execute S1

S1: Discovery of requirements by answering probing questions (What/Why, How and Which) based on the internal actions of the ASD, the Functional and Cognitive verbs, and the System Promise.

In the first place, to discover the *System Promise*, the System owner (investors) must answer the question: “*what is the main Promise that the Future system must ensure to their agents?*”. After a collaborative and conscious process which includes the phases of creation, assessment, negotiation, and decision, the System owner discovers the System promise: “the system must secure the customer´s loyalty.” Taking into account this *System promise*, the next sections present the discovery of requirements for each interaction action: Consult rooms availability, and Reserve rooms.

Discovery of requirements from the internal action 1: Consult rooms availability

Let us consider the following elements corresponding to the "Reserve rooms" scenario (Figure 18):

- System owner: investors.

- Future System Scenario: Reserve rooms.

- Internal action: Consult rooms availability.

- Functional and Cognitive verbs = {Acquire, Preserve, Transform (Create, Modify, Move and Destroy), Solve (resolve, reason), Find, and Communicate}

- System promise: "the system must secure the customer´s loyalty".

These elements allow us to instantiate the questions that lead the System owner reflection towards the discovery of the Future system requirements. The System owner can select the order of the verbs that will generate the questions. The instantiated questions and the System owner´s answers are:

a. Acquire (Create – Assess – Negotiate – Decide)

When consulting rooms availability:
¿What information must be acquired in order to "secure the customer´s loyalty"?

Answer (investors):
- We have to know the customer, his preferences and loyalty in order to offer customized alternatives.

⇨

Acquire the period, number of rooms, and maximum cost is not enough. Besides this information it is necessary to acquire:

- The customer´s Id, preferred hotels and rooms.

¿How this information can be acquired?

Asking the customer

Asking the prospect

Automatically from the system

¿Which of these alternatives must be offered by the new system in order to "secure the customer´s loyalty"?

The new system shall acquire:

- From customers and prospects: period, number of rooms, and maximum cost.
- From customers: the customer´s Id.
- From the system: the customer´s preferences and loyalty.

Note: the investors decided don´t ask the prospects their preferences.

b. Preserve

When consulting rooms availability:
¿What information must be preserved in order to "secure the customer´s loyalty"?

Answer:

1. With the aim to assess the customer or prospect´s needs, we have to preserve the entered data: period, number of rooms, mininal cost and customer´s Id.
2. The requested consult (for the requester).

¿How this information can be preserved?

1. Alternatives:
- Everytime is made a consult.
- Everytime the system manager needs a backup.

2. Alternatives:
- In a .doc format.
- In a .xls format.
- In a .pdf format
- In a .jpg format.

¿Which of these alternatives must be offered by the new system in order to "secure the customer´s loyalty"?

The new system shall :

- Preserve the consult data everytime they are entered.
- Preserve the consult data everytime the system manager needs a backup.
- Preserve the customer and prospect´s consult in .pdf format.

c. Transform

When consulting rooms availability:
¿What information must be transformed (created, modified, moved or destroyed) in order to "secure the customer´s loyalty"?

Answer:
- The consult of rooms availability.

¿How this information can be transformed?

- Creating a consult of rooms availability
- Modifying a consult of rooms availability
- Destroying a consult of rooms availability

¿Which of these alternatives must be offered by the new system in order to "secure the customer´s loyalty"?

The new system shall :
- Create consults of rooms availability.
- Modify consults of rooms availability.
- Destroy consults of rooms availability.

d. Solve

When consulting rooms availability:
¿What unconformities or challenges must be solved (resolved, reasoned) in order to "secure the customer´s loyalty"?

Answer:
1. The processing of consults is slow.
2. The requester make mistakes when introducing periods.
3. The requester cannot consult known hotels and rooms.
4. There are no availability of rooms in the customer´s desired dates.
5. The information does not match with the customer´s preferences.

¿How these unconformities can be solved?

1. Alternative:
- The processing of the request must be instantaneous (NFR).

2. Alternative:
- Dates could be typed or selected by means of calendars.

3. Alternatives:
- Hotels and rooms could be typed or selected from images or lists.

4. Alternatives:
- Suggest alternative rooms and hotels.
- Suggest the use of a pending list.

5. Alternative:
- Customized the information according to customer´s preferences.

¿Which of these alternatives must be offered by the new system in order to "secure the customer´s loyalty"?

The new system shall:
- Process the requests instantly (NFR).
- Select the dates from calendars.
- Select hotels and rooms from images.

The new system shall:
- Suggest alternative rooms and hotels.
- Suggest the use of a pending list.
- Customized the information according to customer´s preferences.

e. Find

When consulting rooms availability:
¿What information must be found in order to “secure the customer´s loyalty”?

Answer:

1. We must find the best alternatives of hotels , rooms , services , and costs according to the customer´s preferences and prospect needs..
2. We must find discounts according to customers´ loyalty.

⇨ ¿How these information can be found?

1. Alternatives:
- Finding the availability of specific (selected by the customer) hotels, rooms and services.
- Finding alternatives of hotels, rooms and services. acoording to dates, number of rooms, and maximum cost.

2. Alternatives:
- Finding discounts according to visit frequency.
- Finding discounts according to the customer´s points registered in a loyalty program.

⇩

¿Which of these alternatives must be offered by the new system in order to “secure the customer´s loyalty”?

The new system shall:
- Find the availability of specific (selected by the customer) hotels, rooms and services.
- Find the alternatives of hotels, rooms and services. acoording to dates, number of rooms, and maximum cost.
- Find discounts according to the customer´s points registered in a loyalty program.

f. Communicate

When consulting rooms availability:
¿What information must be communicated in order to “secure the customer´s loyalty”?

Answer:
We must send all the information required by the requester to make a reservation. Besides the rooms, number of beds, amenities and costs.; the customer need to know alternatives of hotels and rooms according to his preferences and discounts according to his loyalty.

⇨ ¿How this information can be communicated?

In a customized manner for customers.

In a non-customized manner for prospects.

⇩

¿Which of these alternatives must be offered by the new system in order to “secure the customer´s loyalty”?

The new system shall :
- To customers: communicate customized alternatives of hotels, rooms, services and loyalty discounts.
- To prospects: communicate information of rooms, number of beds, amenities, services, costs.

The requirements discovered from the internal action "*Consult rooms availability*" are:

The new system shall: 1. Acquire from customers and prospects: period, number of rooms, and maximum cost. 2. Acquire from customers: the customer´s Id. 3. Acquire from the system: the customer´s preferences and loyalty. 4. Preserve the consult data everytime they are entered. 5. Preserve the customer and prospect´s consult in .pdf format. 6. Create consults of rooms availability. 7. Modify consults of rooms availability. 8. Destroy consults of rooms availability. 9. Process the requests instantly (NFR). 10. Select the dates from calendars. 11. Select hotels and rooms from images. 12- Suggest alternative rooms and hotels. 13. Suggest the use of a pending list. 14. Customized the information according to customer´s preferences. 15. Find the availability of specific (selected by the customer) hotels, rooms and services. 16. Find the alternatives of hotels, rooms and services. acoording to dates, number of rooms, and maximum cost. 17. Find discounts according to the customer´s points registered in a loyalty program. 18. To customers: communicate customized alternatives of hotels, rooms, services, and loyalty discounts. 19. To prospects: communicate information of rooms, number of beds, amenities, services, costs.

The process continues instantiating and answering the questions corresponding to the internal action 2, Reserve rooms, as is presented in the next section.

Discovery of requirements from the internal action 2: Reserve rooms

The scenario elements to be considered are:

- System owner: investors.

- Future System Scenario: Reserve rooms.

- Internal action: Reserve rooms.

- Functional and Cognitive verbs = {Acquire, Preserve, Transform (Create, Modify, Move and Destroy), Solve (resolve, reason), Find, and Communicate}

- System promise: "the system must secure the customer´s loyalty".

Considering these elements, the instantiated questions and the System owner´s answers are:

a. Acquire (Create – Assess – Negotiate – Decide)

When reserving rooms:
¿What information must be acquired in order to “secure the customer´s loyalty”?

Answer (the Investors must: Create – Assess – Negotiate - Decide):

1. The selected options of rooms and services (confirm (Yes)), and the customer or prospect´s credit card number.
2. Customer´s Id., period, number of rooms, and services.

¿How these information can be acquired?

1. Alternative:
- From the customer or prospect.
- The credit card number could be obtained from the customer´s loyalty system.

2. Alternative:
- From “consult rooms availability”.

¿Which of these alternatives must be offered by the new system in order to “secure the customer´s loyalty”?

The new system shall:

- Acquire from customers and prospects: the selected options of rooms and services (confirm yes), and the credit card number
- Acquire from “consult rooms availability”: Customer´s Id., period, number of rooms, and services.

b. Preserve

When reserving rooms:
¿What information must be preserved in order to “secure the customer´s loyalty”?

Answer:
1. The reservation: date and hour of the reservation , period, hotels, rooms, services, costs, terms and conditions.

¿How these information can be preserved?

In a .doc format	In a .xls format	In a .pdf format	In a .jpg format

¿Which of these alternatives must be offered by the new system in order to “secure the customer´s loyalty”?

The new system shall:

- Preserve the reservation in .pdf format.

c. Transform

When reserving rooms:
¿What information must be transformed (created, modified, moved, destroyed) in order to "secure the customer´s loyalty"?

Answer:
1. The reservation.

¿How these information can be transformed?

Create a reservation in a .doc format	Create a reservation in a .xls formatc	Create a reservation in a .pdf format	Create a reservation in a .jpg format	Destroy a reservation

¿Which of these alternatives must be offered by the new system in order to "secure the customer´s loyalty"?

The new system shall :
- Create a reservation in a .pdf format.
- Destroy a reservation.

d. Solve

When reserving rooms:
¿What unconformities must be solved (resolved, reasoned) in order to "secure the customer´s loyalty"?

Answer:
1. The reservation process is slow.

¿How these unconformities can be solved?

Alternative:
The reservation process could be instantly.

The new system shall :
- Perform the reservations instantly.

e. Find

When reserving rooms:
¿What information must be found in order to "secure the customer´s loyalty"?

Not apply; see the internal action: consult rooms availability

f. Communicate

When reserving rooms:
¿What information must be communicated in order to "secure the customer´s loyalty"?

Answer:
The reservation: period, hotels, rooms, services, costs, terms and conditions.

⇨ ¿How this information could be communicated?

Online

E-mail

⇩

¿Which of these alternatives must be offered by the new system in order to "secure the customer´s loyalty"?

The new system shall :
- Communicate the reservation online.
- Communicate the reservation by e-mail.

The requirements discovered from the internal action "*Reserve rooms*" are:

The new system shall:

20. Acquire from customers and prospects: the selected options of rooms and services (confirm yes), and the credit card number
21. Acquire from "consult rooms availability": Customer´s Id., period, number of rooms, and services.
22. Preserve the reservation in .pdf format.
23. Create a reservation in a .pdf format.
24. Destroy a reservation.
25. Perform the reservations instantly.
26. Communicate the reservation online.
27. Communicate the reservation by e-mail.

These requirements are the input of the next DREQUS stage as is presented in the following section.

3.2.4. Stage 2 of DREQUS (S2-S11): Discovery of Use Cases from the Requirements discovered in Stage 1.

This stage is aimed to tackle the following CAREs issue:

- The current solutions do not facilitate the transition from the discovered requirements towards the future system specification.

The guidelines and rules proposed to achieve each of the strategies (S2-S11) are detailed in the next paragraphs:

3.2.4.1. S2: Discovery of Future System Scenarios by requirements similarity and assessing whether the requirements occur in normal or alternative scenarios.

a. Cluster the discovered requirements (from stage 1) according to their similarity.

According to experience, similar requirements tend to belong to the same scenario [48]. Preparing the requirements, by grouping them into similar clusters, will facilitate the execution of the posterior steps.

In order to perform the requirements clustering, we have evaluated the next tools: String similarity [49], SEMILAR: a semantic similarity toolkit [50], LSA (Latent Semantic Analysis) [51], ReqSimile [52] and SenseClusters [53]. Finally, ReqSimile offered the best results. This can be explained due to the nature of this tool, which is specially designed to deal with requirements (e.g. generally, requirements are expressed as short texts). ReqSimile uses the *Cosine Similarity* which in the domain of Requirements Engineering has shown a good performance compared with other techniques [54]. The ReqSimile process is as follows: this tool uses linguistic engineering to calculate the similarity between requirements by using lexical similarity as a way of approximating semantic similarity. On a lexical level, the authors consider a requirement as a sequence of words. It processes the sequence to drop words that have a purely grammatical role and to remove affixes and other lexical components not needed for comparison purposes. Then ReqSimile uses the *Cosine similarity measure* as it does not depend on the relative size of the input. It also considers the weight of the words in the requirements (the number of times each word occurs). The result from calculating the Cosine measure is a number between 0 (no similarity) and 1 (total similarity) that indicate the similarity between the requirements based on the words they have in common. For details on the calculation steps see [48].

On the other hand, several researchers have shown the adequacy of the agglomerative hierarchical clustering algorithm, in different problems requiring the orthogonal requirements clustering [55], [56-58]. Based on these works, we selected the agglomerative hierarchical clustering algorithm; we have also selected the Ward method because it works well in finding tightly bound or compact clusters [59].

The guideline followed to cluster the discovered requirements is:

Guideline Id.: CSR.
Input: discovered requirements.
Output: requirements clustered by similarity.
Process:
Step 1. Calculate the matrix of similarities using ReqSimile.
Step 2. Cluster the requirements using the matrix of similarities and the agglomerative hierarchical algorithm (Ward method).
Step 3. Make manual adjustments (if it is necessary). Due that requirements texts could present issues that affect the results (e.g. completeness, anaphora, etc.), if it is required, the Requirements Engineer can validate the automated results and make the necessary adjustments.

b. Cluster requirements into "normal scenario" and "alternative scenarios"

Hereafter, each rule is introduced using the following template <Goal, Body, Comment>. The *goal* is expressed in the following notation [60]:

So (source): corresponds to the input or inputs required to accomplish the goal.
Res (result): establishes the expected results with the goal achievement.
Man (manner): defines the way the goal is achieved.

The *body* is expressed as a sequence of steps to be followed when applying the rule. The *comment* explains the rule.

Rule Id.: CS1.
Goal: Cluster (from requirements clustered by similarity)$_{So}$ (requirements clustered into requirements of the "normal scenario" and requirements of the "alternative scenarios")$_{Res}$ (reasoning on the normal and alternative behaviors)$_{Man}$
Body: step 1: create the "Normal Scenario" (NS) (initially this scenario is empty).
Step 2: take the next unprocessed cluster that has the least number of requirements (if there are several take any of them).
Step 3:

```
While unprocessed requirements exist in the cluster
        Take the next unprocessed requirement
        If the requirement is alternative then
                Create an alternative scenario
                Move the requirement to the alternative scenario (AS)
        Else
                Move the requirement to the NS
        End-if
End-while
```

Step 4: Go to step 2 while there are unprocessed clusters.
Step 5. Eliminate the empty clusters.

Comment: The Requirements Engineer must evaluate if each requirement is alternative (or normal), which tends to be a tedious work as the number of requirements increases. The proposed algorithm facilitates the Requirements Engineer´s work, taking advantage of the following considerations:

- Each input cluster consists of similar requirements.
- Clusters of bigger size mainly consist of normal requirements which, according to experience, are easier to discover than the alternative ones.
- Alternative requirements tend to exist in smaller clusters.

Thus, the algorithm leads and focuses the Requirements Engineer´s attention in the assessment of requirements, from smaller clusters to bigger ones, asking if the requirement is alternative (or normal) due that is more likely to find alternative requirements in smaller clusters.

3.2.4.2. S3: Identify the goal related to each scenario by reasoning on the goals of the Primary Agent (using the Cockburn´s test).

Rule Id.: G1
Goal: Discover (from normal and alternative scenarios)$_{So}$ (scenarios goals)$_{Res}$ (reasoning on the goals of the primary agents)$_{Man}$
Body: step 1: for each normal and alternative scenarios: identify the primary agent of the scenario.
Step 2: for each primary agent: identify the high level goal of the primary agent using the test: “Does the primary agent´s job performance depend on how many of these you do in a day?”

Comment: the guiding rule G1 allows the Requirements Engineer to discover the scenario goals. For each scenario, the Requirements Engineer must identify the goal that the Primary Agent (i.e. one having a goal requiring the assistance of the system [61]) wants to accomplish through the scenario. The discovered goals must satisfy the test proposed by Cockburn [61]: “Does the primary agent´s job performance depend on how many of these you do in a day?”

3.2.4.3. S4: Discovery of Future System Scenarios through merging scenarios by goals similarity.

Guideline Id.: MSG

Input: goals and their corresponding scenarios.

Output: scenarios merged by goal.

Process:

Assess and decide

If exist scenarios that share the same goal then

- Merge them into one scenario.

End-if

Answer:

- Scenarios merged by goal.

Note: the scenarios are separated according to the conditions encountered, and grouped together as they have the same goal [61].

3.2.4.4. S5: Discovery of Future System Scenarios by reasoning on failure conditions (exceptional scenarios).

Rule Id.: E1

Goal: Discover (from normal and alternative scenarios)$_{So}$ (exceptional scenarios)$_{Res}$ (reasoning on failure conditions)$_{Man}$

Body: step 1: for each requirement of the normal and alternative scenarios: identify possible conditions of failure.

Step 2: for each condition of failure: set a scenario to deal with the failure.

Comment: the guiding rule E1 aims to discover exceptional scenarios from the requirements of the normal and alternative scenarios. For each requirement, the Requirements Engineer must look for all possible conditions that prevent the successful accomplishment of the requirement. For each failure condition the Requirements Engineer must set a scenario to deal with it.

3.2.4.5. S6: Identify Use Cases by establishing a relation between the scenario goal (from the normal and alternatives scenarios) and the required Use Cases (UC) to meet these goals.

Rule id.: UC1

Goal: Discover (from goals of the normal and alternative scenarios)$_{So}$ (use cases)$_{Res}$ (establishing a relation between the scenario goals and the required use cases to accomplish these goals)$_{Man}$

Body: step 1: for each goal of the normal and alternative scenarios: identify the use cases required to accomplish the goal taking into account the next guideline:

- "In general, define one use case for each user goal. Name the use case similar to the user goal. For example, Goal: Process a sale; Use Case: *Process Sale*." [62].

Comment: the guiding rule UC1 aims to identify the use case(s) associated to each scenario goal. For each scenario goal, the Requirements Engineer must identify the use case(s) required of the system in order to accomplish the scenario goal. The discovered Use Cases must satisfy the guideline proposed by Larman [62].

3.2.4.6. S7: Discovery of new Use Cases, from the discovered Use Cases, by composition: using the producing / consuming principle.

Rule id.: C1

Goal: Discover (from an existing use case)$_{So}$ (new use cases)$_{Res}$ (using the producing / consuming principle)$_{Man}$

Body: step 1: for each requirement of the normal and alternative scenarios: identify the resources (information) necessary for the requirement accomplishment.

Step 2: construct interactions pairs (produce, consume) for each identified resource.

Step 3: suggest a new scenario for every incomplete pair (i.e. in which either the produce interaction or the consume interaction is missing).

Step 4: ask to select the relevant scenarios and name each of them with a goal. The discovered goals must satisfy the test: "Does the primary agent´s job performance depend on how many of these you do in a day?"

Step 5: identify the new use cases related to each discovered goal.

Comment: the guiding rule C1 aids the Requirements Engineer to discover the missing goals of the primary agent (and their respective use cases) associated to the production / consumption of the information necessary to accomplish each requirement [60].

3.2.4.7. S8: Discovery of new Use Cases, from the discovered Use Cases, by composition: reasoning on missing complementary information or services.

Rule id.: C2
Goal: Discover (from an existing use case)$_{So}$ (new use cases)$_{Res}$ (reasoning on missing complementary information or services)$_{Man}$
Body: step 1: for each requirement of the normal and alternative scenarios: identify the missing complementary information or services related to the requirement.
Step 2: suggest a new scenario for every missing complementary information or services and name them with a goal. The discovered goals must satisfy the test: "Does the primary agent´s job performance depend on how many of these you do in a day?"
Step 3: identify the new use cases related to each discovered goal.

Comment: the guiding rule C2 aids the Requirements Engineer to discover the missing goals of the primary agent, and their respective use cases, related to missing complementary information or services required to accomplish each requirement.

3.2.4.8. S9: Identify associations between Use Cases by reasoning on the type of relation existing between them.

Rule id.: AS1
Goal: Discover (from use cases)$_{So}$ (use cases associations)$_{Res}$ (reasoning on the type of relation existing between the use cases)$_{Man}$
Body: for each use case: step1: if the use case is necessary to accomplish another one then

- Establish an <includes> association between the use cases.

Step 2: if the use case extends the behavior of another one then

- Establish an <extends> association between the use cases.

Step 3: if the use case "is a kind of" another one then

- Establish a <generalization / specialization> association between the use cases.

Comment: the guiding rule AS1 assists the Requirements Engineer in the discovery of associations of the types <includes>, <extends> and <generalization / specialization> between the use cases. For each use case, the rule asks to the Requirements Engineer if there are relations of these types between the use case and the other ones.

3.2.4.9. S10: Select the next Use Cases to be processed by reasoning on Use Cases and their relations.

Guideline id.: SUC

Input: new Use cases.

Output: new Use cases to be processed.

Process:

Assess and decide (for each new Use case)

If the (Use case is included by other Use case) or (the Use case is an extension of other Use case) then

- The Use case shouldn't be processed.

Else

If the Use case analysis can lead to the discovery of new scenarios of the system then

- The Use case must be processed: execute S11with the selected Use case.

End-if

End-if

If there are no Use cases to be processed then

- Stop.

End-if

3.2.4.10. S11: For the selected Use case: Build the Abstract Sequence Diagram of the normal scenario by envisioning its future behavior.

Guideline id.: BASD

Input: new use case to be processed.

Output:

Abstract Sequence Diagram corresponding to the normal scenario of the use case to be processed.

Process:

Create, Assess and decide

The System owner, with Requirements Engineer assistance, must envisions the behavior of the functionality in consideration by means of the next steps:

Step 1. Identify the Primary Agent of the normal scenario and his goal.

Considering the normal scenario:

Step 2. Identify the scenario interactions between the Primary Agent and the system.

Step 3. Identify the data required by each interaction in order to be successfully executed.

Step 4. Identify the system internal actions.

Step 5. Identify the data required by each system internal action in order to be successfully executed.

Step 6. Build the Abstract Sequence Diagram taking into account the elements discovered in the previous steps.

Step 7. Start the DREQUS process using this new sequence diagram (execute S1 (stage 1)).

Comment: The Requirements Engineer represents the normal scenario of the functionality by means of an *Abstract Sequence Diagram* with the next characteristics: - The sequence diagram must represent the interaction of only two agents: *The Primary Agent* and the *Future System*; - The sequence diagram must remark the *System Interaction Actions* and the *System Internal Actions*.

In the next section, the stage 2 is illustrated through the “Booking rooms” application case.

3.2.4.11. Illustrating the DREQUS Stage 2 by means of the "Booking rooms system" application case.

Considering the requirements discovered in stage 1; the System owner must execute the next steps with the Requirements Engineer support.

S2: Discovery of Future System Scenarios by requirements similarity and assessing whether the requirements occur in normal or alternative scenarios.

a. Cluster the discovered requirements (from stage 1) according to their similarity.

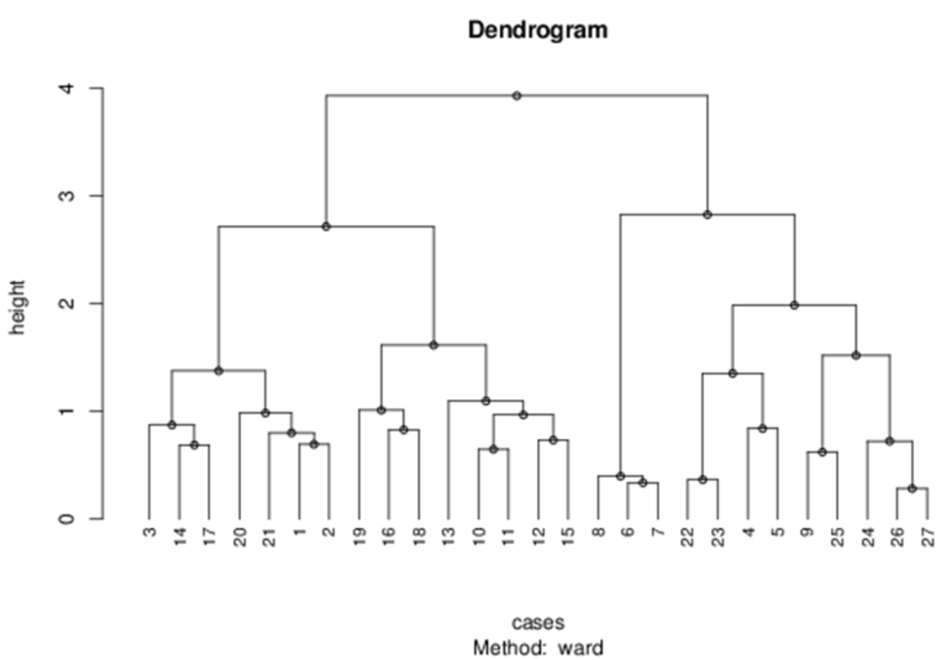
Input: elicited requirements.
Output: requirements clustered by similarity.
Process:
Calculate the matrix of similarities using ReqSimile

In order to calculate the requirements similarity using ReqSimile, we name the set of requirements as RSA (Requirements specification A) and RSB (Requirements Specification B); where (RSA1 = RSB1, RSA2 = RSB2,…, RSA27 = RSB27). After input RSA and RSB; we ask the tool for the similarities between the requirements. The next matrix presents the requirements similarities obtained with ReqSimile:

	RSB1	RSB2	RSB3	RSB4	RSB5	RSB6	RSB7	RSB8	RSB9	RSB10	RSB11	RSB12	RSB13	RSB14	RSB15	RSB16	RSB17	RSB18	RSB19	RSB20	RSB21	RSB22	RSB23	RSB24	RSB25	RSB26	RSB27
RSA1	1,000	0,625	0,589	0,267	0,375	0,316	0,316	0,316	0,289	0,316	0,316	0,316	0,289	0,401	0,316	0,471	0,375	0,335	0,471	0,510	0,589	0,289	0,289	0,354	0,289	0,316	0,316
RSA2	0,625	1,000	0,707	0,267	0,500	0,316	0,316	0,316	0,289	0,316	0,316	0,316	0,289	0,535	0,316	0,236	0,500	0,447	0,236	0,510	0,707	0,289	0,289	0,354	0,289	0,316	0,316
RSA3	0,589	0,707	1,000	0,378	0,471	0,447	0,447	0,447	0,408	0,447	0,447	0,447	0,408	0,630	0,447	0,333	0,589	0,527	0,333	0,481	0,556	0,408	0,408	0,500	0,408	0,447	0,447
RSA4	0,267	0,267	0,378	1,000	0,535	0,507	0,507	0,507	0,309	0,338	0,338	0,338	0,309	0,286	0,338	0,252	0,267	0,239	0,252	0,218	0,378	0,463	0,309	0,378	0,309	0,338	0,338
RSA5	0,375	0,500	0,471	0,535	1,000	0,474	0,474	0,474	0,289	0,316	0,316	0,316	0,289	0,401	0,316	0,236	0,375	0,335	0,236	0,306	0,471	0,722	0,577	0,354	0,289	0,316	0,316
RSA6	0,316	0,316	0,447	0,507	0,474	1,000	0,800	0,800	0,365	0,400	0,400	0,400	0,365	0,338	0,600	0,298	0,316	0,283	0,298	0,258	0,596	0,365	0,548	0,447	0,365	0,400	0,400
RSA7	0,316	0,316	0,447	0,507	0,474	0,800	1,000	0,800	0,365	0,400	0,400	0,400	0,365	0,338	0,600	0,298	0,316	0,283	0,298	0,258	0,596	0,365	0,365	0,447	0,365	0,400	0,400
RSA8	0,316	0,316	0,447	0,507	0,474	0,800	0,800	1,000	0,365	0,400	0,400	0,400	0,365	0,338	0,600	0,298	0,316	0,283	0,298	0,258	0,596	0,365	0,365	0,671	0,365	0,400	0,400
RSA9	0,289	0,289	0,408	0,309	0,289	0,365	0,365	0,365	1,000	0,365	0,365	0,365	0,333	0,309	0,365	0,272	0,289	0,258	0,272	0,236	0,272	0,333	0,333	0,408	0,667	0,365	0,365
RSA10	0,316	0,316	0,447	0,338	0,316	0,400	0,400	0,400	0,365	1,000	0,600	0,400	0,365	0,338	0,400	0,447	0,316	0,283	0,298	0,387	0,298	0,365	0,365	0,447	0,365	0,400	0,400
RSA11	0,316	0,316	0,447	0,338	0,316	0,400	0,400	0,400	0,365	0,600	1,000	0,600	0,365	0,338	0,600	0,447	0,316	0,424	0,298	0,387	0,298	0,365	0,365	0,447	0,365	0,400	0,400
RSA12	0,316	0,316	0,447	0,338	0,316	0,400	0,400	0,400	0,365	0,400	0,600	1,000	0,548	0,338	0,600	0,596	0,316	0,566	0,298	0,258	0,298	0,365	0,365	0,447	0,365	0,400	0,400
RSA13	0,289	0,289	0,408	0,309	0,289	0,365	0,365	0,365	0,333	0,365	0,365	0,548	1,000	0,309	0,365	0,272	0,289	0,258	0,272	0,236	0,272	0,333	0,333	0,408	0,333	0,365	0,365
RSA14	0,401	0,535	0,630	0,286	0,401	0,338	0,338	0,338	0,309	0,338	0,338	0,338	0,309	1,000	0,338	0,378	0,535	0,478	0,378	0,327	0,378	0,309	0,309	0,378	0,309	0,338	0,338
RSA15	0,316	0,316	0,447	0,338	0,316	0,600	0,600	0,600	0,365	0,400	0,600	0,600	0,365	0,338	1,000	0,447	0,316	0,424	0,298	0,258	0,447	0,365	0,365	0,447	0,365	0,400	0,400
RSA16	0,471	0,236	0,333	0,252	0,236	0,298	0,298	0,298	0,272	0,447	0,447	0,596	0,272	0,378	0,447	1,000	0,354	0,527	0,444	0,289	0,333	0,272	0,272	0,333	0,272	0,298	0,298
RSA17	0,375	0,500	0,589	0,267	0,375	0,316	0,316	0,316	0,289	0,316	0,316	0,316	0,289	0,535	0,316	0,354	1,000	0,559	0,236	0,306	0,354	0,289	0,289	0,354	0,289	0,316	0,316
RSA18	0,335	0,447	0,527	0,239	0,335	0,283	0,283	0,283	0,258	0,283	0,424	0,566	0,258	0,478	0,424	0,527	0,559	1,000	0,422	0,365	0,422	0,258	0,258	0,316	0,258	0,424	0,424
RSA19	0,471	0,236	0,333	0,252	0,236	0,298	0,298	0,298	0,272	0,298	0,298	0,298	0,272	0,378	0,298	0,444	0,236	0,422	1,000	0,385	0,333	0,272	0,272	0,333	0,272	0,447	0,447
RSA20	0,510	0,510	0,481	0,218	0,306	0,258	0,258	0,258	0,236	0,387	0,387	0,258	0,236	0,327	0,258	0,289	0,306	0,365	0,385	1,000	0,481	0,236	0,236	0,289	0,236	0,258	0,258
RSA21	0,589	0,707	0,556	0,378	0,471	0,596	0,596	0,596	0,272	0,298	0,298	0,298	0,272	0,378	0,447	0,333	0,354	0,422	0,333	0,481	1,000	0,272	0,272	0,333	0,272	0,298	0,298
RSA22	0,289	0,289	0,408	0,463	0,722	0,365	0,365	0,365	0,333	0,365	0,365	0,365	0,333	0,309	0,365	0,272	0,289	0,258	0,272	0,236	0,272	1,000	0,833	0,612	0,500	0,548	0,548
RSA23	0,289	0,289	0,408	0,309	0,577	0,548	0,365	0,365	0,333	0,365	0,365	0,365	0,333	0,309	0,365	0,272	0,289	0,258	0,272	0,236	0,272	0,833	1,000	0,612	0,500	0,548	0,548
RSA24	0,354	0,354	0,500	0,378	0,354	0,447	0,447	0,671	0,408	0,447	0,447	0,447	0,408	0,378	0,447	0,333	0,354	0,316	0,333	0,289	0,333	0,612	0,612	1,000	0,612	0,671	0,671
RSA25	0,289	0,289	0,408	0,309	0,289	0,365	0,365	0,365	0,667	0,365	0,365	0,365	0,333	0,309	0,365	0,272	0,289	0,258	0,272	0,236	0,272	0,500	0,500	0,612	1,000	0,548	0,548
RSA26	0,316	0,316	0,447	0,338	0,316	0,400	0,400	0,400	0,365	0,400	0,400	0,400	0,365	0,338	0,400	0,298	0,316	0,424	0,447	0,258	0,298	0,548	0,548	0,671	0,548	1,000	0,800
RSA27	0,316	0,316	0,447	0,338	0,316	0,400	0,400	0,400	0,365	0,400	0,400	0,400	0,365	0,338	0,400	0,298	0,316	0,424	0,447	0,258	0,298	0,548	0,548	0,671	0,548	0,800	1,000

b. Cluster the requirements using the matrix of similarities and the agglomerative hierarchical algorithm (Ward method).

We use SPSS [63] to execute the agglomerative hierarchical clustering (Ward method) to the requirements of the *"Booking Rooms System"* discovered in the previous section. We have arrived to the results condensed on next dendrogram:



The main clusters are:

C1: 1, 2, 3, 14, 17, 20, 21

C2: 10, 11, 12, 13, 15, 16, 18, 19

C3: 6, 7, 8

C4: 4, 5, 9, 22, 23, 24, 25, 26, 27

1.1.3. Make manual adjustments (if necessary)

Once revised the clusters, the Requirements Engineer (RE) makes the following changes:

- Move the requirements 4 and 5 from C4 to C3.

The final clusters are:

C1: 1, 2, 3, 14, 17, 20, 21

C2: 10, 11, 12, 13, 15, 16, 18, 19

C3: 4, 5, 6, 7, 8

C4: 9, 22, 23, 24, 25, 26, 27

Cluster the elicited requirements into "normal scenario" and "alternative scenarios".

Input: requirements clustered by similarity.

Output: requirements clustered into requirements of the "normal scenario" and requirements of the "alternative scenarios".

Process:

Assess and decide

¿Which requirements must be performed in the normal scenario? Which requirements must be performed in alternative scenarios?

Answer:

Rule applied: CS1
Requirements of the “Normal Scenario (NS)”
In the normal scenario the following requirements must be performed: 1. Acquire from customers and prospects: period, number of rooms, and maximum cost. 2. Acquire from customers: the customer´s Id. 3. Acquire from the system: the customer´s preferences and loyalty. 4. Preserve the consult data every time they are entered. 6. Create consults of rooms availability. 9. Process the requests instantly (NFR). 10. Select the dates from calendars. 14. Customize the information according to customer´s preferences. 16. Find alternatives of hotels, rooms and services according to dates, number of rooms, and maximum cost. 17. Find discounts according to the customer´s points registered in a loyalty program. 18. To customers: send customized alternatives of hotels, rooms, services, and loyalty discounts. 19. To prospects: send information of rooms, number of beds, amenities, services, costs. 20. Acquire from customers and prospects: the selected options of rooms and services (confirm yes), and the credit card number 21. Acquire from “consult rooms availability”: Customer´s Id., period, number of rooms, and services. 22. Preserve the reservation in .pdf format. 23. Create a reservation in a .pdf format. 25. Perform the reservations instantly. 26. Send the reservation online. 27. Send the reservation by e-mail.

Requirements of "Alternative Scenarios (AS)"	
Extracted from cluster	In alternative scenarios the following requirements must be performed:
C3	AS 1: 5. Preserve the customer and prospect´s consult in .pdf format.
	AS 2: 7. Modify consults of rooms availability.
	AS 3: 8. Destroy consults of rooms availability.
C4	AS 4: 24. Destroy a reservation.
C2	AS 5: 11. Select hotels and rooms from images.
	AS 6: 12. Suggest alternative rooms and hotels.
	AS 7: 13. Suggest the use of a pending list to reserve rooms.
	AS 8: 15. Find the availability of specific (selected by the customer) hotels, rooms and services.

S3: Identify the goal related to each scenario by reasoning on the goals of the Primary Agent (using the Cockburn´s test [61]).

Identify the goals related to each scenario

Input: normal and alternative scenarios.

Output: goals and their corresponding scenarios.

Process:

Assess and decide

¿What goal corresponds to each scenario?

Answer:

Normal scenario			
Requirement	Rule applied	Primary agent	Scenario goal
All the requirements of the normal	G1	Customer or	The customer or prospect wants to

scenario.		prospect	reserve a room.
Alternative scenarios			
Alternative scenario	Rule applied	Primary agent	Scenario goal
5. Preserve the customer and prospect´s consult in .pdf format.	G1	Customer or prospect	The customer or prospect wants to reserve a room with the facility of preserve the consult.
7. Modify consults of rooms availability.	G1	Customer or prospect	The customer or prospect wants to reserve a room with the facility of modify the consult.
8. Destroy consults of rooms availability.	G1	Customer or prospect	The customer or prospect wants to reserve a room with the facility of destroy the consult.
24. Destroy a reservation.	G1	Customer or prospect	The customer or prospect wants to reserve a room with the facility of destroy the reservation.
11. Select hotels and rooms from images.	G1	Customer or prospect	The customer or prospect wants to reserve a room with the facility of find the availability of specific hotels, rooms and services.
12. Suggest alternative rooms and hotels.	G1	Customer or prospect	The customer or prospect wants to reserve a room with the facility of consult alternative rooms and hotels.
13. Suggest the use of a pending list to reserve rooms.	G1	Customer or prospect	The customer or prospect wants to reserve a room using the pending list.
15. Find the availability of specific (selected by the customer) hotels, rooms and services.	G1	Customer or prospect	The customer or prospect wants to reserve a room with the facility of find the availability of specific hotels, rooms and services.

S4: Discovery of Future System Scenarios through merging scenarios by goals similarity.

Input: goals and their corresponding scenarios.

Output: scenarios merged by goal.

Process:

Assess and decide

¿Are there scenarios that share the same goal?

Answer:

- Yes:

Type of scenario	Requirement	Primary agent	Goal
Alternative	11. Select hotels and rooms from images.	Customer or prospect	The customer or prospect wants to reserve a room with the facility of find the availability of specific hotels, rooms and services.
Alternative	12. Suggest alternative rooms and hotels.	Customer or prospect	The customer or prospect wants to reserve a room with the facility of consult alternative rooms and hotels.
Alternative	15. Find the availability of specific (selected by the customer) hotels, rooms and services.	Customer or prospect	The customer or prospect wants to reserve a room with the facility of find the availability of specific hotels, rooms and services.

Then:

- Merge them into one scenario:

Type of scenario	Requirement	Primary agent	Goal
Alternative	11. Select hotels and rooms from	Customer or	The customer or prospect wants to

	images. 12. Suggest alternative rooms and hotels. 15. Find the availability of specific (selected by the customer) hotels, rooms and services.	prospect	reserve a room with the facility of find the availability of specific hotels, rooms and services.

S5: Discovery of Future System Scenarios by reasoning on failure conditions (exceptional scenarios).

Input: scenarios merged by goal.

Output: requirements, conditions and "exceptional scenarios".

Process:

Assess and decide (for each requirement of the scenarios)

¿Which conditions prevent the accomplishment of each scenario goal? And ¿Which exceptional scenarios correspond to each condition?

Answer:

Type of scenario	Requirement	Rule applied	Conditions	Exceptional scenario
Normal	1. Acquire from customers and prospects: period, number of rooms, and maximum cost.	E1	If the entered data are not correct.	The new system shall handle the condition.
	2. Acquire from customers: the customer´s Id.	E1	If the customer Id. is not valid.	
	3. Acquire from the system: the	E1	- If it is not possible to establish connection with the	

	customer´s preferences and loyalty.		service to obtain the customer´s preferences and loyalty. - If there are no preferences or loyalty registered for the customer.	
	4. Preserve the consult data every time they are entered.	E1	In this case there should be no exceptions (if exist they would be low-level exceptions).	
	6. Create consults of rooms availability.	E1	In this case there should be no exceptions (if exist they would be low-level exceptions).	
	9. Process the requests instantly (NFR).	E1	If it is not possible to process the requests in less than 5 seconds.	
	10. Select the dates from calendars.	E1	If there are no availability in the desired dates.	
	14. Customize the information according to customer´s preferences.	E1	If there are no information matching the customer´s preferences	
	16. Find alternatives of hotels, rooms and services according to dates, number of rooms, and maximum cost.	E1	If there are no alternatives of hotels, rooms and services according to dates, number of rooms and maximum cost.	
	17. Find discounts according to the customer´s points registered in a loyalty program.	E1	If there are no discounts according to the customer´s points registered in a loyalty program.	
	18. To customers: send customized alternatives of hotels, rooms, services, and loyalty discounts.	E1	In this case there should be no exceptions (if exist they would be low-level exceptions).	
	19. To prospects: send information of rooms, number of beds, amenities, services, costs.	E1	In this case there should be no exceptions (if exist they would be low-level exceptions).	

	20. Acquire from customers and prospects: the selected options of rooms and services (confirm yes), and the credit card number	E1	If the credit card number is not valid.	
	21. Acquire from “consult rooms availability”: Customer´s Id., period, number of rooms, and services.	E1	If there are no connection or answer of the “consult rooms availability” service.	
	22. Preserve the reservation in .pdf format.	E1	In this case there should be no exceptions (if exist they would be low-level exceptions).	
	23. Create a reservation in a .pdf format.	E1	In this case there should be no exceptions (if exist they would be low-level exceptions).	
	25. Perform the reservations instantly.	E1	If it is not possible to perform the reservation in less than 5 seconds.	
	26. Send the reservation online.	E1	In this case there should be no exceptions (if exist they would be low-level exceptions).	
	27. Send the reservation by e-mail.	E1	If the e-mail cannot be delivered.	
Alternative	5. Preserve the customer and prospect´s consult in .pdf format.	E1	In this case there should be no exceptions (if exist they would be low-level exceptions).	
Alternative	7. Modify consults of rooms availability.	E1	In this case there should be no exceptions (if exist they would be low-level exceptions).	
Alternative	8. Destroy consults of rooms availability.	E1	In this case there should be no exceptions (if exist they would be low-level exceptions).	
Alternative	13. Suggest the use of a pending list to reserve rooms.	E1	In this case there should be no exceptions (if exist they would be low-level exceptions).	
Alternative	24. Destroy a reservation.	E1	If the reservation to destroy does not exist.	
Alternative	11. Select hotels and rooms from	E1	- In this case there should be	

	images. 12. Suggest alternative rooms and hotels. 15. Find the availability of specific (selected by the customer) hotels, rooms and services.		no exceptions (if exist they would be low-level exceptions). - If it is not possible to find alternative rooms and hotels. - If there are no available hotels or rooms or services corresponding to the selected options.	

S6: Identify Use Cases by establishing a relation between the scenario goal (from the normal and alternatives scenarios) and the required Use Cases (UC) to meet these goals.

Input: scenarios merged by goal.

Output: use cases

Process:

Assess and decide

What use case corresponds to each goal?

Answer:

Normal scenario goal	Rule applied	Use case
The customer or prospect wants to reserve a room.	UC1	Reserve rooms.
Alternative scenarios goals	Rule applied	Use case
The customer or prospect wants to reserve a room with the facility of preserve the consult.	UC1	Preserve the consult of rooms.
The customer or prospect wants to reserve a room with the	UC1	Modify the consult of rooms.

facility of modify the consult.		
The customer or prospect wants to reserve a room with the facility of destroy the consult.	UC1	Destroy the consult of rooms.
The customer or prospect wants to reserve a room using the pending list.	UC1	Reserve rooms using a pending list.
The customer or prospect wants to reserve a room with the facility of find the availability of specific hotels, rooms and services.	UC1	Consult the availability of customer´s selections of hotels, rooms and services.
The customer or prospect wants to reserve a room with the facility of destroy the reservation.	UC1	Destroy the reservation.

S7 and S8: Discovery of new Use Cases, from the discovered Use Cases, by composition: using the producing / consuming principle and reasoning on missing complementary information or services.

Input: normal scenario of the discovered use cases.

Output: new use cases.

Process:

Infer (for each use case)

What other use cases are necessary to perform the normal scenario requirements?

Answer:

Normal scenario: reserve rooms.				
Primary agent: customer or prospect.				
Scenario goal: the customer or prospect wants to reserve a room.				
Requirement	Rule applied	Discovered scenario	Missing goals of the primary agent related to the discovered scenario	Discovered use case

1. Acquire from customers and prospects: period, number of rooms, and maximum cost.	C1 (Produce, consume) pairs: - Period, number of rooms, maximum cost: (Customer or prospect, reserve rooms)	None. (There are no incomplete pairs).	None.	None.
	C2 Missing information or services required for the customer or prospect: - None.	None.	None.	None.
2. Acquire from customers: the customer´s Id.	C1 (Produce, consume) pairs: - Customer´s Id: (Customer, reserve rooms)	None. (There are no incomplete pairs).	None.	None.
	C2 Missing information or services required for the customer or prospect: - None.	None.	None.	None.
3. Acquire from the system: the customer´s preferences and loyalty.	C1 (Produce, consume) pairs: - Customer´s preferences: (?, reserve rooms) - Customer´s loyalty: (?, reserve rooms) Note: the system is a vague agent.	Consult the customer´s preferences. Consult loyalty.	The customer wants to consult his preferences. The customer wants to consult his loyalty.	The goal does not fulfill the Cockburn´s test. Consult loyalty.
	C2 Missing information or services required for the customer or prospect: - None.	None.	None.	None.
4. Preserve the	C1			

consult data every time they are entered.	(Produce, consume) pairs: - Consult data: (Customer or prospect, reserve rooms)	None. (There are no incomplete pairs).	None.	None.
	C2 Missing information or services required for the customer or prospect: - None.	None.	None.	None.
6. Create consults of rooms availability	C1 (Produce, consume) pairs: - Hotel and room: (Customer or prospect, reserve rooms)	None. (There are no incomplete pairs).	None.	None.
	C2 Missing information or services required for the customer or prospect: - Destroy consults of rooms.	Cancel consults of rooms.	The customer or prospect wants to cancel the consults of rooms.	The goal does not fulfill the Cockburn´s test.
	- Consult complementary services.	Consult complementary services.	The customer or prospect wants to consult the complementary services availability.	Consult services.
	- Cancel consults of services.	Cancel consults of services.	The customer or prospect wants to cancel the consults of services.	The goal does not fulfill the Cockburn´s test.
9. Process the requests instantly (NFR).	C1 Not apply for non-functional requirements.	None.	None.	None.

	C2 Not apply for non-functional requirements.	None.	None.	None.
10. Select the dates from calendars.	C1 (Produce, consume) pairs: - Dates: (?, reserve rooms)	Select dates from calendars.	The customer or prospect wants to select the dates from calendars.	The goal does not fulfill the Cockburn´s test.
	C2 Missing information or services required for the customer or prospect: - None.	None.	None.	None.
14. Customize the information according to customer´s preferences.	C1 (Produce, consume) pairs: - Customer´s preferences. (?, reserve rooms)	Consult the customer´s preferences.	The customer wants to consult his preferences.	The goal does not fulfill the Cockburn´s test.
	C2 Missing information or services required for the customer or prospect: - Consult services according to customer´s preferences.	Consult services.	The customer wants to consult services according to his preferences.	Consult services.
16. Find alternatives of hotels, rooms and services according to dates, number of rooms, and maximum cost.	C1 (Produce, consume) pairs: - Dates: (?, reserve rooms) - Number of rooms, maximum cost:	- Select dates from calendars. None. (There are no	- The customer or prospect wants to select the dates from calendars. None.	- The goal does not fulfill the Cockburn´s test. None.

	(Customer or prospect, reserve rooms)	incomplete pairs).		
	C2 Missing information or services required for the customer or prospect: - Consult alternative services.	Consult alternative services.	The customer or prospect wants to consult alternative services.	Consult services.
17. Find discounts according to the customer´s points registered in a loyalty program.	C1 (Produce, consume) pairs: (?,customer´s points) Note: the loyalty program is an external agent.	Consult discounts.	The customer or prospect wants to find discounts.	Consult discounts.
	C2 Missing information or services required for the customer or prospect: - None.	None.	None.	None.
18. To customers: send customized alternatives of hotels, rooms, services, and loyalty discounts.	C1 (Produce, consume) pairs: - Customer´s preferences: (?, reserve rooms)	Consult the customer´s preferences.	The customer wants to consult his preferences.	The goal does not fulfill the Cockburn´s test.
	C2 Missing information or services required for the customer or prospect: - None.	None.	None.	None.
19. To prospects: send information of rooms, number of beds, amenities, services, costs.	C1 (Produce, consume) pairs: - Prospect´s preferences. (Prospects, reserve rooms)	None. (There are no incomplete pairs).	None.	None.
	C2 Missing information or services required for the	None.	None.	None.

	customer or prospect: - None.			
20. Acquire from customers and prospects: the selected options of rooms and services (confirm yes), and the credit card number.	C1 (Produce, consume) pairs: - Rooms, services, credit card number: (Customer or prospect, reserve rooms)	None. (There are no incomplete pairs).	None.	None.
	C2 Missing information or services required for the customer or prospect: - None.	None.	None.	None.
21. Acquire from "consult rooms availability": Customer´s Id., period, number of rooms, and services.	C1 (Produce, consume) pairs: - Customer´s Id., period, number of rooms, and services: (Consult rooms availability, reserve rooms)	None. (There are no incomplete pairs).	None.	None.
	C2 Missing information or services required for the customer or prospect: - None.	None.	None.	None.
22. Preserve the reservation in .pdf format.	C1 (Produce, consume) pairs: - Reservation: (?, reserve rooms)	Save reservation.	The customer or prospect wants to preserve the reservation.	The goal does not fulfill the Cockburn´s test.
	C2 Missing information or services required for the customer or prospect: - None.	None.	None.	None.

23. Create a reservation in a .pdf format.	C1 (Produce, consume) pairs: - Reservation: (Reserve rooms, reserve rooms)	None. (There are no incomplete pairs).	None.	None.
	C2 Missing information or services required for the customer or prospect: - Destroy reservation. - Reserve services.	 Destroy reservation. Reserve services.	 The customer or prospect wants to destroy a reservation. The customer or prospect wants to reserve services.	 Cancel reservations. Reserve services.
25. Perform the reservations instantly.	C1 Not apply for non-functional requirements.	None.	None.	None.
	C2 Not apply for non-functional requirements.	None.	None.	None.
26. Send the reservation online.	C1 (Produce, consume) pairs: - Reservation: (?, customer or prospect)	Consult reservations.	The customer or prospect wants to consult reservations.	Consult reservations.
	C2 Missing information or services required for the customer or prospect: - None.	None.	None.	None.

27. Send the reservation by e-mail.	C1 (Produce, consume) pairs: - Reservation: (?, customer or prospect)	Receive reservations by e-mail.	The customer or prospect wants to receive reservations by e-mail.	The goal does not fulfill the Cockburn´s test.
	C2 Missing information or services required for the customer or prospect: - None.	None.	None.	None.

Summary of the new use cases:

- Consult loyalty.	- Cancel reservations.
- Consult services.	- Reserve services.
- Consult discounts.	- Consult reservations.

S9: Identify associations between Use Cases by reasoning on the type of relation existing between them.

Input: use cases

Output: use cases and their associations

Process:

Assess and decide (for each use case)

If the use case is necessary to accomplish another one then

- Establish an <includes> association between the use cases.

If the use case extends the behavior of another one then

- Establish an <extends> association between the use cases.

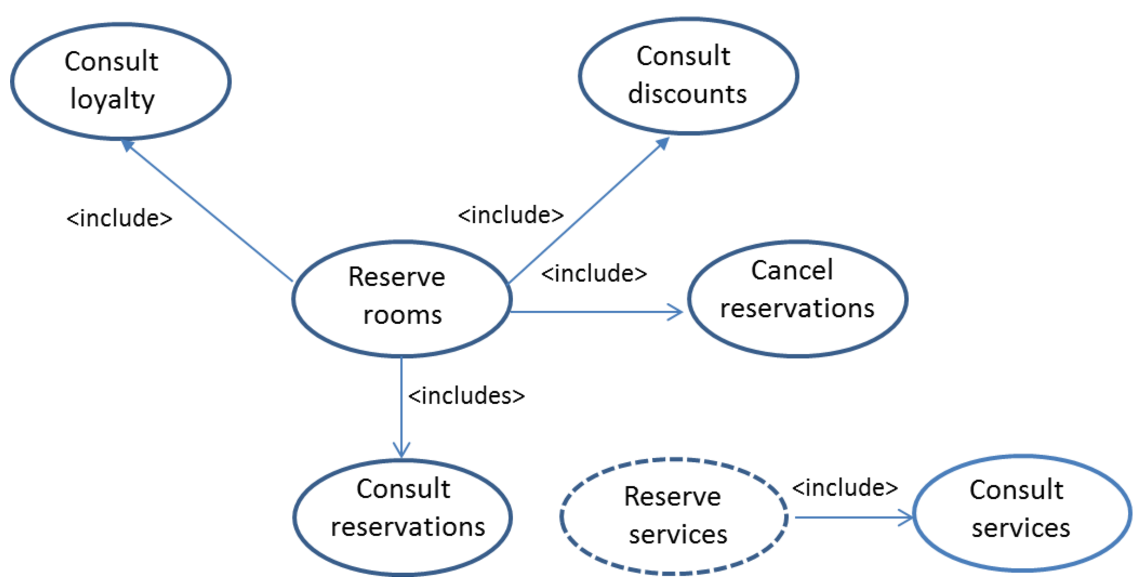
If the use case “is a kind” of another one then

- Establish a <generalization / specialization> association between the use cases.

The existing associations are:

	Reserve rooms	Consult loyalty	Consult services	Consult discounts	Cancel reservations	Reserve services	Consult reservations.
Reserve rooms		Include		Include	Include		Include
Consult loyalty							
Consult services							
Consult discounts							
Cancel reservations							
Reserve services			Include				
Consult reservations							

The resulting use case diagram is:



Note: the use case in dashed line corresponds to the one selected to be analyzed as indicated in the next step.

S10: Select the next Use Cases to be processed by reasoning on Use Cases and their relations.

Input: new Use cases.

Output: new Use cases to be processed.

Process:

Assess and decide (for each new Use case)

If the (Use case is included by other Use case) or (the Use case is an extension of other Use case) then

- The Use case shouldn't be processed.

→This implies to discard the next Use cases: consult loyalty, consult discounts, consult services, consult reservations, and cancel reservations.

Else

If the Use case analysis can lead to the discovery of new scenarios of the system then

- The Use case must be processed: execute S11with the selected Use case.

→ The selected Use case is: reserve services.

End-if

End-if

If there are no Use cases to be processed then

- Stop.

End-if

In summary, the use case to be processed is "reserve services".

S11: For the selected Use case: Build the Abstract Sequence Diagram of its normal scenario by envisioning its future behavior.

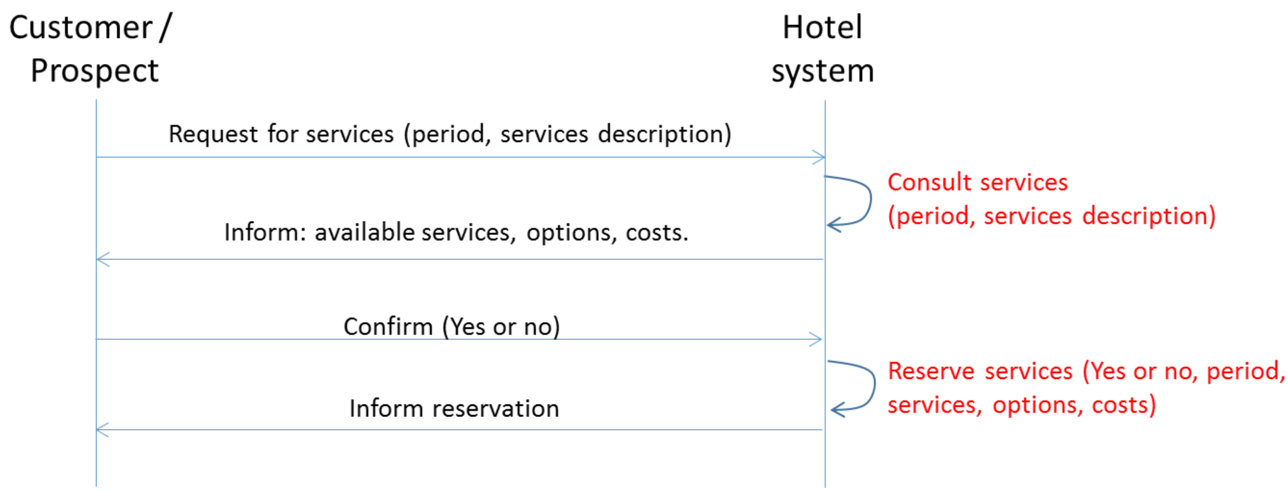
Guideline id.: BASD

Input: Reserve services.

Output:
Abstract Sequence Diagram corresponding to the normal scenario of the use case to be processed.

Process:
Create, Assess and decide
The System owner, with Requirements Engineer assistance, must envisions the behavior of the functionality in consideration by means of the next steps:
Step 1. Identify the Primary Agent of the normal scenario and his goal.
Considering the normal scenario:
Step 2. Identify the scenario interactions between the Primary Agent and the system.
Step 3. Identify the data required by each interaction in order to be successfully executed.
Step 4. Identify the system internal actions.
Step 5. Identify the data required by each system internal action in order to be successfully executed.
Step 6. Build the Abstract Sequence Diagram taking into account the elements discovered in the previous steps.

Applying this guideline, the System owner creates the next ASD for the FSS: *Reserve services:*



Step 7. Start the DREQUS process using this new Abstract Sequence Diagram (execute S1 - stage 1 (Reserve services ASD)).

In this manner, the DREQUS process is repeated for each new Future System Scenario until there are no new scenarios to be processed (this saturation condition is validated by the strategy S10). As summary, Figure 19 presents a glimpse of the DREQUS approach and Figure 20 condenses its main concepts and relations.

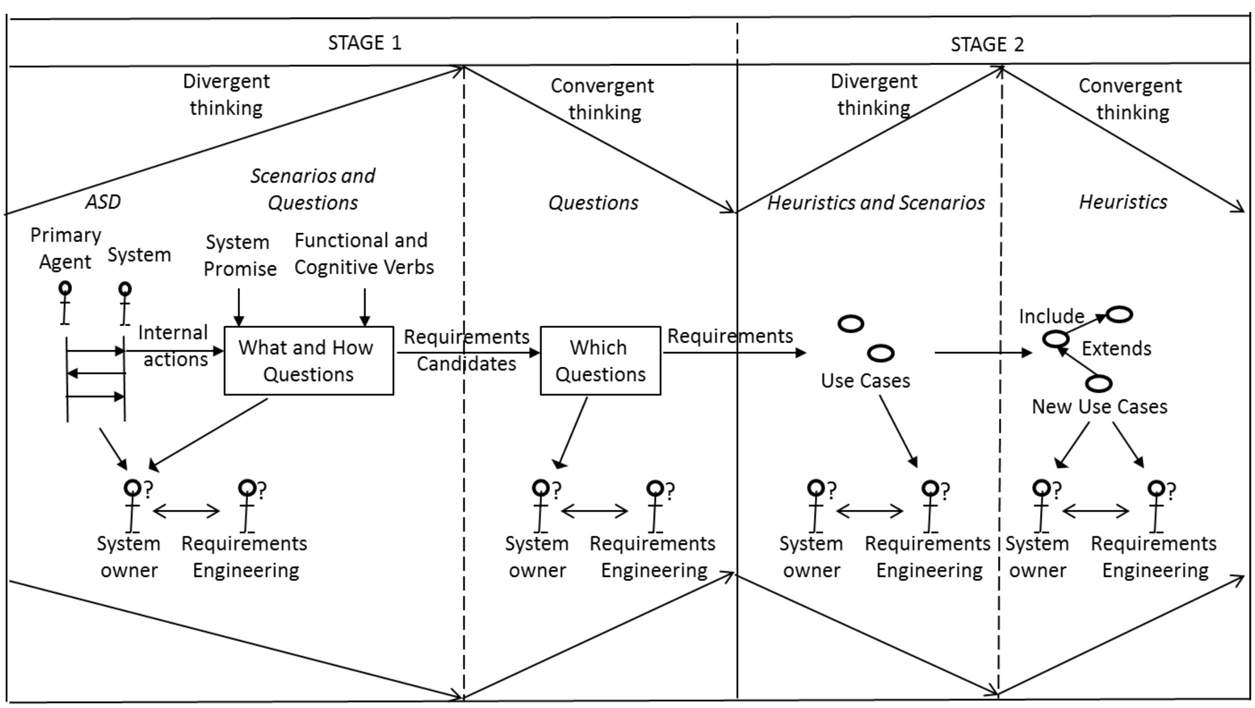


Figure 19. DREQUS in a glimpse.

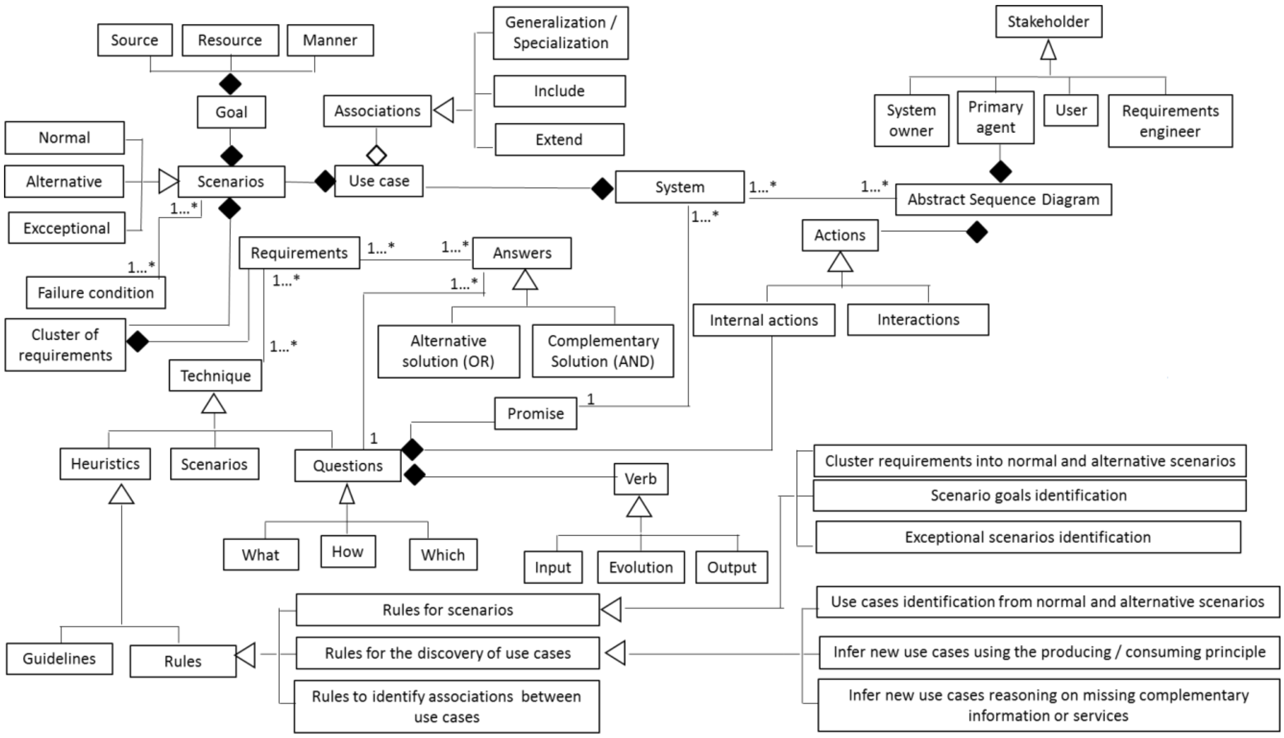


Figure 20. Concepts and relations of the DREQUS approach.

3.3. Conclusions

As evidenced in the state of the art (chapter 2), in recent years, Creativity-based Approaches for Requirements Elicitation (CAREs) emerge as a promising trend aimed to facilitate the discovery of requirements in the context of software development projects. However, these approaches present the next drawbacks:

a) There is a lack of mechanisms aimed to ensure the alignment of the discovered requirements with the system purpose.

b) There is a need to bring support and guidance to the stakeholders in the systematic exploration of the entire SIS and the consequent discovery of requirements.

c) The current solutions do not facilitate the transition from the discovered requirements towards the future system specification.

Based on Design Science Research methodology (DSR), we have conceived and designed a proposal, named DREQUS (Discovery of REQuirements Using Scenarios) to tackle the aforementioned issues. In this chapter, DREQUS was presented and illustrated with a "*Booking rooms system"* case study.

The processes and mechanisms of DREQUS contribute to improving the quality of the discovered requirements and software products derived from them. In order to validate this assertion, two Empirical Studies were performed: their designs, results, and conclusions are presented in the next chapter.

REFERENCES

[1] Cheng, B.H.C., Atlee, J.M.: "Research Directions in Requirements Engineering". In Proceedings Future of Software Engineering (FOSE'07), p. 285-303. 2007.

[2] Hall, T., Beecham, S., Rainer, A.: "Requirements problems in twelve software companies: an empirical analysis". Software, IEE Proceedings -, vol. 149, no 5, p. 153-160. 2002.

[3] Sutcliffe, A., Sawyer, P.: "Requirements elicitation: Towards the unknown unknowns". In Proceedings 21st IEEE Int. Requirements Eng. Conference (RE'2013), Rio de Janeiro, Brazil, p. 92-104. 2013.

[4] Nuseibeh, B., Easterbrook, S.: "Requirements Engineering: a roadmap". In Proceedings of the conference of the Future of Software Engineering, p. 35–46. 2000.

[5] Zowghi, D. and Coulin, C.: Requirements Elicitation: A Survey of Techniques, Approaches, and Tools. In: Aurum, A. and Wohlin, C. (eds.), Engineering and Managing Softaware Requirements. pp. 19 46. Springer Berlin Heidelberg. 2005.

[6] Standish Group. The Chaos Report.. Available at: https://www.standishgroup.com/. 2018.

[7] Hevner, A.R., March, S.T., and Park, J. Design Research in Information Systems Research. MIS Quarterly, 28, 1, 75-105. 2004.

[8] Peffers, K., Tuunanen T., Chatterjee M.A, Rothenberger S. A design science research methodology for information systems research. Journal of Management Information Systems 24, n° 3 pp. 45--77. 2007.

[9] Järvinen, P. Action Research is Similar to Design Science. *Quality & Quantity*, 41, 1, 37-54. 2007.

[10] Jaramillo A. Requirements elicitation approaches: a systematic review. Research Challenges in Information Science (RCIS), IEEE 9th International Conference on. Athens. Greece. 2015.

[11] Jaramillo, A., Assar, S. Leveraging creativity techniques in requirements elicitation: a literature review. Requirements Engineering Magazine. International Requirements Engineering Board. Issue 2016-02. 2016.

[12] Horkoff J., Li T., F., Salnitri M., Cardoso E., Giorgini P., Mylopoulos J.and Pimentel J. Taking Goal Models Downstream: A Systematic Roadmap. Research Challenges in Information Science (RCIS), IEEE Eighth International Conference on. Page(s): 1 – 12. 2014.

[13] Kitchenham B., Dyba T., and Jorgensen M. "Evidence-based software engineering," in Proc. of the 26th International Conference on Software Engineering (ICSE04), pp. 273–281. 2004.

[14] Kitchenham, R. Pretorius, D. Budgen, O. P. Brereton, M. Turner, M. Niazi, and S. Linkman, "Systematic literature reviews in software engineering – a tertiary study," Information and Software Technology, vol. 52, no. 8, pp. 792 – 805, 2010.

[15] Feldt R. and Magazinius A.. Validity Threats in Empirical Software Engineering Research-An Initial Survey. In Proceedings of the 22nd International Conference on Software Engineering and Knowledge Engineering, pp. 374-379. 2010.

[16] Lee S. W. and Rine D. C. Case Study Methodology Designed Research in Software Engineering Methodology Validation. In Proceedings of the Sixteenth International Conference on Software Engineering and Knowledge Engineering (SEKE'04), pp. 117-122, Banff Centre, Banff, Alberta, Canada, June 20 to June 24, 2004.

[17] Sakhnini V., Mich L., Berry D. M. Group versus individual use of power-only EPMcreate as a creativity enhancement technique for requirements elicitation. Springer Science+Business Media New York. 2017.

[18] Harmain, H., Gaizauskas, R.: CM-Builder. An automated NL-Based CASE Tool. Proceedings of the Fifteenth IEEE International Conference on Automated Software Engineering (ASE´00). Grenoble. 2000.

[19] Alley M. The craft of scientific writing. Springer-Verlag New York. ISBN 978-1-4419-8288-9. 2018.

[20] Shaw M. Writing Good Software Engineering Research Papers. In Proceedings of the 25th International Conference on Software Engineering (ICSE03). 2003.

[21] Nguyen L. and Swatman P. A. Promoting and Supporting the Creative and Insight-Driven RE Process Using Design Rationale. Working Paper, 15, UNISA, Adelaide, 2006.

[22] Ben Achour C. and Rolland C. Introducing genericity and modularity of textual scenario interpretation in the context of requirements engineering. Centre de Recherche en Informatique. Université de Paris I – Sorbonne. 1997.

[23] Sternberg R. J. The nature of creativity. Creativity Research Journal, Vol. 18, No. 1, 87–98. 2006.

[24] Berry D., Mich L. and Sakhnini V. The effectiveness of an optimized EPMcreate as a creativity enhancement technique for web site requirements elicitation. Requirements Engineering Journal. 2012.

[25] Herrmann, A., Mich, L., and Berry, D.M. Creativity Techniques for Requirements Elicitation: Comparing Four-Step EPMcreate-Based Processes, Seventh Workshop on Empirical Requirements Engineering (EmpiRE), Banff, AB, Canada, August 2018.

[26] Rolland C. and Prakash N. Matching ERP System Functionality to Customer Requirements. In Proceedings of the 5th IEEE International Symposium on Requirements Engineering, Toronto, Canada. August 27-31, 2001.

[27] Cockburn A. Structuring Use Cases with Goals. Technical report. Human and Technology, 7691 Dell Rd, Salt Lake City, UT 84121, HaT.TR.95.1, http://alistair.cockburn.us/Structuring+use+cases+with+goals, 1995.

[28] Calonius, H.: The Promise Concept. Swedish School of Economics and Business Administration. 1984.

[29] von Bertalanffy L. General Theory Systems: Foundations, Development, Applications. University of Alberta, Edmonton, Canada. George Brazziller. New York. 1969.

[30] Fillmore Ch. “The Case for Case”. In: Universals in Linguistics Theory. E. Bach. R. Harms eds. Holt, Rinehart and Winston Publishing Company. P. 1-90. 1968.

[31] Levin, B. English verb classes and alternations: a preliminary investigation. Chicago Illinois: University of Chicago Press. 1993.

[32] Mendling, J., Recker, J., and Reijers, H. On the usage of labels and icons in business process modeling. International Journal of Information System Modeling and Design, 1(2), 40-58. 2010.

[33] Malone, T. W., Crowston, K., & Herman, G. A. *Organizing Business Knowledge: The MIT Process Handbook*. Cambridge, Massachusetts: The MIT Press. 2003.

[34] Keller, G., & Teufel, T. *SAP R/3 Process Oriented Implementation: Iterative Process Prototyping*. Boston, Massachusetts: Addison-Wesley. 1998.

[35] Metzler, T., and Shea, K. Taxonomy of cognitive functions. Proceedings of the 18th International Conference on Engineering Design. ICED11. 2011.

[36] Princeton University "About WordNet." WordNet. Princeton University. Available at: http://wordnet.princeton.edu/ 2019.

[37] Miller G. A., Beckwith R., Fellbaum C., Gross D. and Miller K. Introduction to WordNet: An On-line Lexical Database. International Journal of Lexicography. Oxford University Press. 1993. Available at: http://wordnetcode.princeton.edu/5papers.pdf

[38] Piasecki M., Marci´nczuk M., Musiał A., Ramocki R., Maziarz M. WordnetLoom: a Graph-based Visual Wordnet Development Framework. Proceedings of the International Multiconference on Computer Science and Information Technology pp. 469–476. IEEE. 2010.

[39] Vercruysse S. Wordvis. NTNU university. Norway. Available at: : http://wordvis.com/

[40] Pedersen T., Patwardhan S., Michelizzi J. WordNet::Similarity - Measuring the Relatedness of Concepts. Copyright c©. American Association for Artificial Intelligence (www.aaai.org). 2004.

[41] Pedersen T., Patwardhan S., Banerjee S. and Michelizzi J. WordNet::Similarity. 2019. Available at: http://maraca.d.umn.edu/cgi-bin/similarity/similarity.cgi.

[42] Vossen P., Bloksma L., Alonge A., Amarina E., Peters C., Castellón I., Matí A and Rigau G. Compatibility in Interpretation of Relations in EuroWordNet. Computers and the Humanities 32: 153–184, Kluwer Academic Publishers. Printed in the Netherlands. 1998.

[43] Ellis N. C., Brook O'Donnell M. and Römer U. The processing of verb-argument constructions is sensitive to form, function, frequency, contingency and prototypicality. Cognitive Linguistics; 25(1): 55 -98. 2014.

[44] Sandiway F. Wnconnect software. Available at: http://dingo.sbs.arizona.edu/~sandiway/wnconnect/index.html. 2003.

[45] Rolland, C., Salinesi, C.: Supporting requirements elicitation through goal/scenario coupling. Conceptual Modeling: Foundations and Applications, LNCS. Vol. 5600. Pp. 398-416. Springer. 2009.

[46] Sutcliffe, A., Maiden, N., Minocha, S., Manuel, D.: Supporting scenario-based Requirements Engineering. IEEE Transactions on software engineering, vol. 24 no. 12, December. 1998.

[47] Aurum, A., Wohlin, C.: "The Fundamental Nature of Requirements Engineering Activities as a Decision-Making Process". Information and Software Technology, Vol. 45, No. 14, pp 945-954. 2003.

[48] Natt och Dag J., Gervasi V., Brinkkemper S., Regnell B. A linguistic-engineering approach to large-scale requirements management. IEEE Software 22(1):32–39. 2005.

[49] String similarity. 2019 Tool available at: at http://www.tools4noobs.com/online_tools/string_similarity/

[50] Semantic similarity org. 2019. Tool available at: http://semanticsimilarity.org/

[51] University of Colorado. 2019. Tool available at: http://lsa.colorado.edu/

[52] Natt och Dag J. 2019. Tool available at: https://sourceforge.net/projects/reqsimile/

[53] Pedersen T., Purandare A., Kulkarni A., Joshi M. 2019. Tool available at: http://www.d.umn.edu/~tpederse/senseclusters.html

[54] Küng J., Muhammad I. A comparative analysis of similarity measurement techniques through SimReq framework. FIT'09. December 16-18. CIIT. Abbottabad. Pakistan. 2009.

[55] Hsia P., Hsu C.T., Kung D.C., and Holder L.B. "User- Centered System Decomposition: Z-Based Requirements Clustering", *Proceedings of the 2nd international Conference on Requirements Engineering,* Colorado Springs, pp. 126. 1996.

[56] Yaung A.T. "Design and implementation of a requirements clustering analyzer for software system decomposition", *ACM/SIGAPP Symposium on Applied Computing: Technological Challenges of the 1990's*, Kansas City, MO, 1992.

[57] Al-Otaiby T.N., AlSherif M., and Bond W.P. "Toward software requirements modularization using hierarchical clustering techniques", *43rd Annual Southeast Regional Conference*, Vol. 2, Kennesaw, GA, pp. 223-228. 2005.

[58] Cleland-Huang J., Laurent P., Duan C. Towards automated requirements triage. 15th IEEE. International Requirements Requirements Engineering Conference. 2007.

[59] Duan C. Clustering and its application in requirements engineering. Technical Report, DePaul University, Chicago, Illinois, 2008.

[60] Rolland C., Souveyet C., Ben Achour C., Guiding goal modeling using scenarios. IEEE Transactions on software engineering, vol. 24, no. 12, December. 1998.

[61] Cockburn A. *Structuring Use Cases with Goals*. Technical report. Human and Technology, 7691 Dell Rd, Salt Lake City, UT 84121, HaT.TR.95.1. Available at: http://alistair.cockburn.us/Structuring+use+cases+with+goals.. 1995.

[62] Larman, C.: *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development* – ISBN 0-13-148906-2. P. 74. 2004.

[63] IBM SPSS Statistics. Available at: https://www.ibm.com/es-es/analytics/spss-trials 2019.

Appendix I. Semantic relatedness between the Functional and Cognitive verbs (1 of 2)

	Acquire	Appear	Appreciate	Assess	Calculate	Care	Coact	Combine	Communicate	Complete	Create	Decide	Deduce	Destroy	Display	Engender	Find	Focus	Interpret	Judge	Know	Lodge
Acquire	1	0,3333	0,25	0,25	0,3333	0,25	0,25	0,25	0,2	0,3333	0,3333	0,3333	0,2	0,3333	0,25	0,25	0,5	0,25	0,25	0,25	0,3333	0,2
Appear	0,3333	1	0,25	0,25	0,3333	0,25	0,25	0,25	0,2	0,3333	0,3333	0,3333	0,2	0,3333	0,25	0,25	0,3333	0,25	0,3333	0,25	0,3333	0,2
Appreciate	0,25	0,25	1	0,25	0,25	0,2	0,2	0,25	0,1667	0,25	0,25	0,25	0,1667	0,25	0,2	0,2	0,25	0,2	0,2	0,25	0,25	0,1667
Assess	0,25	0,25	0,25	1	0,3333	0,2	0,2	0,2	0,1667	0,25	0,25	0,25	0,2	0,25	0,2	0,2	0,25	0,25	0,2	0,5	0,25	0,1667
Calculate	0,3333	0,3333	0,25	0,3333	1	0,25	0,25	0,25	0,2	0,3333	0,3333	0,3333	0,2	0,3333	0,25	0,25	0,3333	0,25	0,25	0,5	0,3333	0,2
Care	0,25	0,25	0,2	0,2	0,25	1	0,2	0,2	0,1667	0,25	0,25	0,25	0,1667	0,25	0,2	0,2	0,25	0,2	0,2	0,2	0,25	0,1667
Coact	0,25	0,25	0,2	0,2	0,25	0,2	1	0,25	0,25	0,25	0,3333	0,25	0,1667	0,25	0,25	0,2	0,25	0,2	0,2	0,2	0,25	0,1667
Combine	0,25	0,25	0,25	0,2	0,25	0,2	0,25	1	0,3333	0,25	0,25	0,25	0,1667	0,25	0,2	0,2	0,25	0,2	0,2	0,2	0,25	0,1667
Communicate	0,2	0,2	0,1667	0,1667	0,2	0,1667	0,25	0,3333	1	0,2	0,25	0,2	0,1429	0,2	0,2	0,1667	0,2	0,1667	0,25	0,1667	0,2	0,1429
Complete	0,3333	0,3333	0,25	0,25	0,3333	0,25	0,25	0,25	0,2	1	0,3333	0,3333	0,2	0,3333	0,25	0,25	0,3333	0,25	0,25	0,25	0,3333	0,2
Create	0,3333	0,3333	0,25	0,25	0,3333	0,25	0,3333	0,25	0,25	0,3333	1	0,3333	0,2	0,3333	0,25	0,5	0,3333	0,25	0,3333	0,25	0,3333	0,2
Decide	0,3333	0,3333	0,25	0,25	0,3333	0,25	0,25	0,25	0,2	0,3333	0,3333	1	0,2	0,3333	0,25	0,25	0,3333	0,25	0,25	0,5	0,3333	0,2
Deduce	0,2	0,2	0,1667	0,2	0,2	0,1667	0,1667	0,1667	0,1429	0,2	0,2	0,2	1	0,2	0,1667	0,1667	0,3333	0,25	0,1667	0,25	0,2	0,1429
Destroy	0,3333	0,3333	0,25	0,25	0,3333	0,25	0,25	0,25	0,2	0,3333	0,3333	0,3333	0,2	1	0,25	0,25	0,3333	0,25	0,25	0,25	0,3333	0,2
Display	0,25	0,25	0,2	0,2	0,25	0,2	0,25	0,2	0,2	0,25	0,25	0,25	0,1667	0,25	1	0,2	0,25	0,2	0,2	0,2	0,25	0,1667
Engender	0,25	0,25	0,2	0,2	0,25	0,2	0,2	0,2	0,1667	0,25	0,5	0,25	0,1667	0,25	0,2	1	0,25	0,2	0,25	0,2	0,25	0,1667
Find	0,5	0,3333	0,25	0,25	0,3333	0,25	0,25	0,25	0,2	0,3333	0,3333	0,3333	0,3333	0,3333	0,25	0,25	1	0,25	0,25	0,5	0,3333	0,2
Focus	0,25	0,25	0,2	0,25	0,25	0,2	0,2	0,2	0,1667	0,25	0,25	0,25	0,25	0,25	0,2	0,2	0,25	1	0,2	0,3333	0,25	0,1667
Interpret	0,25	0,3333	0,2	0,2	0,25	0,2	0,2	0,2	0,25	0,25	0,3333	0,25	0,1667	0,25	0,2	0,25	0,25	0,2	1	0,2	0,25	0,1667
Judge	0,25	0,25	0,25	0,5	0,5	0,2	0,2	0,2	0,1667	0,25	0,25	0,5	0,25	0,25	0,2	0,2	0,5	0,3333	0,2	1	0,3333	0,1667
Know	0,3333	0,3333	0,25	0,25	0,3333	0,25	0,25	0,25	0,2	0,3333	0,3333	0,3333	0,2	0,3333	0,25	0,25	0,3333	0,25	0,25	0,3333	1	0,2
Lodge	0,2	0,2	0,1667	0,1667	0,2	0,1667	0,1667	0,1667	0,1429	0,2	0,2	0,2	0,1429	0,2	0,1667	0,1667	0,2	0,1667	0,1667	0,1667	0,2	1
Manage	0,3333	0,3333	0,25	0,25	0,3333	1	0,3333	0,25	0,25	0,3333	0,3333	0,3333	0,2	0,3333	0,25	0,25	0,3333	0,25	0,25	0,25	0,3333	0,2
Measure	0,25	0,3333	0,25	1	0,3333	0,2	0,2	0,2	0,3333	0,25	0,25	0,5	0,2	0,25	0,2	0,2	0,25	0,25	0,2	0,5	0,25	0,1667
Modify	0,3333	0,3333	0,25	0,25	0,3333	0,25	0,25	0,3333	0,2	0,3333	0,3333	0,3333	0,2	0,3333	0,25	0,25	0,3333	0,3333	0,25	0,25	0,3333	0,2
Move	0,3333	0,3333	0,25	0,25	0,3333	0,25	0,5	0,3333	0,3333	0,3333	0,5	0,3333	0,2	0,3333	0,3333	0,3333	0,3333	0,25	0,25	0,25	0,3333	0,2
Perceive	0,3333	0,3333	0,3333	0,25	0,3333	0,25	0,25	0,25	0,2	0,3333	0,3333	0,3333	0,2	0,3333	0,25	0,25	0,5	0,25	0,25	0,25	0,3333	0,2
Plan	0,25	0,25	0,2	0,25	0,3333	0,2	0,2	0,2	0,1667	0,25	0,3333	0,25	0,25	0,25	0,25	0,25	0,25	0,3333	0,2	0,3333	0,25	0,1667
Preserve	0,25	0,25	0,2	0,2	0,25	0,2	0,2	0,2	0,1667	0,25	0,25	0,25	0,1667	0,25	0,2	0,2	0,25	0,2	0,2	0,2	0,25	0,1667
Promise	0,25	0,3333	0,2	0,2	0,5	0,2	0,2	0,2	0,25	0,25	0,25	0,25	0,1667	0,25	0,2	0,2	0,25	0,2	0,2	0,2	0,25	0,1667
Put	0,25	0,25	0,2	0,3333	0,3333	0,3333	0,2	0,25	0,25	0,25	0,25	0,25	0,1667	0,25	0,2	0,2	0,25	0,25	0,2	0,5	0,25	0,1667
React	0,25	0,25	0,2	0,2	0,25	0,2	0,3333	0,25	0,25	0,25	0,3333	0,25	0,1667	0,25	0,25	0,2	0,25	0,2	0,2	0,2	0,25	0,1667
Reason	0,25	0,25	0,2	0,25	0,5	0,2	0,2	0,2	0,1667	0,25	0,25	0,25	0,5	0,25	0,2	0,2	0,5	0,3333	0,2	0,3333	0,25	0,1667
Reflect	0,3333	0,3333	0,25	0,25	0,3333	0,25	0,25	0,25	0,25	0,3333	0,3333	0,3333	0,25	0,3333	0,3333	0,25	0,3333	0,3333	0,25	0,3333	0,3333	0,2
Remove	0,3333	0,3333	0,25	0,25	0,3333	0,25	0,25	0,25	0,2	0,3333	0,3333	0,3333	0,2	0,3333	0,25	0,25	0,3333	0,25	0,25	0,25	0,3333	0,2
Schedule	0,2	0,2	0,1667	0,2	0,25	0,1667	0,1667	0,1667	0,1429	0,2	0,2	0,2	0,2	0,2	0,1667	0,1667	0,2	0,25	0,1667	0,25	0,2	0,1429
Search	0,3333	0,3333	0,25	0,25	0,3333	0,25	0,25	0,25	0,2	0,3333	0,3333	0,3333	0,2	0,3333	0,25	0,25	0,3333	0,25	0,25	0,25	0,3333	0,2
Send	0,25	0,25	0,2	0,2	0,25	0,2	0,2	0,2	0,3333	0,25	0,25	0,25	0,1667	0,25	0,2	0,2	0,25	0,2	0,2	0,2	0,25	0,1667
Separate	0,3333	0,3333	0,25	0,25	0,3333	0,25	0,25	0,3333	0,25	0,3333	0,3333	0,3333	0,2	0,3333	0,25	0,25	0,3333	0,25	0,25	0,25	0,5	0,2
Solve	0,25	0,25	0,2	0,25	0,5	0,2	0,2	0,2	0,1667	0,25	0,25	0,25	0,1667	0,25	0,2	0,2	0,25	0,2	0,3333	0,3333	0,25	0,1667
Think	0,3333	0,3333	0,3333	0,3333	0,3333	0,25	0,25	0,25	0,2	0,3333	0,3333	0,3333	0,3333	0,3333	0,25	0,25	0,3333	0,5	0,25	0,5	0,5	0,2
Transform	0,3333	0,25	0,2	0,2	0,25	0,2	0,2	0,25	0,1667	0,25	0,25	0,25	0,1667	0,25	0,2	0,2	0,3333	0,25	0,2	0,2	0,25	0,1667
Understand	0,3333	0,3333	0,5	0,25	0,3333	0,25	0,25	0,25	0,2	0,3333	0,3333	0,3333	0,2	0,3333	0,25	0,25	0,3333	0,25	1	0,25	0,3333	0,2

Appendix I. Semantic relatedness between the Functional and Cognitive verbs (2 of 2)

	Manage	Measure	Modify	Move	Perceive	Plan	Preserve	Promise	Put	React	Reason	Reflect	Remove	Schedule	Search	Send	Separate	Solve	Think	Transform	Understand
Acquire	0,3333	0,25	0,3333	0,3333	0,3333	0,25	0,25	0,25	0,25	0,25	0,25	0,3333	0,3333	0,2	0,3333	0,25	0,3333	0,25	0,3333	0,3333	0,3333
Appear	0,3333	0,3333	0,3333	0,3333	0,3333	0,25	0,25	0,3333	0,25	0,25	0,25	0,3333	0,3333	0,2	0,3333	0,25	0,3333	0,25	0,3333	0,25	0,3333
Appreciate	0,25	0,25	0,25	0,25	0,3333	0,2	0,2	0,2	0,2	0,2	0,2	0,25	0,25	0,1667	0,25	0,2	0,25	0,2	0,3333	0,2	0,5
Assess	0,25	1	0,25	0,25	0,25	0,25	0,2	0,2	0,3333	0,2	0,25	0,25	0,25	0,2	0,25	0,2	0,25	0,25	0,3333	0,2	0,25
Calculate	0,3333	0,3333	0,3333	0,3333	0,3333	0,3333	0,25	0,5	0,3333	0,25	0,5	0,3333	0,3333	0,25	0,3333	0,25	0,3333	0,5	0,3333	0,25	0,3333
Care	1	0,2	0,25	0,25	0,25	0,2	0,2	0,2	0,3333	0,2	0,2	0,25	0,25	0,1667	0,25	0,2	0,25	0,2	0,25	0,2	0,25
Coact	0,3333	0,2	0,25	0,5	0,25	0,2	0,2	0,2	0,2	0,3333	0,2	0,25	0,25	0,1667	0,25	0,2	0,25	0,2	0,25	0,2	0,25
Combine	0,25	0,2	0,3333	0,3333	0,25	0,2	0,2	0,2	0,25	0,25	0,2	0,25	0,25	0,1667	0,25	0,2	0,3333	0,2	0,25	0,25	0,25
Communicate	0,25	0,3333	0,2	0,3333	0,2	0,1667	0,1667	0,25	0,25	0,25	0,1667	0,25	0,2	0,1429	0,2	0,3333	0,25	0,1667	0,2	0,1667	0,2
Complete	0,3333	0,25	0,3333	0,3333	0,3333	0,25	0,25	0,25	0,25	0,25	0,25	0,3333	0,3333	0,2	0,3333	0,25	0,3333	0,25	0,3333	0,25	0,3333
Create	0,3333	0,25	0,3333	0,5	0,3333	0,3333	0,25	0,25	0,25	0,3333	0,25	0,3333	0,3333	0,2	0,3333	0,25	0,3333	0,25	0,3333	0,25	0,3333
Decide	0,3333	0,5	0,3333	0,3333	0,3333	0,25	0,25	0,25	0,25	0,25	0,25	0,3333	0,3333	0,2	0,3333	0,25	0,3333	0,25	0,3333	0,25	0,3333
Deduce	0,2	0,2	0,2	0,2	0,2	0,25	0,1667	0,1667	0,1667	0,1667	0,5	0,25	0,2	0,2	0,2	0,1667	0,2	0,1667	0,3333	0,1667	0,2
Destroy	0,3333	0,25	0,3333	0,3333	0,3333	0,25	0,25	0,25	0,25	0,25	0,25	0,3333	0,3333	0,2	0,3333	0,25	0,3333	0,25	0,3333	0,25	0,3333
Display	0,25	0,2	0,25	0,3333	0,25	0,25	0,2	0,2	0,2	0,25	0,2	0,3333	0,25	0,1667	0,25	0,2	0,25	0,2	0,25	0,2	0,25
Engender	0,25	0,2	0,25	0,3333	0,25	0,25	0,2	0,2	0,2	0,2	0,2	0,25	0,25	0,1667	0,25	0,2	0,25	0,2	0,25	0,2	0,25
Find	0,3333	0,25	0,3333	0,3333	0,5	0,25	0,25	0,25	0,25	0,25	0,5	0,3333	0,3333	0,2	0,3333	0,25	0,3333	0,25	0,3333	0,3333	0,3333
Focus	0,25	0,25	0,3333	0,25	0,25	0,3333	0,2	0,2	0,25	0,2	0,3333	0,3333	0,25	0,25	0,25	0,2	0,25	0,2	0,5	0,25	0,25
Interpret	0,25	0,2	0,25	0,25	0,25	0,2	0,2	0,2	0,2	0,2	0,2	0,25	0,25	0,1667	0,25	0,2	0,25	0,3333	0,25	0,2	1
Judge	0,25	0,5	0,25	0,25	0,25	0,3333	0,2	0,2	0,5	0,2	0,3333	0,3333	0,25	0,25	0,25	0,2	0,25	0,3333	0,5	0,2	0,25
Know	0,3333	0,25	0,3333	0,3333	0,3333	0,25	0,25	0,25	0,25	0,25	0,25	0,3333	0,3333	0,2	0,3333	0,25	0,5	0,25	0,5	0,25	0,3333
Lodge	0,2	0,1667	0,2	0,2	0,2	0,1667	0,1667	0,1667	0,1667	0,1667	0,1667	0,2	0,2	0,1429	0,2	0,1667	0,2	0,1667	0,2	0,1667	0,2
Manage	1	0,25	0,3333	0,5	0,3333	0,25	0,25	0,25	0,3333	0,3333	0,25	0,3333	0,3333	0,2	0,3333	0,25	0,3333	0,25	0,3333	0,25	0,3333
Measure	0,25	1	0,25	0,25	0,25	0,25	0,2	0,3333	0,2	0,2	0,25	0,25	0,25	0,2	0,25	0,2	0,25	0,2	0,3333	0,25	0,25
Modify	0,3333	0,25	1	0,3333	0,3333	0,25	0,25	0,25	0,5	0,25	0,25	0,3333	0,3333	0,2	0,3333	0,25	0,3333	0,25	0,5	0,5	0,3333
Move	0,5	0,25	0,3333	1	0,3333	0,25	0,25	0,25	0,5	0,5	0,25	0,3333	0,3333	0,2	0,3333	0,5	0,5	0,25	0,3333	0,3333	0,3333
Perceive	0,3333	0,25	0,3333	0,3333	1	0,25	0,25	0,25	0,25	0,25	0,25	0,3333	0,3333	0,2	0,3333	0,25	0,3333	0,25	0,3333	0,25	0,5
Plan	0,25	0,25	0,25	0,25	0,25	1	0,2	0,2	0,2	0,2	0,3333	0,3333	0,25	0,5	0,25	0,2	0,25	0,2	0,5	0,2	0,25
Preserve	0,25	0,2	0,25	0,25	0,25	0,2	1	0,2	0,2	0,2	0,2	0,25	0,25	0,1667	0,25	0,2	0,25	0,2	0,25	0,2	0,25
Promise	0,25	0,3333	0,25	0,25	0,25	0,2	0,2	1	0,2	0,2	0,25	0,25	0,25	0,1667	0,25	0,2	0,25	0,2	0,25	0,2	0,25
Put	0,3333	0,2	0,5	0,5	0,25	0,2	0,2	0,2	1	0,2	0,25	0,25	0,25	0,1667	0,25	0,3333	0,3333	0,25	0,3333	0,3333	0,25
React	0,3333	0,2	0,25	0,5	0,25	0,2	0,2	0,2	0,2	1	0,2	0,25	0,25	0,1667	0,25	0,2	0,25	0,2	0,25	0,25	0,25
Reason	0,25	0,25	0,25	0,25	0,25	0,3333	0,2	0,25	0,25	0,2	1	0,3333	0,25	0,25	0,25	0,2	0,3333	0,3333	0,5	0,2	0,25
Reflect	0,3333	0,25	0,3333	0,3333	0,3333	0,3333	0,25	0,25	0,25	0,25	0,3333	1	0,3333	0,3333	0,3333	0,25	0,3333	0,25	0,5	0,25	0,3333
Remove	0,3333	0,25	0,3333	0,3333	0,3333	0,25	0,25	0,25	0,25	0,25	0,25	0,3333	1	0,2	0,3333	0,25	0,3333	0,25	0,3333	0,25	0,3333
Schedule	0,2	0,2	0,2	0,2	0,2	0,5	0,1667	0,1667	0,1667	0,1667	0,25	0,25	0,2	1	0,2	0,1667	0,2	0,1667	0,3333	0,1667	0,2
Search	0,3333	0,25	0,3333	0,3333	0,3333	0,25	0,25	0,25	0,25	0,25	0,25	0,3333	0,3333	0,2	1	0,25	0,3333	0,25	0,3333	0,25	0,3333
Send	0,25	0,2	0,25	0,5	0,25	0,2	0,2	0,2	0,3333	0,2	0,2	0,25	0,25	0,1667	0,25	1	0,3333	0,2	0,25	0,2	0,25
Separate	0,3333	0,25	0,3333	0,5	0,3333	0,25	0,25	0,25	0,3333	0,25	0,3333	0,3333	0,3333	0,2	0,3333	0,3333	1	0,25	0,3333	0,3333	0,3333
Solve	0,25	0,2	0,25	0,25	0,25	0,2	0,2	0,2	0,25	0,2	0,3333	0,25	0,25	0,1667	0,25	0,2	0,25	1	0,25	0,2	0,5
Think	0,3333	0,3333	0,5	0,3333	0,3333	0,5	0,25	0,25	0,3333	0,25	0,5	0,5	0,3333	0,3333	0,3333	0,25	0,3333	0,25	1	0,3333	0,3333
Transform	0,25	0,25	0,5	0,3333	0,25	0,2	0,2	0,2	0,3333	0,25	0,2	0,25	0,25	0,1667	0,25	0,2	0,3333	0,2	0,3333	1	0,25
Understand	0,3333	0,25	0,3333	0,3333	0,5	0,25	0,25	0,25	0,25	0,25	0,25	0,3333	0,3333	0,2	0,3333	0,25	0,3333	0,5	0,3333	0,25	1

CHAPTER 4

Evaluation of DREQUS

This chapter presents the results of two Empirical Studies (ES1 and ES2) aimed at validating the performance of DREQUS concerning the research questions. The ES1 is intended to evaluate DREQUS in comparison to a non-assisted requirements discovery process; in the meantime, the ES2 evaluates DREQUS in comparison to Brainstorming

4.1. Discovery of REQuirements Using Scenarios (DREQUS): Results of an Empirical Study

4.1.1. Introduction

The Requirements Elicitation (RE) phase is one of the most important stages in the development of an information system. One of the main challenges of the RE is to ensure that the system requirements are consistent with the needs of the organization where it will be used [1]. Consequently, much effort has been devoted to developing approaches and tools to assist the requirements engineers in this critical task of the development process [2], [3], [4]. Nevertheless, due to the complexity of the process, there are still challenges that remain as priorities for the RE researchers. In this context, we have proposed a RE approach named Discovery of REQuirements Using Scenarios (DREQUS); in the previous chapter, the mechanisms and methods of DREQUS were detailed. This chapter aims at describing the results of an Empirical Study (ES) conducted to compare the performance of a RE process using DREQUS vs. a non-assisted RE process. More specifically, we are interested in comparing the quality of these RE processes in terms of completeness (recall), precision, and over-specification. We also asked for the participants' feedback to obtain a qualitative assessment of the DREQUS performance.

This chapter is organized as follows: Section 4.1.2. presents the planning and design of the empirical study; Section 4.1.3. deals with the study results; Section 4.1.4. analyzes the threats to the validity and the actions taken to mitigate them; Section 4.1.5. reviews related works and Section 4.1.6. considers the ES conclusions.

4.1.2. Planning and Design of the Empirical Study

The components of this ES are presented in the sequel:

4.1.2.1. Research questions

We are interested in answer the next questions:

Q_1: ¿Does DREQUS have a significant impact on the elicitation of requirements in comparison to a non-assisted process?

Q_2: ¿Does DREQUS aid to the knowledge acquisition of one unexplored domain and the training of requirements engineers?

4.1.2.2. Hypotheses

In order to answer the research questions, we have defined the next hypotheses:

TABLE 1. RESEARCH QUESTIONS AND HYPOTHESES.

Research question	**Hypotheses**
Q_1	H_1: requirements elicited using DREQUS are significantly more complete than those elicited with a non-assisted process.
	H_2: requirements elicited using DREQUS are significantly more precise than those elicited with a non-assisted process.
	H_3: requirements elicited using DREQUS have significantly less over-specification than those elicited with a non-assisted process.
Q_2	H_4: DREQUS aids to the knowledge acquisition of one unexplored domain and the training of requirements engineers.

4.1.2.3. Participants and roles

Considering the different obstacles inherent to industrial studies [8] and the validity of software engineering studies involving students instead of professional software developers [9], we proposed to conduct this study at the University of Antioquia (Colombia). The participants were undergraduate students of RE and the two professors of this course at this institution; their roles are described in Table 2.

TABLE 2. PARTICIPANTS AND THEIR ROLES.

Participant	Role
RE students (Group 1)	Using DREQUS, they elicited the requirements for a new Meeting Scheduler system. This task was carried out by teams consisted of two students each.
RE students (Group 2)	They elicited, without the support DREQUS or any other approach, the requirements for a new Meeting Scheduler system. This task was carried out by teams consisted of two students each.
RE professors.	- In the context of the RE course, they have trained both Groups in RE fundamentals. - They also trained Group 1 on the use of DREQUS. - They assessed the quality of the requirements obtained by both groups of students.

The RE students were homogeneously distributed in both groups; to this aim, we used a pre-test questionnaire asking for their RE experience. All the subjects (students and professors) volunteered their services and received no financial reward.

4.1.2.4. Requirements exemplar

After considering well known and accepted requirements exemplars like “The library” and “Router package” among others, we propose the use of the “Meeting Scheduler” exemplar [10]. We have chosen this exemplar for the next reasons [8], [11]:

• It arises out of a real problem.

• The domain of expertise is accessible.

• It covers many interesting specification issues, for example, complex concurrency and distribution aspects, real-time performance constraints, non-functional requirements such as privacy, usability, and flexibility, optimization requirements, etc.

• It is implementable and easy to validate without an enormous outpouring of implementation resources.

• It is representative of an interesting set of distributed groupware systems.

• We count with a reliable and recent published list of requirements (a benchmark) of this exemplar elaborated for Cleland-Huang [12], [13]. This resource is available at PROMISE repository (http://openscience.us/repo/).

4.1.2.5. Variables

Based on Pastor et al. [14], we identified the variables corresponding to quality indicators of a Software Requirements Specification (SRS). In the assessment of requirements elicitation approaches, the variables completeness (recall), precision and over-specification are widely accepted by the Requirements Engineering community. These quality indicators were calculated as specified by Harmain and Gaizauskas [15]:

Recall (completeness) = NC / Nkey
Precision = NC / (NC + NE)
Over-specification = NE / NKey

Where:
NC: number of requirements coinciding with the reference solution (exemplar).
NKey: number of requirements in the reference solution.
NE: number of requirements not coinciding with the reference solution.

4.1.2.6. The experimental task

The empirical study was designed aiming to verify the performance of the DREQUS approach. To achieve this goal, in a similar manner to [16], we compared DREQUS with a non-assisted process which consists in eliciting software requirements without any support from another technique or a specific tool.

In a preliminary session, using a pre-test questionnaire, we established that all students count with only academic experience in RE obtained in this course; the student's age ranges from 19 to 29 years old. After, we formed, in a random manner, Groups 1 and 2; then into each of groups, once again in a random manner, we formed teams of two individuals each. The results of this process are condensed in table 3:

TABLE 3. CHARACTERISTICS OF GROUPS 1 AND 2.

Group	Men	Women	Number of teams	Student´s experience in RE
1	18	2	10	Academic
2	16	2	9	

In an initial stage (stage 0), the RE professors trained Group 1 on the use of DREQUS (this took two sessions of two hours each). In a posterior session of four hours, Groups 1 and 2 were required to elicit the requirements for a distributed Meeting Scheduler system; Group 1 worked with DREQUS assistance, and Group 2 without the DREQUS support. The Meeting Scheduler domain description which was delivered to the groups appears in [8]. Finally, taken as a benchmark the software requirements specification proposed by Cleland-Huang [12], the RE professors computed and compared the results of the approaches in terms of completeness, precision, and over-specification. Figures 1 and 2 illustrate the whole process.

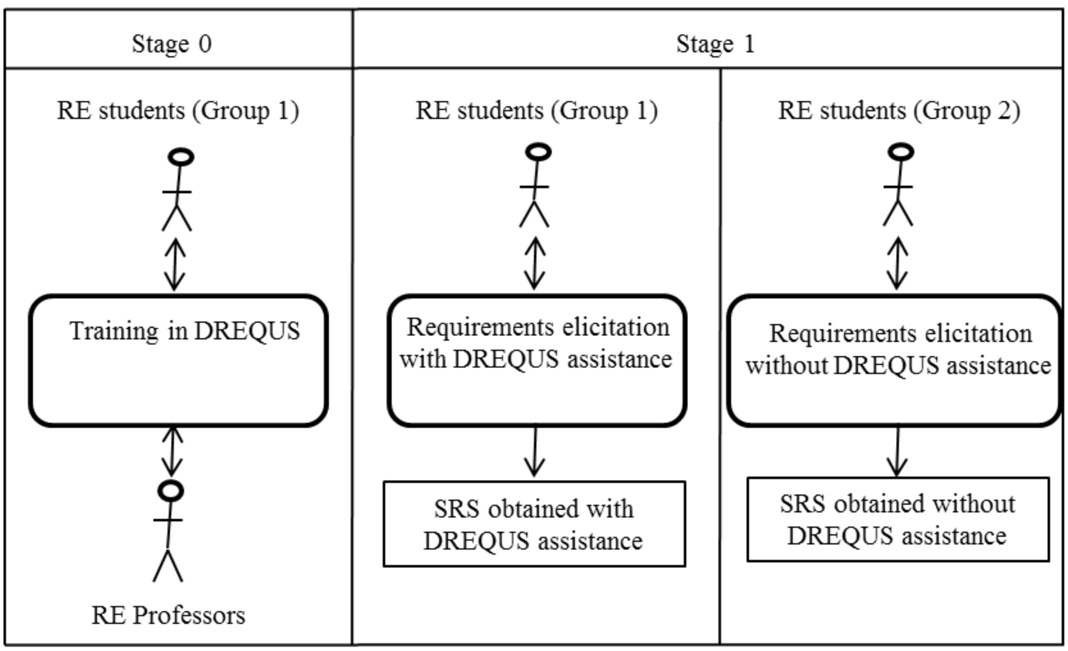


Figure 1. The Meeting Scheduler requirements elicitation process.

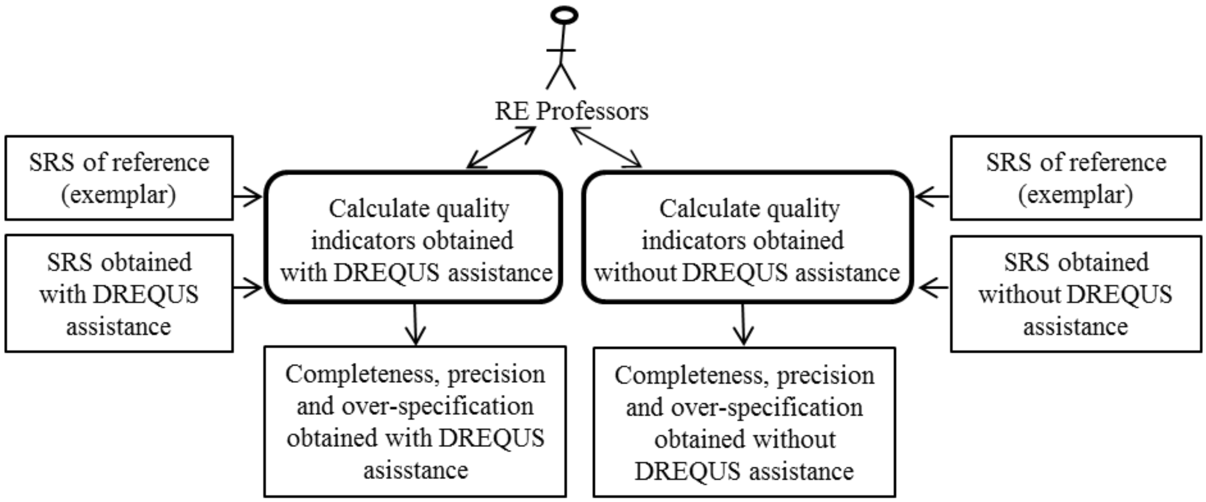


Figure 2. Measurement process of the SRSs quality.

After the quantitative analysis, based on Oliveira et al. [16], Kasunic [17] and the TAM model [18], [19], we conducted a post-task survey with Group 1 aimed at measuring the perceived usefulness and ease of use provided by DREQUS:

(i) Perceived usefulness, which defines the degree in which a person believes that the technology could improve his/her performance at work, and
(ii) Perceived ease of use, which defines the degree in which a person believes that using a specific technology would be effortless [20].

According to Davis [18], these aspects are strongly correlated with user acceptance.

4.1.2.7. Instruments

The used instruments include a pre-test questionnaire which is based on the work of Ben Achour et al. [21], the experimental object corresponding to the problem statement and domain description [8], training materials in DREQUS, a software requirement specification template used to condense the results, and a post-task survey (a 5-points Likert scale) which was solved by the RE students of Group 1.

4.1.3. Empirical Study Results

Using the exemplar proposed by Cleland-Huang [12], [13], the professors assessed the requirements specifications delivered by the teams and calculated, for each of them, the quality indicators: recall, precision, and over-specification; they also consolidated the post-task survey results. These outcomes are summarized in Tables 4, 5 and 11. This information was used to validate the research hypotheses as is presented in the sequel.

TABLE 4. GROUPS PERFORMANCE

GROUP 1: ASSISTED BY DREQUS.											
	S1	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**Mean**
#Reqs.	40	40	37	39	40	41	47	47	49	47	-
NC	35	37	35	38	38	39	47	46	48	42	-
NE	5	3	2	1	2	2	0	1	1	5	-
Recall	64,8%	68,5%	64,8%	70,4%	70,4%	72,2%	87,0%	85,2%	88,9%	77,8%	75,0%
Precision	87,5%	92,5%	94,6%	97,4%	95,0%	95,1%	100,0%	97,9%	98,0%	89,4%	94,7%
Over-spec.	9,3%	5,6%	3,7%	1,9%	3,7%	3,7%	0,0%	1,9%	1,9%	9,3%	4,1%
GROUP 2: NON-ASSISTED BY DREQUS (NAD).											
	S11	**S12**	**S13**	**S14**	**S15**	**S16**	**S17**	**S18**	**S19**	**Mean**	
#Reqs.	24	30	26	27	48	21	48	37	42	-	
NC	14	15	18	17	28	11	31	29	21	-	
NE	10	15	8	10	20	10	17	8	21	-	
Recall	25,9%	27,8%	33,3%	31,5%	51,9%	20,4%	57,4%	53,7%	38,9%	37,9%	
Precision	58,3%	50,0%	69,2%	63,0%	58,3%	52,4%	64,6%	78,4%	50,0%	60,5%	
Over-spec.	18,5%	27,8%	14,8%	18,5%	37,0%	18,5%	31,5%	14,8%	38,9%	24,5%	
Legend: S = specification of requirements, #Reqs = number of requirements, NC = number of requirements coinciding with the reference solution, NE = number of requirements not coinciding with the reference solution											

* NKey = 57.

TABLE 5. SUMMARY OF GROUPS PERFORMANCE.

	Group 1 (mean)	**Group 2 (mean)**
Recall	75,0%	37,9%
Precision	94,7%	60,5%
Over-specification	4,1%	24,5%

The relations between the research questions, the hypotheses and the means of validation are summarized in Table 6.

TABLE 6. RESEARCH QUESTIONS, HYPOTHESES AND MEANS OF VALIDATION.

Question	Hypotheses	Mean of validation
Q_1	H_1, H_2 and H_3	Quality indicators of the SRSs: recall (completeness), precision and over-specification).
Q_2	H_4	A survey to be answered by the RE students of Group 1.

In order to answer Q_1, we validated H_1, H_2, and H_3. To carry out this task, we selected the T-Student test. This test is used to evaluate whether the means of two groups significantly differ from each other. The T-Student test is most commonly applied when an experiment uses a small sample size ($n < 30$). In our case, we have obtained 10 and 9 specifications from Group 1 and Group 2 respectively, in both cases $n < 30$, therefore the T-Student test is suitable to establish the acceptance or rejection of our hypotheses. In order to be applied, the T-Student test requires the compliance of the following conditions:

- Independence: the samples must be independent.
- Normality: it must be verified that the random variable in both groups is normally distributed.
- Equity of variances: it must be verified the equity of variances between the two groups.

As we have previously described, the requirements specifications of Groups 1 and 2 were obtained independently from each other. In this sense, we complied with the independence condition. Using SPSS [22] the normality, equity of variances and T-Student tests were conducted for each variable: recall, precision, and over-specification, as follows (as is widely accepted in empirical studies we worked with $\alpha = 0.05$):

H_{1o}: requirements elicited using DREQUS are not significantly more complete than those elicited with a non-assisted process.

H_1: requirements elicited using DREQUS are significantly more complete than those elicited with a non-assisted process.

Table 7 presents the results obtained after applying the tests to the “recall” variable.

TABLE 7. TESTS FOR RECALL: NORMALITY TEST, TEST OF EQUITY OF VARIANCES AND T-STUDENT TEST.

Normality test (Shapiro-Wilk test; n < 30)			
Recall	P-value (Group 1) = 0.130	>	α = 0.05
	P-value (Group 2) = 0.342	>	α = 0.05
Conclusion: P-value (Groups 1 and 2) > α = 0.05 => the data come from normal distributions. The "recall" variable, in both groups, behaves normally.			
Test of equality of variances (Levene´s test)			
P-value = 0.168		>	α = 0.05
Conclusion: P-value > α = 0.05 => There is no difference between the variances.			
T-student test			
P-value = 0.000002		<	α = 0.05
Conclusion: P-value < α = 0.05 => H1 is accepted: *Requirements elicited using* DREQUS *are significantly more complete than those elicited with a non-assisted process.*			

Graphically:

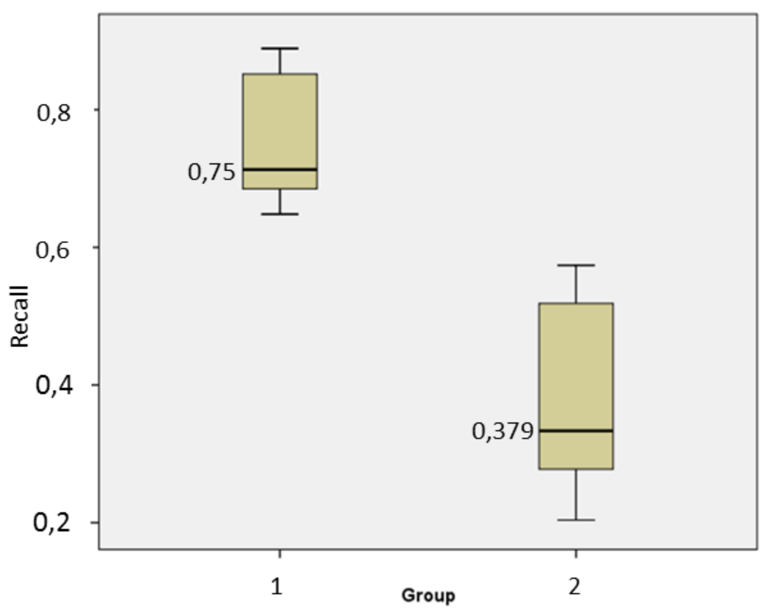


Figure 3. Recall (completeness) of Groups 1 and 2.

H_{2o}: requirements elicited using DREQUS are not significantly more precise than those elicited with a non-assisted process.

H_2: requirements elicited using DREQUS are significantly more precise than those elicited with a non-assisted process.

Table 8 summarizes the results obtained after applying the tests to the “precision” variable.

TABLE 8. TESTS FOR PRECISION: NORMALITY TEST, TEST OF EQUITY OF VARIANCES AND T-STUDENT TEST.

Normality test (Shapiro-Wilk test; n < 30)			
Preci-sion	P-value (Group 1) = 0.489	>	α = 0.05
	P-value (Group 2) = 0.447	>	α = 0.05
Conclusion: P-value (Groups 1 and 2) => the data come from normal distributions. The “precision” variable, in both groups, behaves normally.			
Test of equality of variances (Levene´s test)			
P-value = 0.122		>	α = 0.05
Conclusion: P-value > α = 0.05 => There is no difference between the variances.			
T-student test			
P-value = 0.000003		<	α = 0.05
Conclusion: P-value < α = 0.05 => H2 is accepted: *Requirements elicited using DREQUS are significantly more precise than those elicited with a non-assisted process.*			

Graphically:

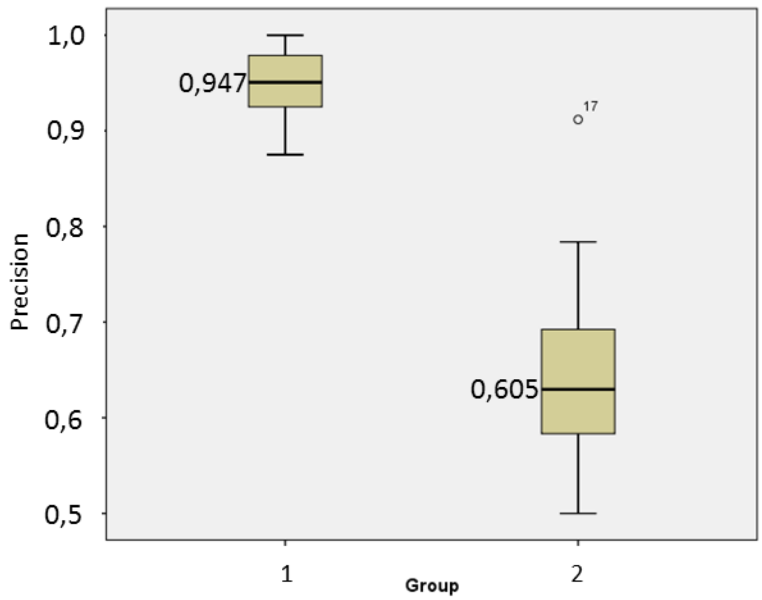


Figure 4. Precision of Groups 1 and 2.

H_{3o}: requirements elicited using DREQUS have not significantly less over-specification than those elicited with a non-assisted process.

H_3: requirements elicited using DREQUS have significantly less over-specification than those elicited with a non-assisted process.

Table 9 condenses the results obtained after applying the tests to the “over-specification” variable.

TABLE 9. TESTS FOR OVER-SPECIFICATION: NORMALITY TEST, TEST OF EQUITY OF VARIANCES AND T-STUDENT TEST.

Normality test (Shapiro-Wilk test; n < 30)			
Over-specification	P-value (Group 1) = 0.118	>	α = 0.05
	P-value (Group 2) = 0.091	>	α = 0.05
Conclusion: P-value (Groups 1 and 2) => the data come from normal distributions. The “over-specification” variable, in both groups, behaves normally.			
Test of equality of variances (Levene´s test)			

P-value = 0.142	>	α = 0.05
Conclusion: P-value > α = 0.05 => There is no difference between the variances.		
T-student test		
P-value = 0.000006	<	α = 0.05
Conclusion: P-value < α = 0.05 => H3 is accepted: *Requirements elicited using DREQUS have significantly less over-specification than those elicited with a non-assisted process.*		

Graphically:

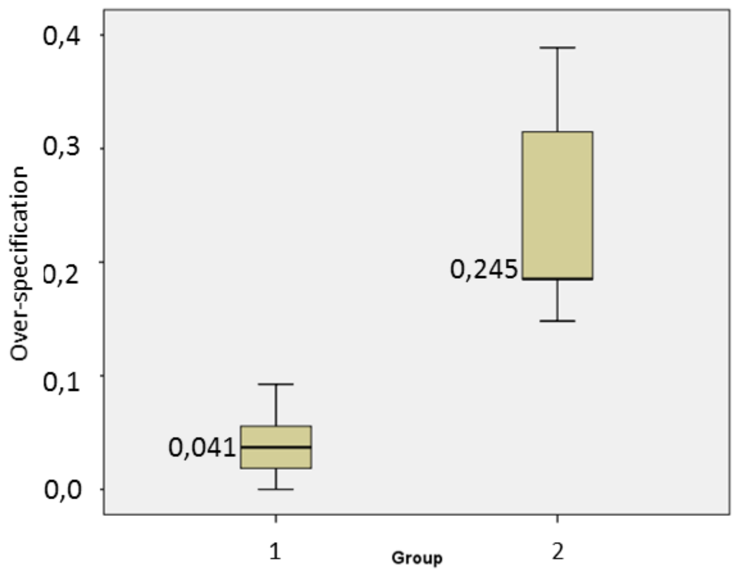


Figure 5. Over-specification of Groups 1 and 2.

Moreover, considering the question Q_2 and its respective hypothesis:

H_4: DREQUS aids to the knowledge acquisition of one unexplored domain and the training of requirements engineers.

We have validated H_4 by means of a post-task survey answered by Group 1. Using a 5-points Likert scale (where: 1 = strongly disagree, 2 = disagree, 3 = neutral, 4 = agree and 5 = strongly agree) the individuals rated the next assertions (Table 10):

TABLE 10. POST-TASK SURVEY ANSWERED BY GROUP 1

Q. Id.	Question
1	¿Does DREQUS facilitate the discovery of functional requirements "hidden" of a system?
2	¿Does DREQUS facilitate the discovery of non-functional requirements "hidden" of a system?
3	¿Does DREQUS improve the correctness of the elicited functional requirements?
4	¿Does DREQUS improve the correctness of the elicited non-functional requirements?
5	¿Does DREQUS improve the completeness of the elicited functional requirements?
6	¿Does DREQUS improve the completeness of the elicited non-functional requirements?
7	¿Does DREQUS, in general, improve the quality of the elicited requirements?
8	¿Is DREQUS easy to learn?
9	¿Is DREQUS easy to be applied?
10	¿Does DREQUS reduce the time invested in the requirements elicitation process?
11	¿Does DREQUS aid to the knowledge acquisition of a dominion unknown for the requirements engineer?
12	¿Does DREQUS increased my knowledge of the requirements elicitation process?
13	¿Does DREQUS aid to the requirements engineers training?
14	¿Does DREQUS help to improve my performance as a requirements engineer?
15	¿Is DREQUS useful for the development of other systems?
16	¿Is DREQUS is recommendable to be used by other requirements engineers?

Tables 11, 12 and Figure. 6 summarize the survey results:

TABLE 11. ANSWERS OF GROUP 1 TO THE POST-TASK SURVEY.

Q. Id.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	Mean
1	5	3	5	5	3	5	4	4	4	4	4	4	3	4	4	5	4	4	5	4	4,2
2	3	3	5	3	3	5	3	4	3	4	4	4	3	3	4	5	5	4	5	4	3,9
3	5	5	4	5	4	4	5	5	3	5	5	4	4	4	4	4	5	5	4	5	4,5
4	3	5	4	4	4	4	4	4	4	5	4	4	4	3	4	4	5	5	4	5	4,2
5	4	4	4	4	4	5	5	5	4	3	4	3	4	5	4	5	5	5	4	5	4,3
6	3	4	4	3	4	5	3	4	5	3	4	3	3	3	4	5	4	4	4	4	3,8
7	4	5	3	5	5	3	5	5	4	5	4	3	3	4	4	4	3	4	3	4	4,0
8	5	4	1	5	4	4	1	3	4	4	4	4	5	4	1	4	4	3	1	3	3,4
9	4	4	5	4	4	4	5	5	3	4	3	4	5	4	1	3	1	5	5	5	3,9
10	3	1	4	5	1	1	5	3	4	3	4	1	5	3	5	5	5	3	4	3	3,4
11	4	3	5	3	3	4	3	4	4	4	4	3	3	4	5	5	5	4	5	4	4,0
12	5	4	5	5	5	5	5	4	4	4	4	5	4	5	4	4	4	4	5	4	4,5
13	4	4	5	4	4	4	3	5	4	4	5	4	4	4	4	5	4	5	5	5	4,3
14	4	4	5	5	4	4	4	3	4	4	4	4	5	4	4	4	4	5	5	5	4,3
15	4	4	5	3	3	3	4	5	4	3	4	4	5	3	4	4	3	4	5	4	3,9
16	5	4	5	4	4	3	3	4	4	4	4	3	3	4	4	4	3	4	5	4	3,9
Legend: P = participant, Q. Id. = question identifier, 1 = strongly disagree , 2 = disagree, 3 = neutral, 4 = agree , 5 = strongly agree.																					

Note: yellow cells correspond to aspects with the lowest ratings.

TABLE 12. MEANS RELATED TO THE SURVEY ANSWERS.

Assertion: DREQUS:	Mean
1. Facilitates the discovery of functional requirements "hidden" of a system.	4,2
2. Facilitates the discovery of non-functional requirements "hidden" of a system.	3,9
3. Improves the correctness of the elicited functional requirements.	4,5
4. Improves the correctness of the elicited non-functional requirements.	4,2
5. Improves the completeness of the elicited functional requirements.	4,3
6. Improves the completeness of the elicited non-functional requirements.	3,8
7. In general, improves the quality of the elicited requirements.	4,0
8. Is easy to learn.	3,4
9. Is easy to apply.	3,9
10. Reduces the time invested in the requirements elicitation process.	3,4
11. Aids to the knowledge acquisition of a dominion unknown for the requirements engineer.	4,0
12. Has increased my knowledge of the requirements elicitation process.	4,5
13. Aids to the requirements engineers training.	4,3
14. Helps to improve my performance as a requirements engineer.	4,3
15. Is useful for the development of other systems.	3,9
16. Is recommendable to be used by other requirements engineers.	3,9

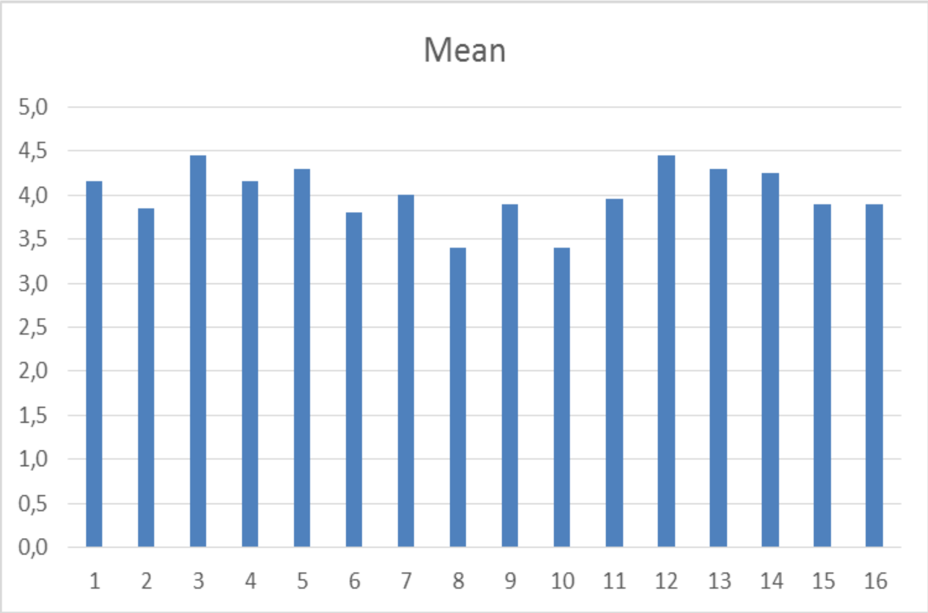


Figure 6. Means related to the survey answers.

Taking into account the hypothesis of interest, H_4, we have established a relation between the hypotheses aspects to be validated and the assertions that allow to measuring each of the required aspects (Table 13):

TABLE 13. ASPECTS TO EVALUATE H_4 AND ASSERTIONS THAT EVALUATE EACH ASPECT.

Aspect to evaluate	Assertions that evaluate the aspect
DREQUS aids to the knowledge acquisition of one unexplored domain.	1, 2, 3, 4, 5, 6, 7, and 11.
DREQUS aids to the training of requirements engineers.	12, 13, and 14.
DREQUS is useful.	15, and 16.
DREQUS is easy to use.	8, 9, and 10.

Considering the Tables 12, 13 and the hypothesis H_4, DREQUS aids to the knowledge acquisition of an unexplored domain and the training of requirements engineers, we observed that the students of Group 1 agree to the assertions: 1-7, 11, 15 and 16 which correspond to "DREQUS aids to the knowledge acquisition of one unexplored domain" and they also agree to the assertions: 12, 13 and 14 which correspond to "DREQUS aids to the training of requirements engineers". Therefore, the hypothesis H_4 is accepted.

Conclusion: DREQUS aids in the knowledge acquisition of one unexplored domain and the training of requirements engineers.

Note: the answers to the assertions 8 and 10 which correspond to the easiness of use indicate possible drawbacks that must be assessed in future works to arrive at pertinent conclusions.

4.1.4. Threats to Validity

Considering possible threats to the validity of our empirical study we have defined methodological aspects to mitigate them [23], [24]:

The *construct validity* refers to how well the studied parameters and their outcomes are adequate to the research questions addressed. In this sense, we have selected a set of quality indicators aimed at measuring the performance of the RE processes in comparison; the selected indicators: recall, precision, over-specification, usefulness, and ease of use are widely accepted by the Requirements Engineering community [14], [26]. Likewise, the quality indicators measures were obtained utilizing an exemplar proposed in [12] and a post-task survey; these results allowed the hypotheses contrast through T-tests and the survey analysis which are frequently used by the community in similar studies [21], [25].

The *internal validity* concerns factors that influencing the independent variable. To mitigate these factors, we have evaluated the students´ experience in RE; according to these results, the subjects were randomly assigned to the Groups and teams. We also have assigned different classrooms and the same duration for the task execution; besides, all the teams' results were assessed, with the same instruments, by independent professors.

The *conclusion validity* concerns the relations between the treatments and the outcomes. In our study, we have used the Shapiro-Wilk test to assess the normality of the variables (we used this test due to its suitability when the size of the sample is < 30), the Levene´s test to evaluate the equality of variances and T-Test to the hypotheses validation. All these tests are quite robust to contrast the quantitative hypotheses in a reliable manner. Besides, one of the hypotheses was evaluated using a post-task survey in which respondents provided feedback on their perceptions about the DREQUS approach. This type of qualitative analysis is frequently used by the Requirements Engineering community and contributes when other types of assessments are difficult to be carried out, e.g. [17], [24], [25].

The *external validity* refers to the ability to generalize the research findings obtained to other domains under different settings [24]. Regarding this aspect, this empirical study presents preliminary results of the DREQUS performance in comparison to a non-assisted process; more studies are required to make generalizations in other contexts (e.g. in comparison to other RE approaches).

4.1.5. Related Work

Threats to the validity of our empirical study were mitigated following considerations proposed in Kitchenham et al. [23] and Feldt and Magazinius [24]. Empirical studies in the field of Requirements Engineering are described in [25-28], [32-33]. In particular, empirical studies to evaluate requirements elicitation approaches can be found at [16], [29], [30]. Meanwhile, Davis et al. [31] report a systematic review of empirical studies concerning the effectiveness of elicitation techniques.

4.1.6. Conclusions

We have conducted an empirical study to assess the effectiveness and acceptability of a proposed approach for the Discovery of REQuirements Using Scenarios (DREQUS). In this study, two groups were required to elicit the requirements for a Meeting Scheduler system; Group 1 was assisted by DREQUS and Group 2 worked without the assistance of any particular technique or tool. Once finished the Requirements Elicitation (RE) task, we have performed a quantitative and qualitative comparison of the RE processes carried out by both groups. The results of this process led to the following findings:

- The requirements elicited using DREQUS are significantly more complete than those elicited with a non-assisted process.
- The requirements elicited using DREQUS are significantly more precise than those elicited with a non-assisted process.
- The requirements elicited using DREQUS have significantly less over-specification than those elicited with a non-assisted process.
- DREQUS aids in the knowledge acquisition of one unexplored domain and the training of requirements engineers.

These results indicate that, in comparison to a non-assisted process, the use of DREQUS improves the quality of the elicited requirements,. Besides, they also indicate that DREQUS is useful for the knowledge acquisition of one unexplored domain and the training of requirements engineers.

As stated in section 4.1.4, more studies are required to make generalizations in other contexts (e.g. in comparison to other RE approaches). Considering this, the next chapter presents an ES aimed to compare the performance of DREQUS in comparison to Brainstorming.

References

[1] O. Pastor, I. Villanueva and J. Sánchez. Elicitación de requisitos en sistemas de gestión orientados a procesos. Proceedings of the 11th International Workshop on Requirements Engineering (WER´ 05). Pp. 38-49. Porto, Portugal. 2005.

[2] A. Jaramillo. Requirements elicitation approaches: a systematic review. Research Challenges in Information Science (RCIS), IEEE 9th International Conference on. Athens. Greece. 2015.

[3] S. Easterbrook and B. Nuseibeh Bashar. "Requirements Engineering: A Roadmap". Proceedings of the Conference the Future of Software Engineering. ICSE 00. Limerick Ireland 2000.

[4] B. Cheng and J. Atlee Joann. Research Directions in Requirements Engineering. Future of Software Engineering (FOSE'07). 2007.

[5] C. Rolland and C. Salinesi. Supporting requirements elicitation through goal/scenario coupling. Conceptual Modeling: Foundations and Applications, LNCS. Vol. 5600. Pp. 398-416. Springer. 2009.

[6] J. Mendling, J. Recker, H. Reijers. On the usage of labels and icons in business process modeling. International Journal of Information System Modeling and Design, 1(2), 40-58. 2010.

[7] T. Metzler, K. Shea. Taxonomy of cognitive functions. Proceedings of the 18th International Conference on Engineering Design. ICED11. 2011.

[8] A. van Lamsweerde, M. S. Feather, S. Fickas, and A. Finkelstein “Requirements and specification exemplars,” Automated Software. Engineering, vol. 4, no. 4, pp. 419–438, 1997.

[9] B. Regnell, M. Höst and C. Wohlin. Using students as subjects — A comparative study of students and professionals in lead-time impact assessment. Empirical Software Engineering, 5, 201–214. Kluwer Academic Publishers, Boston. Manufactured in The Netherlands. 2000.

[10] A. van Lamsweerde, R. Darimont and P. Massonet. Goal-directed elaboration of requirements for a meeting scheduler: problems and lessons learnt. Proceedings of the Second IEEE International Symposium on Requirements Engineering, 1995.

[11] E. Yu and L. M. Cysneiros. Agent-Oriented methodologies – Towards a challenge exemplar. Faculty of Information Studies. Department of Computer Science. University of Toronto. Canada. 2001.

[12] J. Cleland-Huang et al. http://promisedata.googlecode.com/svn/trunk/text/nfr/nfr.arff. 2007.

[13] J. Mylopoulos, G. Guizzardi. J. Horkoff, F. Li, A. Borgida, and L. Liu. From stakeholder requirements to formal specifications through refinement. Requirements Engineering Foundation for Software Quality. REFSQ 2015.

[14] Ó. Pastor, A. González, N. Condori-Fernández and S. España. An empirical comparative evaluation of requirements engineering methods. Brazilian Computer Society. 2010.

[15] H. Harmain and R. Gaizauskas. CM-Builder: An automated NL-Based CASE Tool. Proceedings of the 15th IEEE International Conference on Automated Software Engineering (ASE´00). Grenoble. 2000.

[16] M. Oliveira, D. Viana, T. Conte, S. Vieira and S. Marczak. Evaluating the REMO-EKD technique: a technique for the elicitation of software requirements base don EKD organizational models. Empirical Requirements Engineering (EmpiRE). IEEE Third International Workshop on. P. 9-16. 2013.

[17] M. Kasunic. Designing an effective survey. Carnegie Mellon. Software Engineering Institute. Pittsburgh PA. 2005.

[18] F. Davis, Perceived usefulness, perceived ease of use, and user acceptance of information technology. In: MIS Quarterly, v. 13, n. 3, 1989.

[19] V. Venkatesh, M. Morris, G. Davis and F. Davis, User acceptance of information technology: Toward a unified view, MIS Quarterly, v. 27, n. 3, pp. 425-478. 2003

[20] O. Laitenberger and H. Dreyer, Evaluating the Usefulness and the Easeof Use of a Web-based Inspection Data Collection Tool, in: Proc. X Software Metrics Symposium, Bethesda - MD, pp. 122-132. 1998.

[21] C. Ben Achour, N.A.M. Maiden. Empirical Study of Use Case Authoring: Experimental Material. Internal Report, Centre de Recherche en Informatique, Université de Paris 1 – Sorbonne. 1999.

[22] IBM SPSS Statistics, http://www-01.ibm.com/software/analytics/spss/products/statistics/.

[23] B. Kitchenham, T. Dyba, and M. Jorgensen, "Evidence-based software engineering," in Proc. of the 26th International Conference on Software Engineering (ICSE04), 2004, pp. 273–281. [Online]. Available: http://dl.acm.org/citation.cfm?id=998675.999432

[24] R. Feldt and A. Magazinius. Validity Threats in Empirical Software Engineering Research-An Initial Survey. In Proceedings of the 22nd International Conference on Software Engineering and Knowledge Engineering, pp. 374-379. 2010.

[25] M. Mahaux, L. Nguyen, L. Mich, A. Mavin. A framework for understanding collaborative creativity in requirements engineering: Empirical validation. Empirical Requirements Engineering (EmpiRE), IEEE Fourth International Workshop on. 2014.

[26] M. Morandini, A. Marchetto, A. Perini. Requirements comprehension: A controlled experiment on conceptual modeling methods. Empirical Requirements Engineering (EmpiRE), 2011 First International Workshop on. 2011.

[27] K. Labunets, F. Paci, F. Massacci, R. Ruprai. An experiment on comparing textual vs. visual industrial methods for security risk assessment. Empirical Requirements Engineering (EmpiRE), 2014 IEEE Fourth International Workshop on. 2014.

[28] M. Daneva, D. Damian, A. Marchetto, and O. Pastor. Empirical research methodologies and studies in Requirements Engineering: How far did we come? The Journal of Systems and Software 95. 1–9. 2014.

[29] S. Vieira, D. Viana, R. Nascimento, T. Conte. Using Empirical Studies to evaluate the REMO Requirement Elicitation Technique. Proc. 24th International Conference on Software Engineering and Knowledge Engineering, SEKE. Redwood City, CA, USA, 2012, pp. 33-38. 2012.

[30] N. Ikram, S. Siddiqui, N. F. Khan. Security requirement elicitation techniques: The comparison of misuse cases and issue based information systems. Empirical Requirements Engineering (EmpiRE), 2014 IEEE Fourth International Workshop on. 2014.

[31] A. Davis, O. Dieste, A. Hickey, N. Juristo, A. M. Moreno. Effectiveness of Requirements Elicitation Techniques: Empirical Results Derived from a Systematic Review. Requirements Engineering, 14th IEEE International Conference. 2006.

[32] N. Condori-Fernandez, M. Daneva, K. Sikkel, R. Wieringa, O. Dieste, O. Pastor. A systematic mapping study on empirical evaluation of software requirements specifications techniques. Empirical Software Engineering and Measurement, 2009. ESEM 2009. 3rd International Symposium on. 2009.

[33] S. Besrour, L. Bin Ab Rahim, P. D. D. Dominic. Assessment and evaluation of requirements elicitation techniques using analysis determination requirements framework. Computer and Information Sciences (ICCOINS), 2014 International Conference on. 2014.

[34] A. Jaramillo. Non-functional requirements elicitation from business process models. Fifth IEEE International Conference on Research Challenges in Information Science, May 19-21 2011, Guadeloupe - French West Indies, France. 2011.

4.2. Brainstorming versus a Scenario-based Approach: Results of an Empirical Study

4.2.1. Introduction

Recently, in the Requirements Elicitation (RE) domain, a line of works that recognizes the importance of creativity in the requirements definition process has emerged. Creativity is “the ability to produce work that is both novel (i.e. original, unexpected) and appropriate (i.e. useful, adaptive concerning task and constraints)” [1-3]. An idea is creative when it brings a new insight into a given situation. The process of creativity includes the ability to change one's approach to a problem, to produce ideas that are both relevant and unusual, to see beyond the immediate situation, and to redefine the problem or some aspect of it.

In the field of requirements engineering, several authors have emphasized the importance of treating requirements elicitation as a creative problem-solving process [4]. Indeed, while requirements were traditionally considered to exist in an implicit manner in the mind of stakeholders and the analyst´s job is to capture them, this view is now considered to drastically reduce the scope of the requirements engineering phase. Instead, invention is claimed to be an essential part of the requirements engineering activity, and “requirements analysts are ideally placed to innovate” [5].

Under the vision of these approaches and motivated by their promises, we initiated a line of works to understand creativity techniques and to leverage them for enhancing the elicitation process. Building on a literature review we conducted recently [6], we have proposed a requirements elicitation approach named DREQUS (Discovery of REQuirements Using Scenarios). DREQUS fosters the stakeholders´ creativity by means of the integration of scenarios and a set of rules and guidelines which facilitate the requirements invention process. This chapter describes the results of an Empirical Study (ES) conducted to compare the performance of DREQUS vs. Brainstorming. More specifically, we are interested in comparing the quality of these approaches in terms of recall (completeness), precision, and over-specification.

The organization of this chapter is as follows: Section 4.2.2. presents the empirical study planning and design; Section 4.2.3. deals with the study results; Section 4.2.4. introduces related works and the threats to the empirical study validity and the actions taken in order to try to mitigate them and Section 4.2.5. considers the ES conclusions.

4.2.2. Empirical Study Planning and Design

The empirical study components are presented in the next paragraphs:

4.2.2.1. Research question: ¿Does DREQUS have a better impact on the elicitation of requirements with respect to a Brainstorming process?

4.2.2.2. Hypotheses: to answer the research question we defined the next hypotheses:

- $H1_0$: Requirements elicited using DREQUS are not significantly more complete than those elicited with a Brainstorming process.
- $H2_0$: Requirements elicited using DREQUS are not significantly more precise than those elicited with a Brainstorming process.
- $H3_0$: Requirements elicited using DREQUS do not significantly present less over-specification than those elicited with a Brainstorming process.

4.2.2.3. Participants and roles: Considering the different obstacles inherent to industrial studies [9] and the validity of software engineering studies involving students instead of professional software developers [10], we conducted the study at the University of Antioquia (Colombia). The participants were undergraduate students of Requirements Engineering and the two professors of this course at this institution. In alignment with the value theory, the student´s performance in the activities of the empirical study (quizzes and software requirements specifications) received the qualification of 10% of the RE course. This also contributed to the professor's motivation. The participant's roles are described in Table 1.

Table 1: Participants and roles.

Participant	Role
Pairs of RE students*	▪ Using Brainstorming and DREQUS, they elicited the requirements for two different systems (Problems 1 and 2)
RE professors.	▪ Studied Brainstorming and DREQUS and trained the pairs on Brainstorming and DREQUS. Besides, they assessed the quality of the requirements obtained by the pairs.
* During the elicitation processes, the students performed both roles: Requirements engineers and users.	

4.2.2.4. Reference solutions: we used the "Meeting Scheduler (MS)" and the "Movie Streaming System" solutions. We have chosen these solutions for the next reasons [9]: They arise out of a real problems, the domains of expertise are accessible, they cover many interesting specification issues (for example, complex concurrency and distribution aspects, non-functional requirements such as privacy, usability, and flexibility, etc.), they are implementable and validatable without an enormous outpouring of implementation resources, they are representatives of an interesting set of distributed groupware systems and we count with a reliable and recent published list of requirements (a benchmark) of these problems elaborated by Cleland-Huang [11].

4.2.2.5. Variables: In the assessment of requirements elicitation approaches the variables recall (completeness), precision and over-specification are widely accepted by the RE community as relevant quality indicators [12]. They were calculated as specified by Harmain and Gaizauskas [13]:

$\text{Recall} = N_{coincidences} / N_{key}$

$\text{Precision} = N_{coincidences} / (N_{coincidences} + N_{extra})$

$\text{Over-specification} = N_{extra} / N_{key}$

Where $N_{coincidences:}$ number of requirements coinciding with the reference solution, $N_{key:}$ number of requirements in the reference solution and $N_{extra:}$ number of requirements not coinciding with the reference solution.

4.2.2.6. The experimental task: To evaluate the effectiveness of DREQUS, we compared requirements elicitation processes assisted by DREQUS against requirements elicitation processes assisted by Brainstorming. We have selected Brainstorming due that is the most well-known and widely-used creativity fostering technique and is probably the only such technique used in most software developing companies [14]. The empirical study consists of two main phases: Preparation and execution as is presented in Fig. 1.

4.2.2.6.1. Preparation phase: Sets the empirical study conditions which are necessary for the execution phase. The steps of this phase are:

1. Fill demographic questionnaire: In a preliminary session, the students answered a pre-test questionnaire from which we established that all students count only with academic experience in requirements elicitation. The age of the students ranges from 19 to 24. After, we formed, in a random manner, Groups 1 and 2; then into each group, once again randomly, we formed the pairs. Table 2. presents the characteristics of both groups.

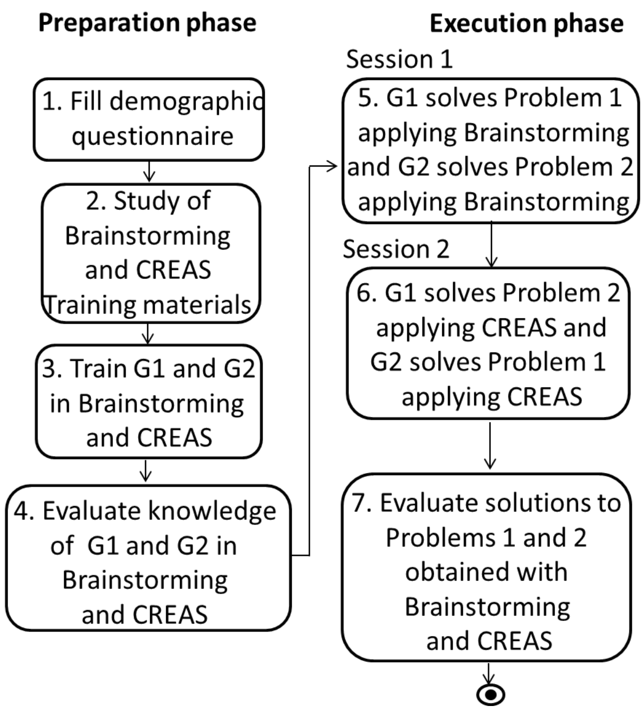


Figure 1: Empirical study process

Table 2: Characteristics of the Groups

Group	**Men**	**Women**	**Number of Pairs**	**Students´ experience in requirements elicitation**
1	19	1	10	Academic
2	19	1	10	

2. Study of Brainstorming and DREQUS training materials: In this step, professors and pairs studied Brainstorming and DREQUS by themselves. To do this, we provided them with training materials (these materials are available at https://goo.gl/hKgUPD).

3. Train G1 and G2 in Brainstorming and DREQUS: In a 2 hours session, the professors trained both Groups in Brainstorming (1 hour) and DREQUS (1 hour).

4. Evaluate the knowledge of G1 and G2 in Brainstorming and DREQUS: Two days after the training session, the professors made a written test aimed to evaluate the sufficiency of G1 and G2 in Brainstorming and DREQUS. Previously, the professors established the value of the means equal or superior to 4.0 (being 5.0 the maximum value) as the acceptance criterion. As a result, the means of both Groups were 4.3 and 4.1 for Brainstorming and DREQUS respectively; therefore, the Groups approved the tests, showing sufficiency to apply the approaches to a real problem.

4.2.2.6.2 Execution phase:

5. Using Brainstorming, G1 solved Problem 1 and G2 solved Problem 2: These tasks were carried out in a posterior session of two hours. The descriptions of the problems appear on the site: https://goo.gl/CAF4ws

6. Using DREQUS, G1 solved Problem 2 and G2 solved Problem 1: These tasks were performed in another session of two hours which took place two days after step 5.

7. Evaluate the solutions obtained with Brainstorming and DREQUS: Using the reference solutions, the RE professors computed and compared the results of the approaches in terms of recall, precision and over-specification.

4.2.2.7.Instruments: The used instruments include: - A pre-test questionnaire and a software requirements specification template used to condense the results (both are available at https://goo.gl/U7AYC0), - the experimental objects corresponding to the problems statements and domains descriptions, and training materials in Brainstorming and DREQUS.

4.2.3. Empirical Study Results

The study was conducted in May 2018; in the execution phase, we cited both groups for the first session of two hours. In this session, the groups were assigned to different classrooms and then they were asked to elicit the requirements for a Meeting Scheduler system (Group 1) and a Movie Streaming system (Group 2). Both groups were required to use Brainstorming. Each pair was provided with the problem statement, a domain description and a template to be filled up with the discovered requirements. The pairs were required to not use additional documentation or help.

Two weeks after the first session, a second one was carried out under the same conditions of session 1. This time, the groups were asked to elicit the requirements for a Meeting Scheduler system (Group 2) and a Movie Streaming system (Group 1) using DREQUS.

Using the reference solutions, the professors evaluated the requirements specifications delivered by the teams and calculated, for each of them, the quality indicators: Recall, precision, and over-specification (data available at https://goo.gl/ZyQNH0). This information was used to validate the research hypotheses as is presented in the next section.

4.2.3.1. Data Analysis

In order to analyze the obtained data, we have used the repeated measures (or within subjects) General Linear Model (GLM). We selected this model considering its advantages concerning to the study restrictions [15]*:* - Limited number of subjects: The repeated measures design reduces the variance of estimates of treatment effects, allowing statistical inference to be made with fewer subjects and - Efficiency: Repeated measures designs allow many experiments to be completed more quickly, as fewer groups need to be trained to complete an entire experiment. For example, experiments in which each condition takes only a few minutes, whereas the training to complete the tasks take as much, if not more time.

Two conditions must be accomplished to apply the GLM test: Sphericity and homogeneity of the covariance matrices.

Sphericity: this property can be verified with the Mauchly's test; nevertheless, in our study, there are only two levels of repeated measures (using Brainstorming and using DREQUS) which precludes a Sphericity violation [16].

Homogeneity of the covariance matrices: We used the Box's M test to evaluate this condition. Having as a H_o: The observed covariance matrices of the dependent variable are equal across groups we obtained: M = 65,104, F = 1,951, df1 = 21, df2 = 1191,671, Sig. = 0,006. Box's M is highly sensitive, therefore we worked with an $\alpha = 0{,}001$ [17]. The obtained Sig. = 0,006 verify the null hypothesis, meaning that the data are homogeneous.

The power ($1 - \beta$) of a statistical test is the probability that its null hypothesis (H_o) will be rejected given that it is in fact false. Power in software engineering experiments tends to be low, e.g. Dybå et al. [18] report values of 0,39 for medium effect sizes and 0,68 for large effect sizes. Low values of power mean that non-significant results may involve accepting null hypotheses when they are false [19].

GLM test shows if there is (or not) a significant difference between treatments of each factor; however, this test does not indicate the size of that difference. Considering this and the fact that our dependent variables (recall, precision, and over-specification) do not follow a normal distribution, we used Cliff's delta non-parametric test to measure the sizes of the effects. Cliff's delta ranges in the interval [-1, 1] and is considered small for $0{,}148 \le d < 0{,}33$, medium for $0{,}33 \le d < 0{,}474$ and large for $d \ge 0{,}474$. A positive sign means that the values of the first treatment are greater than the values of the second one (inversely for a negative sign) [20]. We have used Cliff's delta calculator, proposed by Macbeth et al. [21], to perform these tests.

GLM tests were executed using SPSS v. 20. The Problem variable was introduced as a "Between-Subjects Factor". In the next section, the main outcomes of the tests are presented.

4.2.3.2. Results

a) Recall: The higher value is associated with better recall. Figs. 3(a) and 3(b) present a graphic comparison of the recall obtained with Brainstorming and the recall obtained with DREQUS. From these graphics, we observe that DREQUS offers a better recall in both cases (Problems 1 and 2). The medians obtained with DREQUS are 0,3610 (for problem 1) and 0,3543 (for problem 2); we estimate that these results are influenced by the duration of the sessions (2 hours by session) and they could improve, not only for DREQUS but also for Brainstorming, with more time available when the approaches will be applied. We also observe, see Fig. 2(a), that DREQUS result with problem 1 is similar to the DREQUS result with problem 2 and the same occurs with Brainstorming; we estimate that this is indicative of the similarity of the problems complexity.

The hypothesis that we are intended to validate is $H1_o$: Requirements elicited using DREQUS are not significantly more complete than those elicited with a Brainstorming process. We have measured completeness in terms of recall. The p-value for Approach is 0,008 (Fig. 3(b)) which indicates that there is a significant difference between Brainstorming and DREQUS. Statistical power is 0,798 (α = 0,05) which precludes the possibility of a Type II error (the probability of falsely retaining an incorrect H_o).

Cliff's delta is - 0,425 evidencing a medium effect in the direction that DREQUS has better recall than Brainstorming.

The Approach * Problem interaction Is not significant (p-value = 0,607) which implies that problem type does not influence recall (this also corresponds to the visual inspection of Fig. 3(a)). Nevertheless, statistical power is low (0,079; α = 0,05) meaning that may be making false negatives possible and larger sample sizes are required to make a more reliable assessment of the existence (or not) of this interaction.

In summary, we conclude that $H1_o$ is rejected: There is a significant difference between the completeness (recall) obtained with DREQUS and the completeness obtained with Brainstorming. Besides, the evidence shows that DREQUS provides a better recall than Brainstorming.

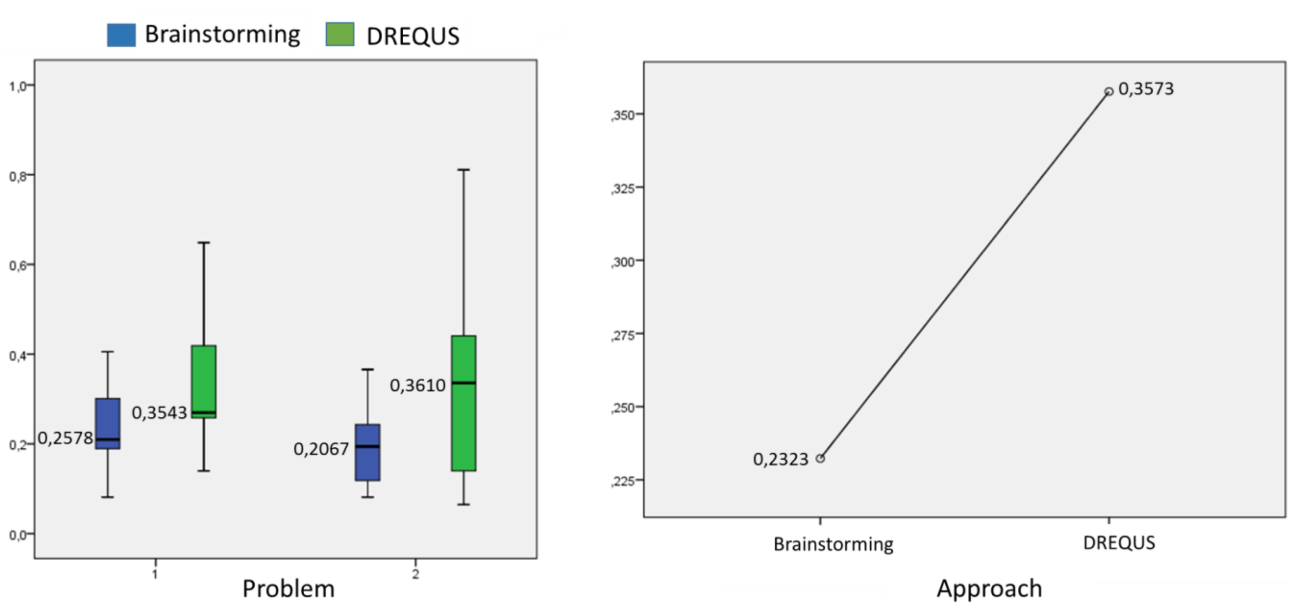
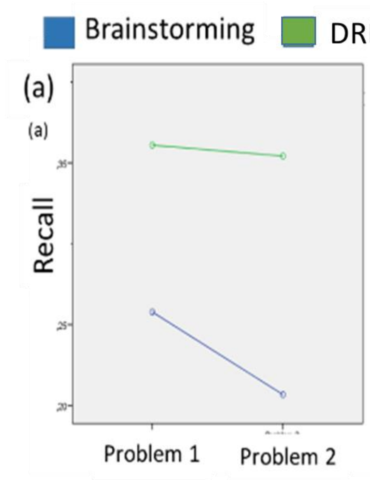


Figure 2: (a) Box-and-whisker plot for Recall. (b) Profile Plot of both approaches



(b)

Source	Approach	Type III Sum of Squares	df	Mean Squares	F	Sig.
Approach	Linear	,157	1	,157	8,734	,008
Approach*Problem	Linear	,005	1	,005	,274	,607
Error(Approach)	Linear	,324	18	,018		

Figure 3: (a) Profile plot of the Approach * Problem interaction. (b) GLM test for Recall.

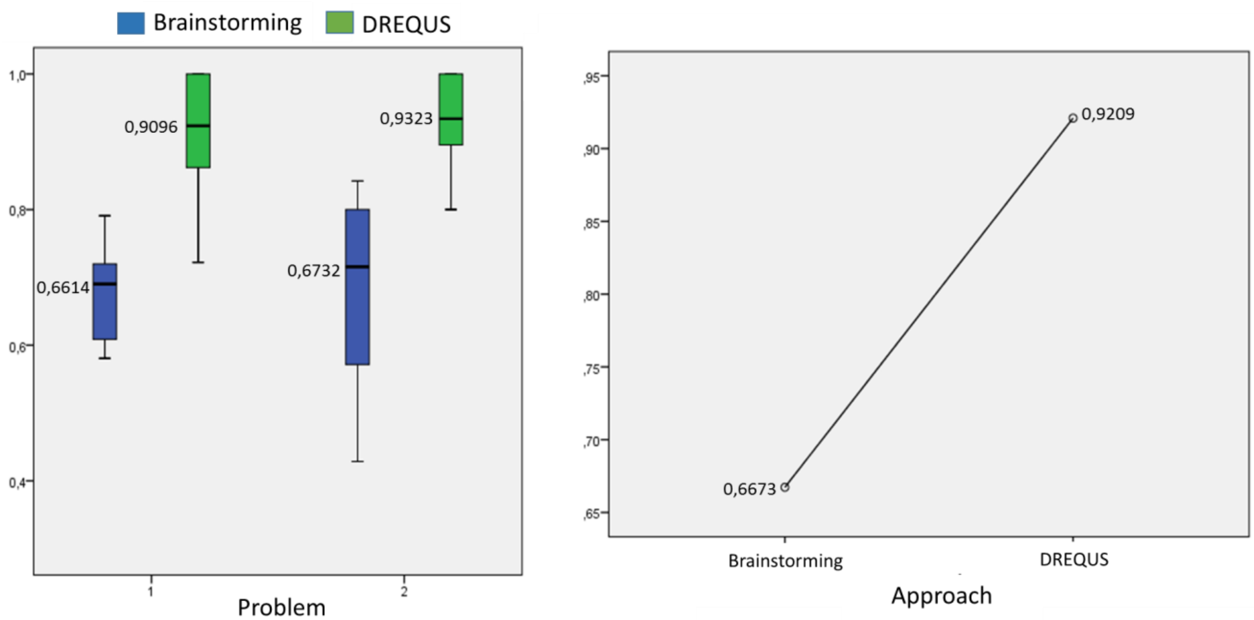
b) Precision: The higher value is associated with better precision. Figs. 5(a) and 5(b) show the graphic comparison between the precision obtained with Brainstorming and the precision obtained with DREQUS. We observe that DREQUST presents a better precision than Brainstorming. The medians obtained with DREQUS 0,9096 and 0,9323 (for problem 1 and problem 2 respectively) show a high level of precision considering the experimental setting (e.g. problems complexity and time restrictions). We note that the results of DREQUS with problem 1 are alike to DREQUS with problem 2 and the same happens with Brainstorming (Fig. 4(a)). In consonance with the previous section (recall results), we estimate that this indicates that the problems have similar complexities.

The hypothesis to validate is $H2_o$: Requirements elicited using DREQUS are not significantly more precise than those elicited with a Brainstorming process. The p-value for Approach is 0,000 (Fig. 5(b)) which indicates that there is a significant difference between Brainstorming and DREQUS. Statistical power is 1,000 which means that there is no possibility of a Type II error.

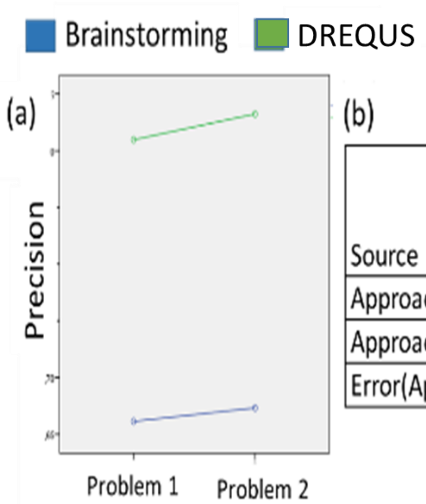
Cliff's delta is -0,9425 denoting a large effect in the direction that DREQUS has better precision than Brainstorming.

The Approach * Problem interaction is not significant (p-value = 0,858) evidencing that the problem type does not influence precision (Fig. 5(a) allow us to make a visual verification of this aspect). However, statistical power is low (0,053; $\alpha = 0,05$) therefore false negatives are possible and larger sample sizes are required to make a more reliable assessment of the existence (or not) of this interaction.

Summarizing, we conclude that $H2_o$ is rejected: There is a significant difference between precision using Brainstorming and precision using DREQUS. In this sense, results show that DREQUS offers better precision than Brainstorming.



5: (a) Figure 4: Box-and-whisker plot for Precision. (b) Profile Plot of both approaches.



(b)

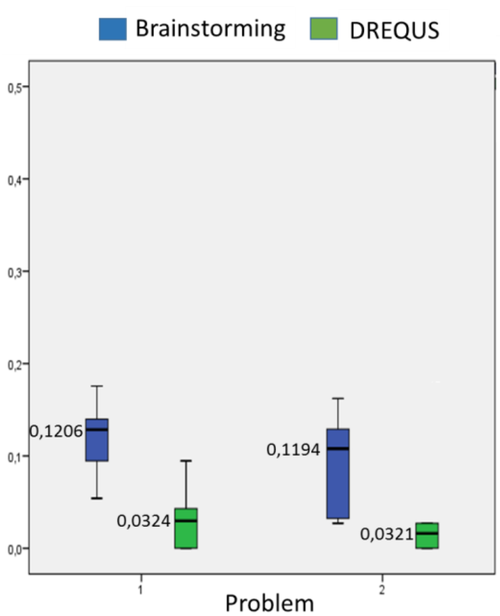
Source	Approach	Type III Sum of Squares	df	Mean Squares	F	Sig.
Approach	Linear	,643	1	,643	72,558	,000
Approach*Problem	Linear	,000	1	,000	,033	,858
Error(Approach)	Linear	,160	18	,009		

Figure 5: (a) Profile plot of the Approach * Problem interaction. (b) GLM test for Precision.

c) Over-specification: The lower value is associated with better over-specification. A graphic comparison between the over-specification obtained with Brainstorming and the over-specification obtained with DREQUS appears in Figs. 7 (a) and 7(b). We observe that there are important differences between the over-specification obtained with Brainstorming and the over-specification obtained with DREQUS; presenting DREQUS a better result in comparison to Brainstorming. The medians obtained with DREQUS 0,0321 (for problem 1) and 0,0324 (for problem 2) show low levels of over-specification which is a desirable characteristic in any requirements elicitation approach. Reaffirming the findings related to recall and precision (precedent sections), once again we observe a similar performance of both approaches in relation to the problems which allow us to estimate that effectively the complexity of problems 1 and 2 is similar (Fig. 6(a)).

The hypothesis to validate is $H3_o$: Requirements elicited using DREQUS do not significantly present less over-specification than those elicited with a Brainstorming process. The p-value for Approach (0,000; Fig. 7(b)) indicates that there is a significant difference between Brainstorming and DREQUS. Statistical power is 0,992 which precludes the possibility of a Type II error.

Cliff's delta is 0,7875 evidencing a large effect in the direction that DREQUS presents a better over-specification than Brainstorming.



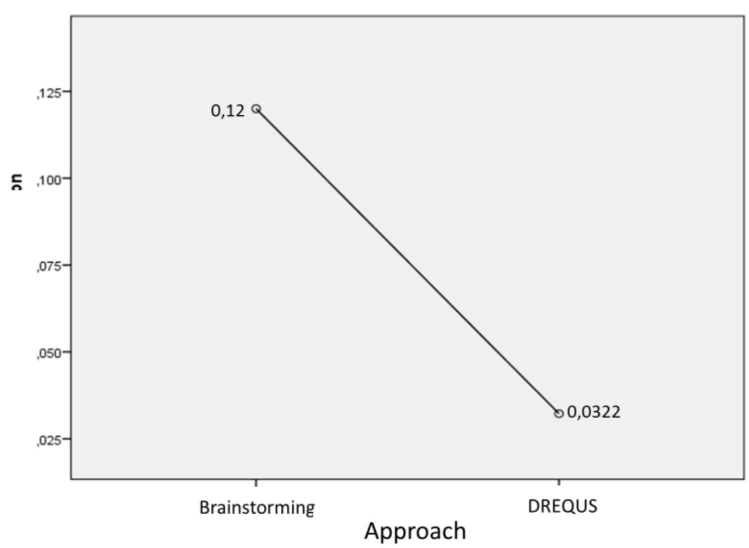
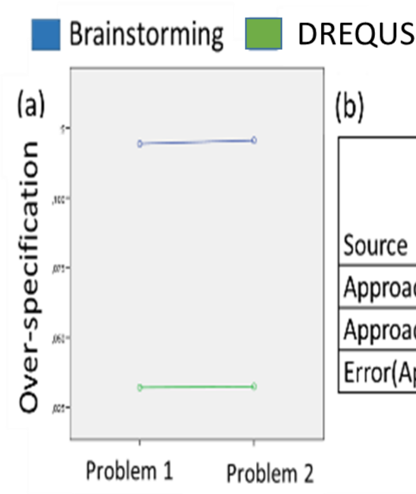


Figure 6: (a) Box-and-whisker plot for Over-specification. (b) Profile Plot of both approaches.

The Approach * Problem interaction is not significant (p-value = 0,981) indicating that the problem type does not influence over-specification (a visual inspection of this aspect can be made in Fig. 7(a). The statistical power of the interaction is low (0,05; $\alpha = 0,05$) which indicates that false negatives are possible and larger sample sizes are required to make a more reliable assessment of the existence (or not) of this interaction.

In conclusion, we reject $H3_o$: There is a significant difference between the over-specification obtained using Brainstorming and the over-specification obtained with DREQUS. Results also evidence that DREQUS presents a better over-specification in comparison to Brainstorming.



(b)

Source	Approach	Type III Sum of Squares	df	Mean Squares	F	Sig.
Approach	Linear	,077	1	,077	21,524	,000
Approach*Problem	Linear	2,095E-06	1	2,095E-06	,001	,981
Error(Approach)	Linear	,064	18	,004		

Figure 7: (a) Profile plot of the Approach * Problem interaction. (b) GLM test for Over-specification.

4.2.4. Related Works and Threats to Validity

Empirical studies in the field of Requirements Engineering are described in [23, 24, 25, 26, 27]. In particular, empirical studies to evaluate requirements elicitation approaches can be found at [28] and [31]. Davis et al. [29] report a systematic review of empirical studies concerning the effectiveness of elicitation techniques. Finally, a review of creativity based requirements elicitation approaches can be found in [6].

Threats to the validity of our empirical study were mitigated following considerations proposed by Kitchenham et al. [22] and Madeyski and Kitchenham [30]. In this sense, we have defined methodological, aspects to mitigate possible validity threats:
The construct validity refers to how well the studied parameters and their outcomes are adequate to the research question addressed. In this sense, we have selected a set of quality indicators aimed at measuring the performance of the requirements elicitation processes in comparison; the selected indicators: recall, precision, over-specification are widely accepted by the Requirements Engineering community [12], [23]. Likewise, the quality indicators measures were obtained using a reference solution proposed by Cleland-Huang [11]; these results allowed the hypotheses contrast through GLM-tests that are frequently used by the software engineering community in similar studies.

Internal validity is concerning with factors that influencing the independent variables. To mitigate these factors, we have evaluated the students´ experience in requirements elicitation; according to these results, the subjects were randomly assigned to the groups and teams. To avoid the demotivation of subjects, we have made an experimental design aligned with the goals of the requirements engineering course and participant's expectations. To avoid that pairs can acquire knowledge with the first treatment and apply it to the second one, we have used two different problems, thus the subjects do not get this possibility. We also have assigned different classrooms and the same duration for the execution of the tasks; besides, all the teams' results were assessed, with the same instruments, by independent professors.

The conclusion validity concerns the relationship between the treatments and the outcomes. To avoid threats related to the subjects' heterogeneity, we have measured this aspect employing a pre-test questionnaire and we have also trained and assessed the pairs' knowledge in Brainstorming and DREQUS. To avoid threats with the sample size (low statistical power), we have used the repeated measures (or within subjects) General Linear Model (GLM); this test is quite robust to contrast the quantitative hypotheses in a reliable manner. Besides, we calculated the statistical power for each null hypothesis.

The external validity refers to the ability to generalize the research findings to other domains under different settings [22]. Regarding this aspect, this empirical study presents preliminary results of DREQUS and Brainstorming and more studies are required to make generalizations in other contexts.

4.2.5. Conclusions

In order to assess the effectiveness of the DREQUS (Discovery of REQuirements Using Scenarios) approach, we have conducted an empirical study in which we compared requirements elicitation processes assisted by DREQUS against requirements elicitation processes assisted by Brainstorming. The study was carried out with 40 participants (requirements engineering students) and two professors at the University of Antioquia – Colombia. Comparisons of the requirements specifications quality were made in terms of completeness (recall), precision, and over-specification obtained with both approaches. In the sequel, the main results are presented:

Regarding recall, there is a significant difference between Brainstorming and DREQUS. The evidence shows that DREQUS provides a better recall than Brainstorming. We estimate that the medians obtained with DREQUS: 0,3610 (for problem 1) and 0,3543 (for problem 2); were influenced by the duration of the sessions (2 hours by session) and they could improve, not only for DREQUS but also for Brainstorming, with more time available when the approaches will be applied.

Considering precision, there is a significant difference between Brainstorming and DREQUS. Results show that DREQUS offers better precision than Brainstorming. The medians obtained with DREQUS 0,9096 and 0,9323 (for problem 1 and problem 2 respectively) evidence a high level of precision considering the restrictions of the experimental setting (e.g. problems complexity and time restrictions).

Concerning over-specification, there is a significant difference between Brainstorming and DREQUS. Results indicate that DREQUS presents a better over-specification in comparison to Brainstorming. The medians obtained with DREQUS, 0,0321 (for problem 1) and 0,0324 (for problem 2), show low levels of over-specification which is a desirable characteristic in any requirements elicitation approach.

In all cases (recall, precision, and over-specification): - Results of DREQUS obtained for problem 1 are similar to the DREQUS results obtained for problem 2 and the same occurs with Brainstorming; we estimate that this is indicative of the similarity of the problems complexity which evidences the suitability of the problems for the study; - The Approach * Problem interaction is not significant which implies that the problem type does not influence the dependent variables. Nevertheless, the statistical power is low meaning that may be making false negatives possible and larger sample sizes are required to make a more reliable assessment of the existence (or not) of these interactions.

In conclusion, regarding the research question, the results indicate that, in comparison to Brainstorming, the use of DREQUS improves the quality of the elicited requirements.

References

[1] Cheng, B.H.C., Atlee, J.M. 2007. "Research Directions in Requirements Engineering". In Proceedings Future of Software Engineering (FOSE'07), p. 285-303.

[2] Nuseibeh, B., Easterbrook, S. 2000. "Requirements Engineering: a roadmap". In Proceedings of the conference of the Future of Software Engineering, p. 35–46.

[3] Sternberg, R.J., Lubart, T. 1995. Defying the Crowd: Cultivating Creativity in a Culture of Conformity, Free Press.

[4] Maiden, N., Jones, S., Karlsen, K., Neill, R., Milne, A. 2010. Requirements Engineering as Creative Problem Solving: A Research Agenda for Idea Finding. 18th IEEE International Requirements Engineering Conference (RE), Sydney.

[5] Robertson, J. 2005. "Requirements analysts must also be inventors", *IEEE Software*, vol. 22, nº 1, p. 48, 50.

[6] Jaramillo, A., Assar, S. 2016. Leveraging creativity techniques in requirements elicitation: a literature review. Requirements Engineering Magazine. International Requirements Engineering Board. Issue 2016-02.

[7] Mendling, J., Recker, J., Reijers, H. 2010. On the usage of labels and icons in business process modeling. International Journal of Information System Modeling and Design, 1(2), 40-58.

[8] Metzler, T., Shea, K. 2011. Taxonomy of cognitive functions. Proceedings of the 18th International Conference on Engineering Design. ICED11.

[9] van Lamsweerde, A., Feather, M.S., Fickas, S., Finkelstein A. 1997. "Requirements and specification exemplars," Automated Software. Engineering, vol. 4, no. 4, pp. 419–438.

[10] Regnell, B., Höst, M., Wohlin, C. 2000. Using students as subjects—A comparative study of students and professionals in lead-time impact assessment. Empirical Software Engineering, 5, 201–214. Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

[11] Cleland-Huang, J. 2019. https://terapromise.csc.ncsu.edu/repo/requirements/nfr/nfr.arff

[12] Pastor, Ó., González, A., Condori-Fernández, N., España, S. 2010. An empirical comparative evaluation of requirements engineering methods. Brazilian Computer Society.

[13] Harmain, H., Gaizauskas, R.: CM-Builder. 2000. An automated NL-Based CASE Tool. Proceedings of the Fifteenth IEEE International Conference on Automated Software Engineering (ASE´00). Grenoble.

[14] Berry, D., Mich, L., Anesi, C. 2005. Applying a pragmatics-based creativity-fostering technique to requirements elicitation. Requirements Engineering Journal.

[15] Barret, J.R. 2013. "Particulate Matter and Cardiovascular Disease: Researchers Turn an Eye Toward Microvascular Changes". Environmental Health Perspectives. 121-282.

[16] Meyers, L.S. 2006. Applied multivariate research: design and interpretation/ Lawrence S. Meyers, Glenn Gamst, A.J. Guarino, SAGE Publications, Thousand Oaks.

[17] Tabachnick, B.G., Fidell, L.S. 2001. Using Multivariate Statistics. 4th Edition. New York: Allyn & Bacon.

[18] Dybå, T., Kampenes, V.B., Sjøberg, D.I.K. 2006. A systematic review of statistical power in software engineering experiments, Inf. Softw. Technol. 48. 745–755.

[19] Panach, J.I., España S., Dieste Ó., Pastor Ó., Juristo N. 2015. In search of evidence for model-driven development claims: An experiment on quality, effort, productivity and satisfaction. Information & Software Technology 62: 164-186.

[20] Grissom, R.J., Kim J.J.: Effect Sizes for Research. 2005. A Broad Practical Approach, Taylor & Francis Group.

[21] Macbeth, G., Razumiejczyk, E., Ledesma, R.D.: Cliff's Delta Calculator. 2011. A non-parametric effect size program for two groups of observations. Universitas Psychologica, 10 (2), 545-555.

[22] Kitchenham, B., Dyba, T., Jorgensen, M. 2004. "Evidence-based software engineering," in Proc. of the 26th International Conference on Software Engineering (ICSE04), pp. 273–281.

[23] Morandini, M., Marchetto, A., Perini, A. 2011. Requirements comprehension: A controlled experiment on conceptual modeling methods. First International Workshop on Empirical Requirements Engineering (EmpiRE).

[24] Mahaux, M., Nguyen, L., Mich, L., Mavin, A. 2014. A framework for understanding collaborative creativity in requirements engineering: Empirical validation. Workshop on Empirical Requirements Engineering (EmpiRE), IEEE Fourth International.

[25] Labunets, K., Paci, F., Massacci, F., Ruprai, R. 2014. An experiment on comparing textual vs. visual industrial methods for security risk assessment. IEEE Fourth International Workshop on Empirical Requirements Engineering (EmpiRE).

[26] Daneva, M., Damian, D., Marchetto, A., Pastor, O. 2014. Empirical research methodologies and studies in Requirements Engineering: How far did we come? The Journal of Systems and Software 95. 1–9.

[27] Condori-Fernandez, N., Daneva, M., Sikkel, K., Wieringa, R., Dieste, O., Pastor, O. 2009. A systematic mapping study on empirical evaluation of software requirements specifications techniques. ESEM 2009. 3rd International Symposium on Empirical Software Engineering and Measurement.

[28] Oliveira, M., Viana, D., Conte, T., Vieira, S., Marczak, S. 2013. Evaluating the REMO-EKD technique: a technique for the elicitation of software requirements base don EKD organizational models. Empirical Requirements Engineering (EmpiRE). IEEE Third International Workshop on. P. 9-16.

[29] Davis, A., Dieste, O., Hickey, A., Juristo, N., Moreno, A.M. 2006. Effectiveness of Requirements Elicitation Techniques: Empirical Results Derived from a Systematic Review. Requirements Engineering, 14th IEEE International Conference.

[30] Madeyski L., Kitchenham B. 2018. Effect sizes and their variance for AB/BA crossover design studies. Empirical Software Engineering. 23: 1982. https://doi.org/10.1007/s10664-017-9574-5.

[31] Hermann, A., Mich,, L., Berry, D. 2018. Creativity Techniques for Requirements Elicitation: Comparing Four-Step EPMcreate-Based Processes. IEEE 7th International Workshop on Empirical Requirements Engineering (EmpiRE).

CHAPTER 5

Discussion, Conclusions and Future Research

5.1. Discussion and Conclusions

The poor discovery of the stakeholders´ requirements is still one of the main causes of the failure of software development projects. Motivated by this fact, we have proposed an approach for the systematic Discovery of REQuirements Using Scenarios (DREQUS). This proposal belongs to the family of Creativity-based Approaches for Requirements Elicitation (CAREs).
The performance of DREQUS was evaluated in terms of *Recall (Completeness), Precision, and Over-specification* through the Empirical Studies (ES) presented in the previous section. Regarding the research questions, the results obtained from the ES are:

RQ1: ¿How the alignment of the discovered requirements with the system purpose can be ensured?

To provide this alignment, DREQUS proposes the use of the *System promise* concept: which is an explicit declaration or assurance made to a system agent regarding to the future, stating that the system will do or refrain from some specified act, or that the system will give or bestow some specified thing.

The efficacy of the proposed mechanism was evaluated using the metrics: precision and over-specification. The ES results evidenced that the *System promise* is useful to the effective alignment of the discovered requirement with the system purpose.

RQ2.: ¿How effective guidance for the systematic exploration of the Solution Ideas Space and requirements discovery can be provided?

Based on the General Systems Theory (GST), DREQUS makes a partition of the Solution Ideas Space (SIS) into *Input, Evolution and Decision, and Output Ideas*. As a complement of the SIS partitioning, DREQUS proposes to use a set of *Functional and Cognitive Verbs* to discover requirements in each partition. These elements facilitate the systematic exploration of the SIS improving its coverage.

The efficacy of these mechanisms was evaluated through the "recall" metric. The ES allowed us to confirm the hypotheses that the proposed mechanisms facilitate the effective and systematic exploration of the SIS for the discovery of requirements.

RQ3: ¿How the transition from the discovered requirements towards the future system specification can be facilitated?

DREQUS facilitates the requirements specification through Use cases. This task is performed utilizing algorithms for requirements similarity calculation and clustering, and a proposed set of rules, and guidelines. The efficacy of DREQUS solving RQ3 was demonstrated through a case study considering the discovery of requirements for a "Booking rooms system".

In conclusion, and regarding the main research question:

Main RQ: ¿How the Requirements Engineer can be effectively assisted in the systematic exploration of the Solution Ideas Space and the consequent discovery of requirements for a future system?

The process and mechanisms implemented by DREQUS, as evidenced in the precedent section, facilitate the effective and systematic exploration of the SIS and the consequent discovery of requirements for a future system.

Limitations of DREQUS

The Empirical Studies evidenced the lack of usability of DREQUS as a drawback perceived by the Requirements Engineers. Considering that most of the processes performed by DREQUS can be automated and integrated; e.g. requirements similarity and clustering, we are implementing a tool for the proposed approach. We estimate that this software will substantially improve the perceived usability of DREQUS.

Implications of DREQUS

DREQUS proposes a solution aimed to tackle the problem: "The poor discovery of the stakeholders´ requirements is still one of the main causes of the failure of software development projects."

The research outcomes evidence that DREQUS contributes to improving the quality of the discovered requirements and, as a positive consequence for the software industry, the quality of software products derived from them.

5.2. Future Research

Regarding future works, we identify the next research agenda:

a. The development of a tool that implements the DREQUS approach

Due to the complexity of developing a tool that implements the proposed approach, taking into account time and human restrictions, we executed the Empirical Studies without the support of this valuable resource. In fact, a survey performed in the first Empirical Study evidenced the lack of usability of DREQUS as a drawback perceived by the Requirements Engineers. Considering that most of the processes performed by DREQUS can be automated; e.g. requirements similarity and clustering, currently we are developing a tool that implements the approach. With no doubt, this tool will improve the perceived easiness of DREQUS and the quality of the discovered requirements.

b. The identification of sets of verbs useful in other domains

The current version of DREQUS is useful for the discovery of requirements in the domain of Business Information Systems (BIS). In future research, we are interested in identifying sets of verbs useful to the discovery of requirements in other domains; e.g. video games and healthcare.

c. The use of a taxonomy of Non-Functional Requirements (NFRs) to improve the discovery of this type of requirements

Considering our work entitled "Non-functional requirements elicitation from business process models", we are interested in improving the discovery of NFRs using the taxonomy of requirements proposed in the referred paper. We conjecture that this will improve the discovery of NFRs; nevertheless, empirical evidence will also be necessary to validate this hypothesis.

d. The assessment of the level of innovation (novelty and utility) of the discovered requirements

The measurement of how innovative are the requirements discovered by DREQUS is a relevant question; nevertheless, due to scope reasons, this will be considered as part of our future research.

We will study this aspect to make the necessary improvements to ensure the novelty and usability of the discovered requirements as one of the main advantages of DREQUS.

e. The validation of DREQUS in industrial contexts

To this aim, we are prioritizing the implementation of DREQUS and making contact with the local industry of software aiming at performing two initial Empirical Studies in software factories contexts.

CHAPTER 6

Publications

The publications this thesis has produced so far are:

Jaramillo A.; Urrego G. On the use of business processes models to discover system requirements. International Conference on Information Management and Engineering. ICIME 2019. ACM. London, England. 2019.

Jaramillo A.; Urrego G. Brainstorming versus a scenario-based approach: results of an empirical Study. International Conference on Software and Information Engineering. ICSIE 2019. ACM. Cairo, Egypt. 2019.

Jaramillo A.; Assar S. Leveraging creativity techniques in requirements elicitation: a literature review. Requirements Engineering Magazine. International Requirements Engineering Board. Issue 2016-02. Karlsruhe. Germany. IREB. 2016.

Jaramillo A. Requirements elicitation approaches: a systematic review. Ninth International Conference on Research Challenges in Information Science RCIS 2015. Athens. Greece**.** IEEE. 2015.

Jaramillo A. Non-functional requirements elicitation from business process models. Fifth IEEE International Conference on Research Challenges in Information Science. RCIS 2011. Guadeloupe - French West Indies, France. IEEE. 2011.