DSS from an RE Perspective: a Systematic Mapping

Stephany García, Oscar Romero,[1] Ruth Raventós

Departament d'Enginyeria de Serveis i Sistemes d'Informació (ESSI)
Universitat Politècnica de Catalunya, BarcelonaTECH

{sgarcia | oromero | raventos@essi.upc.edu}

Abstract

Decision support systems (DSS) provide a unified analytical view of business data to better support decision-making processes. Such systems have shown a high level of user satisfaction and return on investment. However, several surveys stress the high failure rate of DSS projects. This problem results from setting the wrong requirements by approaching DSS in the same way as operational systems, whereas a specific approach is needed. Although this is well-known, there is still a surprising gap on how to address requirements engineering (RE) for DSS.

To overcome this problem, we conducted a systematic mapping study to identify and classify the literature on DSS from an RE perspective. Twenty-seven primary studies that addressed the main stages of RE were selected, mapped, and classified into 39 models, 27 techniques, and 54 items of guidance. We have also identified a gap in the literature on how to design the DSS main constructs (typically, the data warehouse and data flows) in a methodological manner from the business needs. We believe this study will help practitioners better address the RE stages of DSS projects.

***Keywords**: Decision support systems, requirements engineering, business intelligence*

[1] Corresponding author

1. Introduction

The amount of digital data in the universe is growing beyond all bounds. This growth presents enormous challenges and new business opportunities. As data continues expanding, decision-makers and stakeholders are increasingly demanding computerized support with intelligent solutions that can convert data into meaningful information and improve an organization's decision-making capabilities. This has led to the creation of a dominant technology called *decision support systems* (DSS) that can integrate heterogeneous sources in an analytical fashion to provide a better decision-making process (Paim & Castro, 2003). DSS are capable of providing assistance to managers at various organizational levels for analyzing strategic information by collecting vast amounts of data on organizational behavior (Vaisman & Zimányi, 2014). Successful DSS implementation projects have shown a high degree of user satisfaction and return on investment (Paim & Castro, 2003). Despite the benefits and potential of these systems, several surveys have indicated that the failure rate of DSS projects in case studies and the literature is high (Cabibbo & Torlone, 1998; Lehner, Albrecht & Wedekind, 1998; Paim & Castro, 2003; Vassiliadis, 2000; Giorgini, Rizzi & Garzetti, 2008). The problem results from setting the wrong requirements by approaching DSS in the same way as operational systems (i.e., information systems meant to assist day-by-day business operations), without considering that their development is different (Paim & Castro, 2003; Wrembel & Koncilia, 2007). DSS development is expected to: deal with diverse domain terminology that typically leads to poor communication between business users and IT professionals; provide substantial business resource commitments throughout the entire project; integrate heterogeneous sources demanding complex procedures to control the integration and transformation phases; analyze the quality and completeness of data sources; create analytical results that can be explored across multiple levels (e.g., dimensions); provide traceability of data sources; provide comprehensible design of the unified view; and keep the data repository up-to-date according to the user specifications.

Given the specificity of DSS, new architectural solutions (data warehousing) and modeling techniques (the multidimensional model) were developed. In addition to the aforementioned characteristics, eliciting the business needs is a crucial aspect of DSS (Kimball, 1998). During such processes, new ideas often arise as business users begin to realize the potential of these systems and leverage the capabilities for the decision-making process. Typically, several iterations are required to design the main data warehouse constructs. This process tends to be error-prone and demands several rounds of redesign of the data warehouse to satisfy all the business requirements stated by the stakeholders. Although the relevance of requirements engineering (RE) is well-known, the above-mentioned characteristics stress that DSS require a more specialized, extensive, and detailed RE process than traditional operational systems.

Considering the previously mentioned issues, it is reasonable to suggest that leading information systems professionals lack a holistic vision of RE processes for DSS. By providing this vision, practitioners will be able to make crucial choices with the certainty that the distinctive characteristics of DSS are being taken into account. This fact was clearly observed when, jointly with the World Health Organization, our team started the Chagas Information DSS (CIDSS); a strategic and critical DSS that required a systematic approach (Raventós et al., 2015). We decided to conduct a mapping study of RE for DSS. As a result, we mapped the relevant literature to a generic framework that structured the requirements process (Pohl, 2010) with the aim of classifying and organizing the existing work.

This paper is structured as follows: Section (2) introduces the main terminology related to DSS to contextualize our approach. Section (3) describes the process of how we identified the need for our mapping study and its execution (including the definition of the research questions), and describes the planning and process to extract the selected literature (inclusion/exclusion criteria). Section (4) presents the results of the mapping study and the classification of DSS from an RE perspective. In addition, this section provides a discussion summarizing the highlights extracted from the mapping. To conclude our paper, Sections (5) and (6) provide the threats-to-validity and conclusions.

2. Background

This section begins by contextualizing our approach with regard to existing RE approaches. This is followed by a description of the background of business intelligence and DSS. And finally, this section presents our motivation for mapping DSS from an RE perspective.

2.1. Klaus Pohl's approach to RE

Since the mid-1970s the definition of RE has evolved from initially being concerned with software systems (IEEE-Std.'729' 1983; IEEE-Std.'830' 1984) to a broader perspective incorporating aspects of systems and organisations (Greenspan et al. 1994; Loucopoulos & Karakostas, 1995; Pohl, 1996; Yu, 1997; Zave, 1997) and including the organizational context (Kavakli & Loucopoulos, 2003; Pohl 2010; van Lamsweerde 2001).

For the purpose of this mapping study, we have used the framework known as "The RE Framework" by Klaus Pohl (2010). This framework (one of several RE frameworks) defines the major structural building-blocks and elements of the RE process (such as elicitation, negotiation, documentation, goals, and validation); provides a well-structured base for the fundamentals, principles, and techniques of the RE process; and does not adhere to a specific methodology or a type of software project. Moreover, according to Pohl, this framework consolidates various research results and has been successfully used by several organizations when structuring their RE process.

The RE framework (see Figure 1) defines the major structural elements of a required engineering process and consists of the following building blocks (Pohl, 2010):

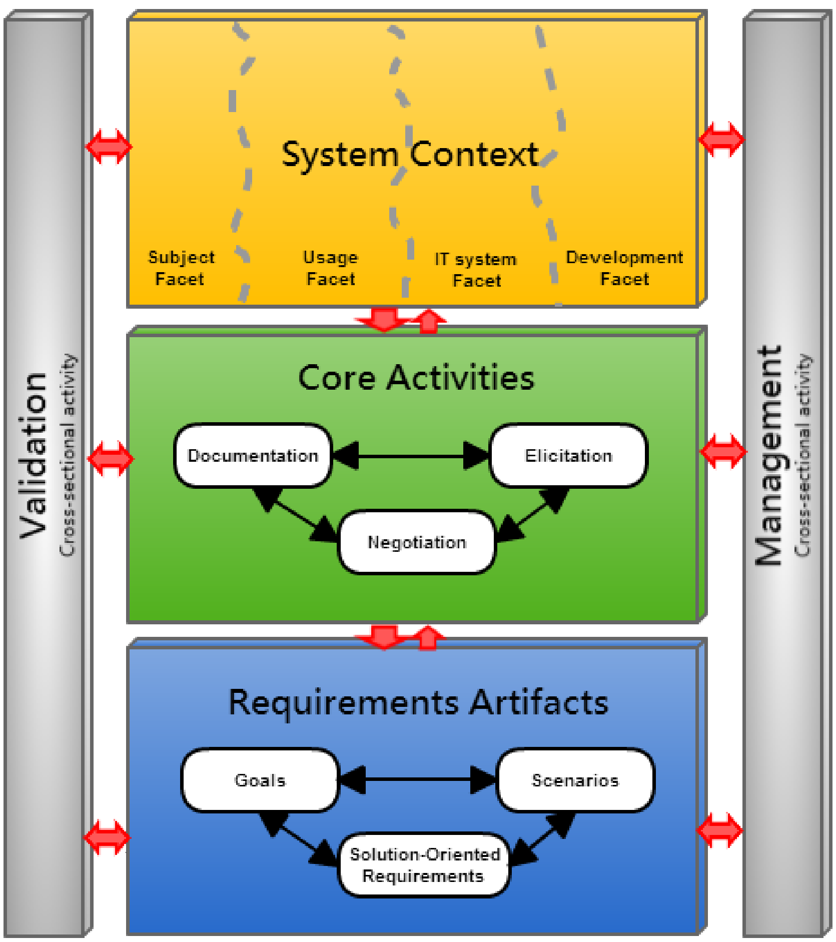

Figure 1: RE framework building-blocks (Pohl, 2010)

System context: a large number of aspects that are relevant to the system to be developed (including business processes, hardware, system users – to name a few). The framework structures the system context into four parts: subject; usage; IT system; and development facet.

Three core activities: three core RE activities (elicitation, negotiation and documentation) are performed iteratively to establish the vision (the goal to achieve with the system) within the existing context.

Two cross-sectional activities: validation and management are the two cross-sectional activities that support the previous core activities and validate the results of requirements engineering.

Requirement artefacts: the framework uses the term "requirement artefact" to refer to a documented requirement, i.e. a requirement using a specific documentation format. The framework differentiates three types of requirement artefacts, namely: goals; scenarios; and solution-oriented requirements. The first describes goals, i.e., intentions with regard to the objectives, properties, or use of the system. Scenarios document specific examples of system usage. And, finally, solution-oriented requirements define the data perspective, the functional perspective, and the behavioural perspective of a software-intensive system. Furthermore, solution-oriented requirements comprise of (solution-oriented) quality requirements and (solution-oriented) constraints.

2.2. Decision support systems

Business intelligence (BI) has become a huge industrial domain and a major economic driver that enables organizations to gather, transform, integrate, and summarize business data to generate analytical information suitable for decision-making. In today’s knowledge society, business success or failure is largely based on how effectively data is monitored and analyzed to predict future trends and make the best decisions. Currently, the most successful organizations embrace IT solutions as a strategic asset and embed them in decision making processes.

However, the BI concept is not new. Indeed, it dates back to 1958 when H. P. Luhn defined it as “the ability to apprehend the interrelationships of presented facts in such a way as to guide action towards a desired goal”. This definition remains accurate today, showing that the need to learn from business data (the truly final goal) has always been there. We now have better means to reach this goal thanks to advances over the past 20 years. BI is currently presented as two sides of the same coin. Following the original definition, there is a strong emphasis on the business aspects behind BI, as depicted in the *BI cycle* definition (Vaisman & Zimányi, 2014): different processes gather, process, and analyze business data in order to provide objective evidence to support organizational decision-making processes and align them with a business strategy. Decisions are then monitored by newly gathered data, which is processed and analyzed, and as a result new decisions are subsequently made. This never-ending cycle is part of the core strategy of data-oriented companies. There is little doubt that information systems (i.e., IT solutions) must be developed within the organizations to support the BI cycle. Obviously, this IT-aspect of BI was not originally developed, but evolved to the point that BI can no longer be conceived without its IT counterpart.

The information systems developed to support the BI cycle are commonly known as DSS; and data warehousing is the current de-facto implementation standard. Data warehousing is based in three main layers (Inmon, 2001): (1) the data warehouse (DW), which Bill Inmon defined as “a subject oriented, nonvolatile, integrated, time variant collection of data in support of management’s decisions”. Additionally, several data marts may be defined to better accommodate the analytical needs of various organizational units (e.g., departments that may focus on different data subsets). (2) The second layer contains the extraction, transform, and load (ETL) processes that nurture the DW with relevant data for decision making processes (from a range of potentially heterogeneous data sources such as databases, Excel files, free text, e-mails, etc.). The ETL processes extract the data, clean (i.e., apply quality rules), homogenize, and transform it prior to loading into a consolidated single schema (the DW schema). For this reason, ETL are typically represented as data workflows. Finally, the third layer contains (3) the exploitation tools aimed at analyzing the DW data (either in a basic fashion – such as reporting, or as summaries, or in a more advanced fashion – such as by means of data mining algorithms). Typically, to enable goal-oriented analysis, data marts are built on top of the DW. Data marts are a subset of a DW tailored to the DSS processing needs, customized to fit the needs of a segment or department (Inmon, 2001). Related to data warehousing, OLAP (on-line analytical processing) has expanded as a simple user-friendly yet powerful analysis-oriented framework based on the multidimensional analysis metaphor (Kimball, 1998). OLAP is considered a valuable metaphor for decision making and currently most DSS adhere to OLAP and are therefore modeled according to the multidimensional model. The DW usually refers to the data repository at the core of a data warehousing system (which also refers to the ETL and exploitation tools). However, these are commonly used as synonyms.

Although the DW architecture and its main constructs were defined in the late 90s by Bill Inmon (for many, the father of data warehousing), there was still a lack for clear methodologies for developing such systems. In 1998, Ralph Kimball created the most popular methodology (Kimball, 1998) and 20 years later there are many other detailed guidelines for developing a DSS (e.g., Golfarelli & Rizzi, 2009;

Vaisman & Zimányi, 2014). All agree that designing a DSS is not the same as designing operational systems. Table 1 summarizes the main differences identified in the literature. While operational systems focus on assisting the user in day-by-day business operations (i.e., automating the operational processes), DSS support decision making by providing an accurate analysis of the business data (Wrembel & Koncilia, 2007). Accordingly, OLAP is a common choice to support innovative and unexpected queries, while OLTP (online transactional processing) is the usual choice for operational systems (whose queries are predefined and repeated continuously during the day-by-day business operations).

While DSS are used by managers and directors, operational systems are used at a lower level for conducting the company's operational processes – and often used for writing current operations (i.e., data) into the system with small transactions. The underlying databases are designed to support such workloads and are typically normalized to avoid inconsistency problems due to redundant data. In contrast, ETL is the only process allowed to write data into a DSS and nurture it with data from relevant sources (typically from within the company). Thus, DSS are considered as a decision-oriented view of existing business data, and consequently, are created as result of a reengineering process rather than from scratch (Golfarelli & Rizzi, 2009).

End-users typically query DSS by means of summaries (i.e., *GROUP BY* queries) that access thousands of tuples to compute, for instance, the average sales ratio per shop. In this kind of analysis, the temporal perspective provided by historical data (e.g., how are sales in comparison with previous months?) is essential. To deal with unexpected queries that potentially access large amounts of tuples and prove complex, DSS typically abandon the database normalization theory and systematically denormalize the data written by the ETL (e.g., by implementing a star-schema (Kimball, 1998)). This factor, together with the fact that to provide a historical perspective, data should never be deleted from the DSS, requires a large amount of disk space to store the DW.

	Operational	**Decisional**
Objective	Business Operation	Business Analysis
Main functions	Daily operations (OLTP)	DSS (OLAP)
Usage	Repetitive (predefined)	Innovative (unexpected)
Design orientation	Functionality	Subject
Kind of users	Clerk	Executives
Number of users	Thousands	Hundreds
Accessed tuples	Hundreds	Thousands
Data sources	Isolated	Integrated
Granularity	Atomic	Summarized
Time coverage	Current	Historical
Work units	Simple transactions	Complex queries
Requirements	Performance & consistency	Performance & precision
Size	Mega/Gigabytes	Giga/Tera/Petabytes

Table 1: Differences between an operational a decisional system

Current methodologies (e.g., Golfarelli & Rizzi, 2009; Kimball, 1998; Vaisman & Zimányi, 2014) claim that these differences must be considered from the very first stage of the creation of a DSS: i.e., the RE phase, which must finish by producing the design of the ETL and DW. However, the DSS/DW community has primarily focused on the design and modeling of DSS, and much less attention has been paid to how to ensure RE phase deals with such differences in a methodological manner.

2.3. Systematic mapping study

As presented in the previous subsection, DSS have grown to become an integral element of the business processes for countless companies throughout the world. These systems feed employees with knowledge and insights to make the best decisions in accordance with their needs. It is well-known that the starting point of every information system, including DSS, is the RE phase. The recognized RE framework presented in subsection 2.1 defines the major structural elements of an RE process, and provides the comprehensive fundamentals, principles, and techniques of RE for software systems. As discussed in subsection 2.2, DSS differ from operational systems in many aspects. In fact, the literature argues that the requirement process of DSS has particularities that require applying certain methodologies that differ from those used for operational systems for two reasons: firstly, traditional methodologies have been designed with operational systems in mind; and secondly, specific methodologies applicable for DSS arose as ad-hoc answers to practical needs (focusing on populating the DW while ignoring important matters such as completeness, correctness, and satisfaction of stakeholder goals) (Wrembel & Koncilia, 2007). As a result, a consensus in the community emphasized the literature indicating that the failure in DW projects lies in a poor requirements definition phase (Cabibbo & Torlone, 1998; Lehner, Albrecht & Wedekind, 1998; Paim & Castro, 2003; Vassiliadis, 2000; Giorgini, Rizzi & Garzetti, 2008). Given this, our main motivation for the systematic mapping study is to obtain an insight about how widely the area of RE for DSS has been studied. In addition, we expect that our study can establish promising future research directions with respect to this research area.

Accurate mapping requires a precise methodology. In the field of software engineering, we can find clear guides for conducting such studies (Kitchenham & Charters, 2007). Petersen et al. (2008) explain that a study of systematic mapping is a method that provides an overview of a research area in a wide and horizontal manner, while also identifying and presenting (usually in a visual manner) the quantity and type of research results that have been published. For such a purpose, the study in the following section is reported in accordance with the guidelines proposed by Petersen et al. (2008) and Kitchenham & Charters (2007) in order to reduce the risk of bias and incompleteness in the results.

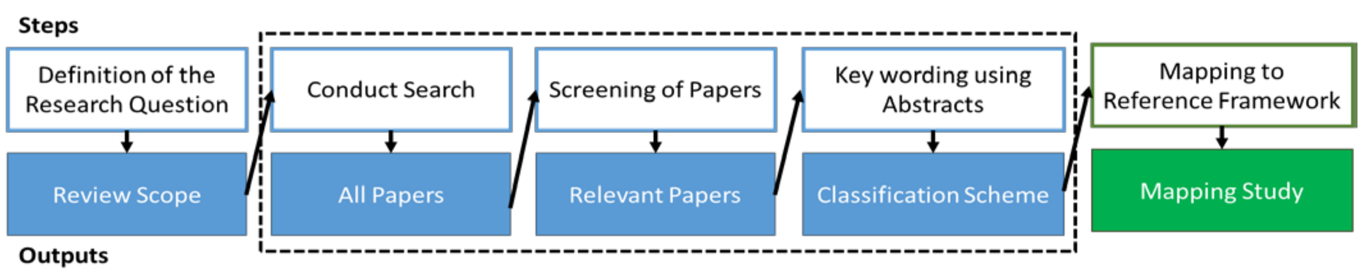

Figure 2: The systematic mapping process (adapted from Petersen et al. 2007)

As depicted in Figure 2, the initial step of the systematic mapping process proposed by Petersen et al. (2008) is to define the research question(s) (see Section 3.2 from this study). Based on the question(s), diverse search keywords are created with the objective of conducting the search from the selected databases, journals, conferences, or other sources (see Section 3.3). The documentation not matching the inclusion criteria is filtered out, and eventually only the relevant papers are taken into consideration for the final set (see Section 3.4). For the final step, we have adapted Petersen's process to map the literature to Pohl's RE framework with the objective of creating a classification (see Figure 2). In accordance with the guidelines proposed by Petersen, we have presented our output/results in tables to aid understanding.

3. Systematic Mapping Study

This section consists of three subsections prior to the results of the systematic mapping. The initial subsection provides the identification of the need for a mapping study. This is followed by the definition of the research questions. We then introduce the search query and databases used to make our search. The final subsection describes step by step the planning carried out to identify and extract the relevant scientific literature after filtering with our inclusion and exclusion criteria.

3.1. Identification of the need for a mapping study

Prior to undertaking a systematic mapping study, Kitchenham & Charters (2007) propose confirming the need for such a study. The need of a complete literature review of the RE process for DSS was identified during the initial requirements phase of a joint project between the *World Health Organization* (WHO) and *Universitat Politècnica de Catalunya* (UPC). This project is described in Raventós et al. (2015), and aimed to advance disease control by: creating a CIDSS to facilitate access to heterogeneous sources; creating interactive data across multiple levels; providing disease statistics; and creating dynamic dashboards and maps with information from infected areas worldwide. The CIDSS project involved specialized stakeholders from various domains (including health, entomologists, national authorities, and software analysts) who have complementary yet non-trivially integrable perspectives of Chagas disease. In consequence, software analysts from UPC began to deal with the highly-complex domain of Chagas disease by collecting the requirements before creating the CIDSS. Given the risk of producing ambiguous requirements leading to project failure, our research team analyzed the existing literature regarding aspects related to the RE process for DSS in order to produce complete requirements. No comprehensive study with regard to RE for DSS was found and accordingly we decided to conduct a systematic mapping study.

3.2. Definition of the research questions

After identifying the need to execute a mapping study, a well-defined plan that described the dynamics of the complete process was required. Hence, our team formulated the research questions. According to the guidelines proposed by Kitchenham & Charters (2007), the most important activity during planning is to formulate the right research question – as this will drive the entire systematic mapping. All of the subsequent phases (including planning) are highly dependent on this formulation. Given this, one main and four sub-research questions were derived from the objective of this study:

- **RQ 1: What are the models, techniques, and guidance (classification) proposed for the RE process tailored to DSS?**
  Rationale: There is a vast number of studies on RE for software systems including operational systems. However, our objective is to identify the requirements engineering practices for DSS, and classify these into *models*, *techniques*, and *guidance*. According to the degree of formalism: a *model* represents a practice that includes a documented representation of information to achieve a specific goal. A *technique* is a practice describing a method or a procedure of how a task can be accomplished. And *guidance* is a practice that provides directions on how to create a model or execute a technique in an informal manner.

Since our main research question is fairly broad, we have complemented RQ 1 with four additional questions using the building blocks from the RE framework as a defined base. The purpose is to identify how the model, techniques, and guidance proposed in RQ 1 relate to each of the building blocks from the RE framework described in Section 2.1. These are described below:

- **RQ 2: What are the RE core activities (documentation, elicitation, negotiation) best suited for DSS?**
  Rationale: the answer to this RQ will provide the classification (RQ 1) of the core activities proposed by the literature that contributes to the creation of the requirement artifacts.

- **RQ 3: What are the RE artifacts (goals, scenarios, solution-oriented requirements) best suited for DSS?**
  Rationale: the answer to this RQ will provide the classification (RQ 1) of the requirements artifacts that are extracted from the iterative activities in RQ 2.

- **RQ 4: What are the activities for managing the RE process for DSS?**
  Rationale: management activities are applied throughout the entire lifecycle of DSS, so our interest is in finding and classifying (RQ 1) management activities and understanding how these help streamline the complexity of DSS projects.

- **RQ 5: What are the activities to validate the RE process and its output for DSS?**
  Rationale: just like management, validation is applied throughout the entire lifecycle of DSS. However, the aim to also validate the output from the DSS. The answer to this RQ will provide us with a list and classification (RQ 1) and how these activities have been specifically applied for DSS regarding the specifications and fulfillment of the intended purpose of the system.

3.3. Keywording and automatic search

The search strategy used for our study involved defining both the search string (query) and the electronic databases to be queried. First, the string utilized for searching titles and abstracts was composed of a combination of the following keywords and acronyms: *decision support systems*, *DSS*, *data warehouse*, *data warehousing*, *DW*, *requirements engineering*, *RE*, *and requirements*. Figure 3 depicts the search string derived from the combination of above keywords.

((“decision support system”) ***OR*** *(“DSS”)* ***OR*** *(“data warehouse”)* ***OR*** *(“data warehousing”)* ***OR*** *(“DW”))*
AND
((“requirements engineering”) ***OR*** *(“RE”)* ***OR*** *(“requirement”))*

Figure 3: Search string

Subsequently, the automatic search encompassed those electronic databases considered as the most relevant scientific sources (Dyba et al. 2007) and likely to contain relevant studies. We used the search string on the following electronic databases: *ACM digital library*, *IEEE Xplore*, *SpringerLink*, and *ScienceDirect Elsevier*. These databases have search engines that enable us to identify the occurrence of our defined string in both the title and abstract. To guarantee up-to-date results, we took as a reference the seminal book from Kimball about building a DW (Kimball, 1998) and limited our search results to those published between 1997 (one year before Kimball’s book to include contemporary efforts) and 2014. Results are summarized in Table 2.

Electronic Database	Number
ACM Digital Library	1,507
IEEE Xplore	184
ScienceDirect – Elsevier	161
SpringerLink	432
Total initial set	**2,284**

Table 2: Results from the database search

3.4. Planning

The steps executed to conclude with a final set in accordance with the Petersen et al. (2008) systematic mapping process were the following (see summary in Figure 4):

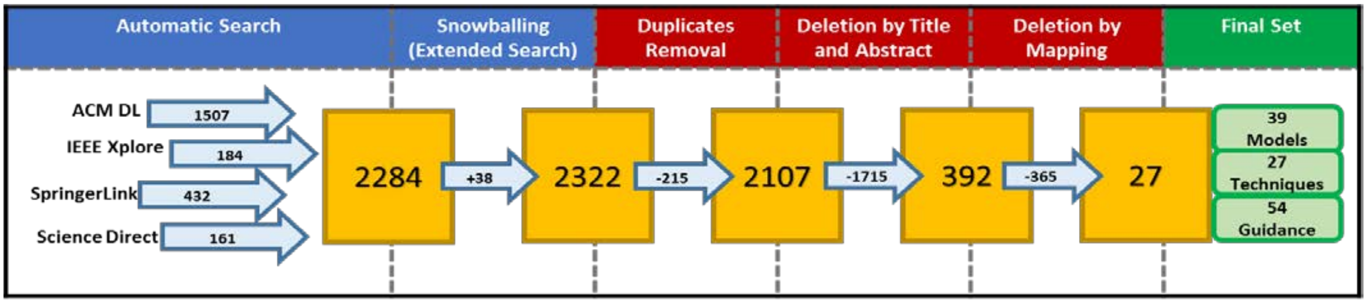

Figure 4: Stages of the study selection process

(1) **Automatic search:** the initial process was executed by our research team by exploring the published documentation considering the keywords (query) defined in the previous Section 3.3. The total initial set encompassed a sum of 2,284 documents including conferences, journals, books, and lectures.

(2) **Extended search (snowballing technique):** to systematically search for related literature, we implemented the snowballing technique suggested in (Wohlin, 2014) by using the collected literature from the previous step and iteratively extending our results by revising the citations and references. A total of 46 additional papers were identified. The full text was studied if the information was insufficient for a decision after the initial screening. We had meetings with various researchers from DSS and RE fields and aimed to reach a consensus on the paper selection. From the 46 papers, only 38 papers were added to our set. Additional keywords identified during this step were the following: requirements analysis; information requirements; information needs; OLAP; data mining; and conceptual model. These added keywords were used to filter in step (4) by title and abstract.

(3) **Complete sample:** we stopped the iteration process when we found that no new elements were being added to our list (215 repeated documents were eliminated from our initial search).

(4) **Filtering and classification:** the resulting literature was then filtered by title and abstract, including only the papers that were related to the field of DSS and/or included the selected keywords from step (2). This narrowed down our set to 392 documents. Subsequently, after analyzing the content in the remaining literature, we applied another filter that defined a set of inclusion and exclusion criteria for each retrieved study. The consensus on the selection criteria was established by researchers/experts from the DSS and RE fields. The inclusion criteria applied was:

   a. The literature described at least one model, technique, or guidance in the context of DSS
   b. The literature mapped one building-block from Pohl's (2010) framework

The exclusion criteria were the following:

   a. Studies describing the requirements process for other systems not related to DSS
   b. Studies that dealt with already structured requirements without describing the activities that produced them

As a result, 27 documents were included as our final set. The complete list of papers involved on our final set is depicted in Appendix 1. From the overall result, we have identified and classified (see Section 4 describing the classification process) the following from the 27 papers that fulfilled the selection criteria:

- 39 models
- 27 techniques
- 54 elements of guidance

(5) **Comprehensive classification:** finally, we organized and aggregated the final literature according to the building-blocks from the framework. We represented these using a literature classification table that includes the models, techniques, and guidance identified (see Section 4).

4. A Classification of DSS from an RE Perspective

This section provides the results of the systematic mapping classified (RQ 1) according to the RE framework. Each subsection described below refers back to each of the sub-research questions defined in Section 3.2. As such, subsection 4.1 presents the results of RQ 2 by defining the different core activities (4.1.1 elicitation, 4.1.2 negotiation, and 4.1.3 documentation) executed during the RE process of DSS. This is followed by subsection 4.2, which describes the requirements artifacts from RQ 3 (4.2.1 goals, 4.2.2 scenarios, and 4.2.3 solution-oriented requirements) derived from the core activities. Subsections 4.3 and 4.4 correspond to RQ 4 and RQ 5 (namely, the cross-sectional activities: management and validation). To conclude this section, subsection 4.5 presents a discussion of our findings.

In the following subsections, the mapping study results are represented in tables with the aim of providing a complete view of the practices proposed in the reviewed literature. These practices were mapped to the RE framework by Pohl (2010) and where columns represent the building blocks (requirements artifacts, core activities, validation, and management), and the rows represent the reference of the investigated literature. Inside, the various practices found within the literature are arranged according to the degree of formalism: model; technique; guidance; or a check-mark.

Degree of Formalism

Specifications for mapping study table	
Model	- Documented specification (e.g. vocabulary, conceptual model, use case, etc.)
Technique	- Procedure by which a task is accomplished (e.g. interviews, workshops, checklists, etc.)
Guidance	- Advice or information to achieve a task (identify customer needs, traceability, prioritization, etc.)
✓	- Related activity to the specific model, technique, or guidance; no further information on achievement.
->	- Activity that spans across the complete RE lifecycle (e.g. agility).

Table 3: Specifications for literature classification table

As discussed, models, techniques and guidance are categorized by the degree of formalism provided. *Check-marks* represent an implicit relationship between a model, technique, or guidance to the column of the check-mark; and finally, *crossed-lines* represent a practice that spans across all the artifacts or activities throughout the RE lifecycle.

One of the crucial characteristics of the RE framework (see Table 3) is the interrelation between activities and artifacts. Core activities interrelate between one another and contribute to the creation of artifacts, while management and validation are executed throughout the complete requirements process. Therefore, we have represented these practices in the same row in order to visualize the interrelation between activities and artifacts. For instance, an example extracted from Golfarelli & Rizzi (2009) explains that the identification of measures, dimensions, and facts *(elicitation – guidance)* will contribute to the creation of the conceptual model *(solution oriented requirement – model)* as an implicit check-mark in *documentation* for the same row (given that model must be documented). Moreover, *guidance* can stand alone, or complement the 'how-to' for creating *models* or achieving certain *techniques*. *Techniques* are usually implemented for the creation of a model. *Check-mark*s provide an implicit representation of a relationship between activities. The table results are presented in the following subsections and are organized according to the activity or artifact related to each RQ.

4.1. What are the RE core activities (documentation, elicitation, negotiation) best suited for DSS?

The RE process starts with the core activities in order to establish the vision within the existing context. These activities are then executed in an iterative manner to contribute to the creation of the requirements artifacts.

4.1.1. Elicitation

	Requirements Artifacts			Core Activities			Management	Validation
Investigated Literature	Goals	Scenarios	Solution-oriented requirements	Documentation	Elicitation	Negotiation		
(Golfarelli & Rizzi, 2009)			**Conceptual model**	✓	*Identify: measures, dimensions, facts*			
	Collaboration of designers and end users		✓	✓	*Elicit goals and limitations*		*Requirements Prioritization*	
				✓	Facilitated Sessions + Brainstorming			
				✓	Interviews			
	✓	✓	✓	✓	*Elicitation source: data processing staff*			
	✓				*Elicitation source: business users*			
				✓	Interviews			
(Kimball, 1988)				✓	Elicitation checklist			
				✓	Facilitated cessions			
				✓	Interviews			
	Identify customer needs			✓	Specialized inteviews			
				✓	Face-to-face interviews			
				✓	interviews by user			
				✓	Data audit interviews			✓
(Paim & Castro, 2003)			**Business rules specification**	✓	*Elicit business rules*			
				✓	Specialized inteviews			
			✓		*Collect data from different sources*			
			✓	*Use pre-existing operational documenta-tion*	*Data extraction & integration procedures*			
			✓	✓	*Data collection procedures*			
		Determine the extent metrics can be operated		✓	*Identify facts, metric operations, and dimension hierarchies.*			
				✓	DW requirments workshop	*Promote fast agreement about the course of actions for system delivery*		
(Stroh et al., 2011))	✓			✓	*Elicit future target requirements*			
	✓	✓			*Include middle management & decision-making competence*			
(Atkins, 2009)				✓	Brainstorm Analysis interviews Group interface Observation Reverse engineering Engineering surveys Workshops	✓		
	✓	✓			*Start w/business requirements*			
(O'Donnell et al., 2012)	✓	✓	✓	✓	prototypes			
(Britos et al., 2008)	✓	✓	✓	✓	*Data, technical resources, and security limits*			

Table 4: Elicitation activity – proposed approaches in literature

During the elicitation activity, requirements are requested from stakeholders and the goal is to gather and improve the understanding of the requirements. Table 4 shows the approaches extracted from the literature for the elicitation activity of RE in DSS. The literature reveals that three main sources are used to obtain the requirements: business users; data processing staff/information administrators; and operational sources. These are described below:

- **Business users:** business users (end-users) decide the future target requirements (Golfarelli & Rizzi, 2009; Stroh et al., 2011). DSS usually start with the business requirements and fully engage the stakeholders (Atkins, 2009). The elicitation process should not just address the employees who have the operational tasks; but, it is also important to include middle-management responsibilities and decision-making competence, such as department or team leaders (Stroh et al., 2011). It is crucial to know how to deal with the type of user from whom the requirements are elicited during the interview, for instance: overbooked users; overzealous users; know-it-alls; clueless users; and nonexistent users (Kimball, 1998).
- **Data processing staff:** a second human source is needed that includes information administrators and data processing staff, being the point of reference for designers (Golfarelli & Rizzi, 2009). The importance of this source is to extract the requirements related to the technical restrictions (such as the size of the data sources) and security matters in order to build the right DSS (Britos, Dieste & García-Martínez, 2008).
- **Data sources:** in addition to the two human sources – data in a DW environment is collected from data sources. Data may be gathered from heterogeneous sources inside and outside the enterprise (Paim & Castro, 2003). Hence, during the requirements elicitation, one should determine which sources are used (e.g. ERP, excel spreadsheets, databases, payroll systems) and how these will be acquired and consolidated in the warehouse (Paim & Castro, 2003).

After identifying the main sources to extract the requirements, it is now possible to start eliciting the relevant requirements from the identified sources. Based on the literature, the following list summarizes the requirements needed during the elicitation activity:

- **Elicit “needs” not “wants”:** most of the practices during the interview process are completed in the same manner as in operational systems. However, in DSS projects, stakeholders usually know the objective of what they want to achieve (e.g. “increase my sales 12% per year”) without knowing exactly which system they want. Therefore, the interviewer must ask: “what do you do (and why)” rather than: “what do you want?” (Kimball, 1998). This will allow the information system professionals to extract the customer needs and provide a solution rather than an unusable system. Subsequently, building prototypes will help users to confirm the elicited requirements and align these with their needs (O’Donnell, Sipsma & Watt, 2012).
- **Elicit business rules:** business rules define or constraint the aspects of a business, while they intend to control and influence its behavior (Perkins, 2000). According to the literature, business rules must be elicited to regulate the DW and the Data Mart functionalities. For instance, (Paim & Castro, 2003) remark that during the DSS development, the guidelines can be defined in terms of business rules to clarify the multidimensional requirements, the source integration premises, and the project objective.
- **Elicit data collection procedures:** Paim & Castro (2003) mention that the sources must be integrated according to the defined rules, considering the data exchange between the identified systems, periodicity, data loading, and priorities regarding what should be executed first.
- **Elicit data extraction & integration procedures:** in DW projects, specifying procedures (extraction and integration) is an important task to control how data is collected from heterogeneous sources inside and outside the enterprise environment, and how this will be integrated into a single warehouse; to achieve this, the literature suggests the analysis of existing documentation (Paim & Castro, 2003).
- **Identify data technical constraints resources & security limits:** while interviewing the data processing staff, requirement limits must be elicited, such as the data matters (access to

information sources and data quality), human and technical resources (hardware and software limitations, users), and finally security concerns (Britos et al., 2008). In addition, this phase also includes collaboration between business users and designers, aiming to achieve a common agreement to specify the goals and limitations of the DW (Golfarelli & Rizzi, 2009). According to the check-marks from Table 4, all these mentioned constraints, limitations, and procedures must be taken into account during the elicitation – given that these will directly influence the final delivery in the solution-oriented requirements to produce the right model.

- **Define/create vocabulary:** the vocabulary and glossary is one of the most addressed practices in the literature due to its great importance during the communication process, and the subsequent creation of a multi-dimensional (MD) model. The creation of the vocabulary must start with the initial phase of a DSS project to avoid linguistic inconsistencies during the elicitation process and the desires may be expressed using homogenized terms (i.e. dimensions) understood by business users, developers, and everyone involved in the project (Azvine, Cui, Nauck & Majeed, 2006). The vocabulary is explained in more detail in the solution-oriented requirements section.

After defining the sources and list of requirements needed, the literature proposes four main techniques to elicit the requirements for DSS. These are:

- **Interviews:** these usually have several preliminary activities and are well described by Kimball (1998). These activities include pre-interview research, interviewee selection, question development, interview scheduling, and preparation. Interviews are usually made with individual users or small homogeneous groups in order to achieve a more detailed list of the specifications (Golfarelli & Rizzi, 2009). Users may be asked various types of questions: open-ended; closed; and evidential (Golfarelli & Rizzi, 2009). Specialized interviews may also include straight questions tailored to the DW issues (such as granularity and multidimensionality matters) and Paim & Castro (2003) suggest using Kimball's interviewing approach. During the interview, Kimball (1998) suggests starting the conversation with business users with an easy topic, for instance their job responsibilities, vision, future objectives, and challenges. For best results, the author recommends face-to-face or voice-to-voice interviews without relying on non-interactive surveys or questionnaires.
- **Facilitated sessions:** unlike interviews which focus on small and homogeneous groups, facilitated sessions are organized in large heterogeneous groups where everyone can actively participate; the final result is a very detailed list of specifications (Golfarelli & Rizzi, 2009; Kimball, 1998).
- **DW requirements workshop:** is a specialized workshop tailored to DW that encourages consensus on the scope of the multidimensional solution (Paim & Castro, 2003). According to the author, this is done in order to achieve an agreement among all the parties about the course of actions of data warehouse/mart delivery and to capture the major functionalities and operational constraints.
- **Observation:** this technique is used to observe the company's actions and personnel activities. Repeated observation will validate the gathered facts. This technique aims to understand what the user needs are for a DW. This technique is effective for defining how individuals execute their quotidian activities and assess their working environment (Atkins, 2009).

Note from check-marks in Table 4 that all techniques during elicitation must provide a document as an output. In addition to the elicitation techniques described above, assistance techniques support the requirements elicitation of DSS. These include the following:

- **Reverse prototyping:** Atkins (2009) proposes reverse engineering to study how other DSS work and were built, and then re-create the system. According to the literature, prototypes can be useful for users to understand the requirements and system capabilities for building the DSS (O'Donnell et al., 2012).

- **Brainstorming:** a creative technique to generate ideas from a group of stakeholders depending on their needs (Atkins, 2009). Golfarelli & Rizzi (2009) remark that facilitated sessions are helpful to encourage brainstorming.
- **Elicitation checklist**: the literature recommends bringing out a list during the interview that includes the data elements in order to track what is needed (Kimball, 1998).

After identifying the relevant sources and eliciting requirements, the literature points out that from this point on, we may start identifying facts, metrics, and dimension hierarchies that will be useful and needed for the system (Paim & Castro, 2003). Golfarelli & Rizzi (2009) agree that the identification of the facts, dimensions, and measures are a crucial step before creating the conceptual design of the DW.

4.1.2. Negotiation

	Requirements artifacts			Core activities			Management	Validation
Investigated literature	Goals	Scenarios	Solution-Oriented Requireme	Documenta tion	Elicitation	Negotiation		
(Paim & Castro, 2003)			✓	**Data warehouse specification**	✓	✓	*Requirements iteration*	
			✓	**Data warehouse specification**		*Overlapping requirements*		
		Use cases		✓		*Reusability of the agreed knowledge*		Requirements conformance
				✓		✓		*Create revision report*
				✓	DW requirments workshop	*Promote fast agreement about the course of actions for system delivery*		
	Define project objectives			✓		*Agreement among all stakeholders*		
(Stroh et al., 2011)				✓		*Identify overlapping requirements*		
(Golfarelli & Rizzi, 2009)	*Investigate available sources*					*Execute negotiation strategy*		

Table 5: Negotiation activity – proposed approaches in literature

The final product must fulfill the needs and wishes of the stakeholders; however, different opinions may contradict one another. Table 5 shows the practices extracted from the literature for the negotiation activity in RE for DSS. The literature indicates that on DW projects, the objectives and project restrictions (such as aggregated information) must be agreed by all the stakeholders involved in the project (Paim & Castro, 2003). In addition, mutually dependent requirements that overlap each other must be identified and solved with a common agreement during this negotiation activity (Stroh et al., 2011). For instance: conflicting points of view regarding the set of dimensions and facts to be identified.

Accordingly, Paim & Castro (2003) claim that redundancy between requirements that oversee the entire DW must be avoided before producing the 'data warehouse specification'. The check-marks in the documentation in Table 5 imply that after achieving an agreement between stakeholders, these must be documented to promote re-usability of 'agreed knowledge' in case similar conflicts take place in the future (Paim & Castro, 2003). As seen from Table 5, negotiation activity in DSS requirements does not change significantly from other systems. Table 5 shows and in accordance with the study made by Paim & Castro (2003), negotiation is addressed by relatively few approaches, especially when describing how the needs must be prioritized. However, one of the differences between DSS and other systems is the concept of the conflict discussed during negotiation; for instance, stakeholders may not agree on the level of detail of the data: such as the dimensions (e.g. country, time, currency) or metrics (e.g. price) (Paim & Castro, 2003). This is an iterative process where requirements are elicited, and later these are negotiated between the stakeholders until agreement is reached on the specified requirements (Paim & Castro 2003).

4.1.3. Documentation

Investigated literature	Requirements artifacts			Core activities			Management	Validation
	Goals	Scenarios	Solution-oriented requirements	Documentation	Elicitation	Negotiation		
(Paim & Castro, 2003)				**Requirements Management plan Project glossary DW vision Data mart vision Use cases specification Multidimensional Requirements specification Non-functional Requirements specification Business rules specification Revision report**				
			DW - ENF framework	*Checklist for quality requirements*				
				List of actions				*When problems detected return*
(Kimball, 1988)				*Create interview summary/results*	✓			
Golfarelli et al. (2011)				*Documentation options: formal and light*				
(Golfarelli & Rizzi, 2009)				**Conceptual model**	*Identify: measures, dimensions, facts*			
(Manzon et al., 2008)				**Conceptual model**				
(Winter & Strauch, 2003)				*Requirements evaluation criteria*				*Requirements prioritization*

Table 6: Documentation activity – proposed approaches in literature

Documentation is an ongoing activity throughout the complete RE cycle. As shown in Table 6, the current literature suggests that documentation activity is at the core of requirements analysis by proposing abundant documentation for models, techniques, and assistance techniques such as: tropos; use cases; process diagrams; conceptual design; glossary; constraints; business rules; identified sources; testing methods; traceability matrices; and information maps (Paim & Castro 2003; Golfarelli & Rizzi, 2009; Britos et al., 2008; Winter & Strauch, 2003).

The focus of this core activity is the documentation and specification of the elicited requirements according to the defined documentation and specification rules. In addition, other important types of information such as interviews or decisions must also be documented.

Depending on the purpose of the documentation, the information resulting from the various requirements engineering activities are documented using representation formats and at different levels of detail (See Figure 5). All specified requirements are also documented requirements and, therefore, documented information.

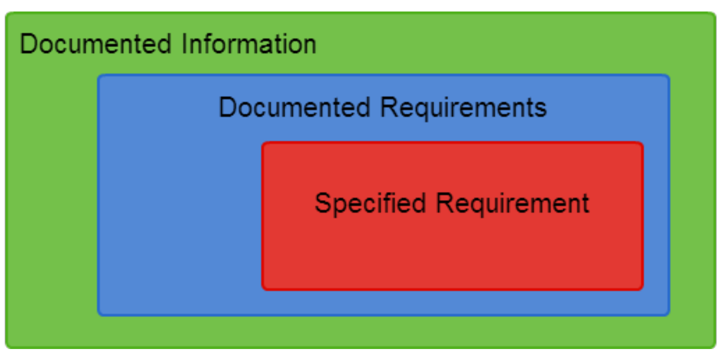

Figure 5: Documented vs specified information / requirements

However, a documented requirement is not necessarily a specified requirement. For example, informal documented interview minutes, sketches, or audio recordings may be documented information that form part of the elicitation activity and analyzed by requirements engineers or other stakeholders to document the specification requirements contained in the documented elicitation results. Therefore, a documented piece of information only represents a documented requirement if it documents information about requirements rather than other types of information, and represents a specified requirement if the documentation complies with the documentation rules and guidelines defined for requirements (Pohl, 2010).

The documentation activity goes hand in hand with the core activities (elicitation and negotiation) and the requirements artifacts: *models* for instance, by its own definition, is intended to provide “documented specifications”. Hence, all *models* must provide documentation as an output through the complete RE process; and for this reason all *models* have a check-mark in documentation. Considering that most of the documentation is very specific and is related to most activities and artifacts, we have decided to discuss all output documents resulting from such activities and artifacts in their respective sections (for instance, use cases defined in scenarios use the documentation “use cases specification”).

Nevertheless, diverse templates or meta-documents are proposed in the literature to document functional, non-functional, and domain-specific requirements that suit the description of a DW and data marts. Some of these examples provided by Paim & Castro (2003) include documenting the following:

- Requirements management plan
- Project glossary
- DW vision
- Data mart vision
- Use cases specification
- Multidimensional requirements specification
- Non-functional requirements specification
- Business rules specification
- Revision report

Golfarelli et al. (2011) explain that the documents produced may be divided into groups corresponding to project milestones. According to the authors of this work, documentation can be “formal and light” and this leads to clear and non-ambiguous-up-to-date documentation.

Moreover, according to Britos et al. (2008), the main focus of requirements documentation in the business intelligence field is to enable the achievement of consistent and traceable requirements during the entire project. The final requirements documentation “data warehouse requirements specification” containing all the requirement details and the multidimensional model will be used as a starting point for developers to create the DSS (Paim & Castro, 2003).

4.2. What are the RE artifacts (goals, scenarios, and solution-oriented requirements) best suited for DSS?

After iterating the core activities from the previous section, the output of the activities is converted to requirements called requirements artifacts. The requirements artifacts are composed of three essential types of requirements, namely: goals; scenarios; and solution-oriented requirements. These are described in the following sections:

Table 7: Goals artifact – proposed approaches in literature

	Requirements artifacts			Core activities			Management	Validation
Investigated Literature	Goals	Scenarios	Solution-Oriented Requirements	Documentation	Elicitation	Negotiation		
(Golfarelli & Rizzi, 2009)	*Collaboration of designers and end users*		✓		*Elicit goals and limitations*		*Requirements prioritization*	
	Investigate available sources					*Negotiation strategy*		
	Tropos			✓				
(Bresciani et al, 2004)	**Tropos**			✓				
(Mathur et al., 2012)	**Tropos**			✓				
(Insfrán et al., 2001)	**Function refinement tree**	**Use Cases**		✓				
(Mazón et al., 2007)	**Goal's classification**			✓				
(Kimball, 1998)	*Identify customer needs*			✓	Specialized interviews			
(Britos et al., 2008)	*Identify customer needs*							
(O'Donnell et al., 2012)	*Identify customer needs*							
(Paim & Castro, 2003)	*Customer needs + business application domain*			**Requirements management plan**				
	Granularity in data marts, constraints for MD analysis, and rules for data exchange and load.			**Requirements management plan**				
	Requirements management planning phase							
	Project objectives					*Agreement among all stakeholders*		
	High-level requirement vision			✓				
	Data warehouse vision			✓				
	Data mart vision			✓				
	Softgoal inter-dependency Graph			✓				
	Non-functional requirements framework (NFR)			✓				

4.2.1. Goals

Goals refer to the stakeholder intentions. Table 7 shows the literature study of goals in RE for DSS. According to Paim & Castro (2003), there is a lack of higher level requirements vision during the design phase and important aspects that are crucial for project success are not taken into account – including user needs, multidimensional restrictions, and quality constraints. Hence, it is crucial to gather all this information and elicit "user needs" (Kimball, 1998; Britos et al., 2008; O'Donnell et al., 2012). In addition, the literature states that the DW vision must be created in advance to describe the motivation, general objectives, project scope, stakeholder profiles, and other matters related to the DW (Paim & Castro, 2003). After eliciting the requirements from the core activities, these are then documented by using several models.

These models include:

- **Tropos**: based on i* framework. According to Mathur, Sharma & Soni (2012) this model has been properly adapted and extended to fit the DW specificities by mostly emphasizing the early requirements that are directly related to the goals that specify why the DW must be developed. The tropos methodology as proposed in Bresciani et al. (2004) is carried out to identify the business goals.

- **DW – non-functional requirements (NFR):** this framework is proposed in the work by Paim & Castro (2002) for building high quality DW specifications. This is applied for requirements such as indexing, disk space optimizers, loading schema, and other key characteristics related to DW design. This graph represents the influence of interdependency of one softgoal on another. For a softgoal to be "satisfied" there must be sufficient positive and little negative evidence against the goal.
- **Function refinement tree (FRT):** An FRT is a tree that refines the goal and purpose of a system and in which the root represents the entire system mission and the leaves represent systems functions. Insfrán, Molina, Martí & Pelechano (2001) proposed the use of the function refinement tree to organize a refinement hierarchy – the leaf nodes being the elementary functions. This gives an entry point to build the use cases.
- **Goal classification:** Mazón, Pardillo & Trujillo (2007) proposed the classification of the goals that decision makers expect to fulfill with their envisaged DW. According to this work, three kinds of goals are proposed depending on the level of abstraction:
    - *Strategic goals***:** illustrate the top level of abstraction by representing the main objectives of the business needing improvement. For example: "increase sales," "decrease expenses".
    - *Decision goals***:** illustrates the medium level of abstraction by answering the question: "how can the strategic goal be achieved?" For example, "decrease sales price" or "invest in marketing".
    - *Information goals***:** illustrates the bottom level of abstraction answering: "how can decision goals be achieved in terms of information required?" For instance: "analyze customer purchases" or "examine stocks".

  Is important to notice that after the hierarchy is defined with its levels of abstraction, the information requirements can be directly obtained from the bottom level, meaning the information goals (Mazón et al., 2007). Subsequently, facts and dimensions may be discovered from these information requirements to specify the corresponding multi-dimensional model needed for the DW.

The check-marks in Table 7 reveal that after the requirements are defined using the different goal models mentioned previously, these are then documented in the "goals documentation."

4.2.2. Scenarios:

Once the goals are elicited, scenarios are used to document specific examples of the system usage by illustrating the fulfillment or non-fulfillment of the goal. Table 8 shows the literature study of scenarios of RE in DSS. The identified scenarios to be documented were the following:

- **Use cases:** this model is one of the most cited in the literature (see Table 8) and is used to specify either textual or graphic scenarios for functional requirements, for instance UML (Paim & Castro, 2003; Insfrán et al., 2001; Molina et al., 2000; Atkins, 2009). The literature suggests that use cases start by delimiting the business processes and continue by creating a use case for each business process (Molina et al., 2000). More specifically for DSS, use cases can be used to describe the main DW functionalities such as: extract, transform and data access (Paim & Castro, 2003). Subsequently, the "use case specification document" is used to detail the procedures required to implement the functionalities of the possible sequences – and enable the reuse of behavior shared among diverse data mart scenarios (Paim & Castro, 2003).

	Requirements artifacts			Core activities			Management	Validation
Investigated literature	Goals	Scenarios	Solution-Oriented Requirements	Documentation	Elicitation	Negotiation		
(Paim & Castro, 2003)		**Use cases**		✓		*Reusability of the agreed knowledge*		Requirements conformance
		Use cases		**Use case specification**			*Reuse of common behavior scenarios*	
		Determine the extent metrics can be operated		✓	*Identify facts, metric operations, and dimension hierarchies.*			
(Insfrán et al., 2001)	**Function refinement tree**	**Use cases**		✓				
(Molina et al., 2000)		**Business / system use case**		✓				
		Sequence diagram		✓				
		Process diagram		✓				
		Workflow models		✓				
(Sellis & Simitsis., 2007)		**ETL workflow**		✓				
(Atkins, 2009)		**Use case User stories Process models Domain models Workflow Models**		✓				

Table 8: Scenarios artifact – proposed approaches in literature

- **Sequence diagram:** Molina et al. (2000) propose the use of sequence diagrams to represent diverse scenarios or objects acting in collaboration with one another to provide an outcome.
- **Process diagram:** illustrates the work-flow providing a more detailed use case. The process diagrams illustrate the different roles and data. Molina et al. (2000) propose that activities in the process diagram with a suitable level of granularity be associated with a single use case from the system.
- **Workflow model:** describes the procedural steps (business processes) that take place in the business. According to Molina et al. (2000), the workflow of the business processes can be modeled by activity diagrams that illustrate the interactions among the roles to achieve a goal. More specifically in DSS, Sellis & Simitsis (2007) propose workflows to be used as the main input to later define the representation of an ETL – including: the identification of sources; extraction of information; transformation from heterogeneous sources; cleansing of data and loading; and other activities. Such type of workflows, typically referred to as data flows, should describe how data must be gathered and transformed to fulfil a certain goal. Current practices tend to directly design ETLs and avoid representing them at a higher abstraction level (El Akkaoui, 2011). However, ETLs are solution-oriented artefacts that depend on aspects such as model transformations that cannot be described in a technology-agnostic manner. In order to improve the design and maintenance of ETLs, data flows models should be first identified and described (typically as workflow models) and be later used as an input to define ETLs.

From the check-marks in Table 8, is noticeable that most of the scenarios identified in the literature are classified as models (use cases, ETL workflows, etc.). Consequently, the outputs must be documented to provide input for the solution-oriented requirements.

4.2.3 Solution-oriented requirements:

Investigated Literature	Requirements artifacts: Goals	Requirements artifacts: Scenarios	Requirements artifacts: Solution oriented requirements	Core activities: Documentation	Core activities: Elicitation	Core activities: Negotiation	Management	Validation
(Azvine et al., 2006)			**Business vocabulary**	✓				
			Vocabulary shared ontology	✓				
(Stroh et al., 2011)			**Glossary**	✓				
(Golfarelli & Rizzi, 2009)			**Glossary**	✓				
			Conceptual model / ETL	✓	*Identify: measures, dimensions, facts*			
(Britos et al., 2008)			**Glossary**	✓				
(Molina et al., 2000)			**Glossary**	✓				
(Strauch, 2002)			**MD vocabulary**	✓				
(Winter & Strauch, 2003)			**Information map**	✓			✓	
			Conceptual model	✓				
(Kimball & Caserta, 2004)		✓	**CRUD matrix**	✓				
(Kimball, 1998)			**Bus matrix**	✓				✓
(Lee & Bryant, 2002)			**Natural language to XML**	✓				
(Paim & Castro, 2003)			**Glossary**	✓				
			Non-functional requirements specification	✓				
			Multidimensional requirements specifications	✓				
			Business rules specification	✓	*Elicit business rules*			
			DW - ENF framework	*Checklist for quality requirements*			✓	
			Determine configurable views	✓				
			MD restrictions and quality constraints	✓				

Table 9: Solution-oriented requirements artifact – proposed approaches in literature

According to our reference framework, *goals* and *scenarios* are the basic foundations for developing solution-oriented requirements. These requirements refer to the documentation of the conceptual solution that satisfies the goals and the scenarios, providing as an outcome a defined basis for the developers to create the system. Table 9 shows the literature study of RE in DSS.

One of the most highlighted practices in the literature is the creation of a *shared terminology.* This avoids linguistic inconsistencies between stakeholders and IT people, as well as homogenizing the terms used (e.g. dimensions) (Stroh et al., 2011). According to the literature, the shared terminology may be expressed and documented differently depending on the level of detail and formalism (see Figure 6):

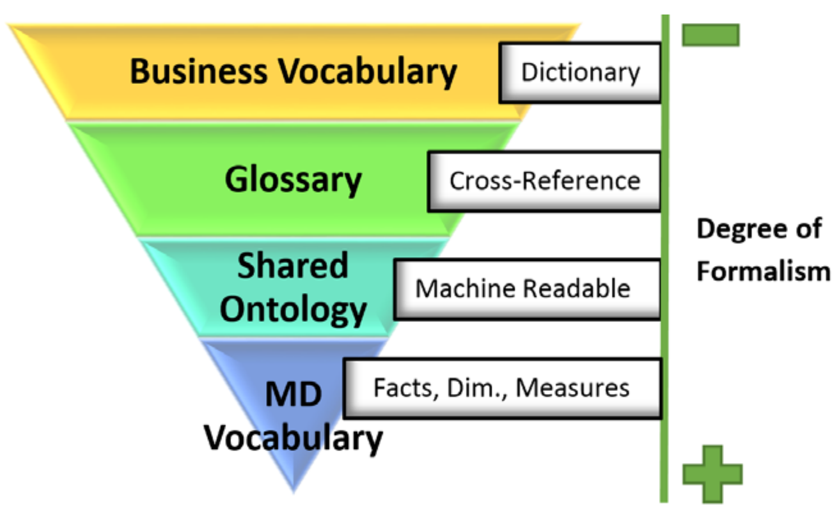

Figure 6: Shared terminology hierarchy proposed in the literature

1. **Business vocabulary:** a common "business vocabulary" is created at the highest level starting from the goals in order to provide communication between business users and IT people – and creating an integrated terminology and view of the data (Azvine et al., 2006). The output is similar to a high-level dictionary where homogenization of terms between the business and IT

terminology is understood and agreed by all the parties involved in the project.

2. **Glossary (with cross-referencing):** at a deeper level, Paim & Castro (2003) propose the use of a project glossary that organizes the DSS/BI terminology. In the glossary, the definitions are firstly identified (along with abbreviations, synonyms, etc.) to establish a general lexis that must be agreed by all parties (Britos et al., 2008). In contrast to the vocabulary, the literature proposes the use of a glossary with a cross-referenced structure to control the traceability relationships from the business processes (Molina et al., 2000). This glossary supports the designer during the creation of the conceptual and design phase (Golfarelli & Rizzi, 2009). The glossary may be represented graphically (using for instance, a UML diagram).
3. **Shared ontology:** a shared ontology is used as a common vocabulary by both parties: stakeholders and developers. This ontology is considered as a superset of the vocabulary, being sufficiently rich to describe the data sources (Azvine et al., 2006). The ontology includes terms used to denote the entities and their relationships (e.g. inheritance, attributes, and any constraints) (Azvine et al., 2006). Unlike the glossary, the ontology is created in *machine-readable format* (e.g. OWL - web ontology language); while the glossary may be just graphically represented.
4. **Multidimensional (MD) vocabulary:** at the highest degree of formalism is the MD vocabulary that focuses on the terms identified and illustrates these as multidimensional concepts (e.g. facts, dimensions, and measures) and the relationship among them (e.g. the aggregate level) (Strauch, 2002).
5. **Information map:** as one of the results of the requirements analysis, a complete and detailed document called the "information map" should be created to show the data sources (Winter & Strauch, 2003).

As the check-marks in Table 9 reveal, all the previously mentioned formalizations must be part of the documentation along with the business rules presented in the elicitation section. In addition to a common vocabulary, the literature identifies two constructs that are crucial during the warehouse design: the conceptual model and the ETLs. A DW conceptual model deals with the earliest stages of the warehouse design (Golfarelli & Rizzi, 2009). During this period, the designer is concerned about the analysis and structure, such as the entities that describe the data and the relationship between the entities. Followed by the design of the ETL, the backbone component of a DW (El Akkaoui, 2011). This process provides the DW with all the integrated data from heterogeneous and distributed data sources by extracting, transforming, and finally loading the data in the DW. Some of the approaches proposed by the literature describe the input(s) to create a DW conceptual model and the ETL: such as the bus matrix, CRUD matrix, and natural language translated to XML. These stand in the middle of the scenarios and solution-oriented requirements because they take into account the elicited goals and business processes in order to find the multidimensional semantics. However, these are still a step away from a final conceptual model/ETL. The literature describes these models as follows:

- **Bus matrix:** a formal documentation proposed by Kimball (1998) that enables users to identify the relation between measures (business processes) and dimensions (e.g. group by, and filter by). According to the literature, the bus matrix is requires the people involved in the project to visualize the key measurement events that must be analyzed. The final delivery of the bus matrix is a document with the dimension and facts providing business users with a picture of the design (Kimball, 1998). This enables an understanding of the business requirements and design to be gained.
- **CRUD matrix:** the literature proposes the use of the CRUD matrix as a helpful tool during the requirements phase to identify which tables in the database are unnecessary, as well as which tables will be heavily used, and finally, which tables are performance bottlenecks. More precisely, Kimball (2004) states that the CRUD matrix ensures that each of the entities has a process to perform: create instances (C), read (R), update (U), and delete (D).
- **Natural language:** Lee & Bryant (2002) proposed that requirements may be written in natural language, although these need to be translated into a formal specification language for execution. The author claims that after gathering the requirements, this document can be converted into

extensible markup language (XML) format in order to be machine-readable.

As a final result, a conceptual model (multidimensionally oriented) is created. The preliminary phases are crucial to create the model for the DW / data marts given the facts, dimensions, and measures identified and agreed by all the users involved (Winter & Strauch, 2003). In addition to the conceptual model, the ETL is created by using as reference the data flow model created around the scenarios, conceptual model, business rules, and sources. Paim & Castro (2003) propose storing the documentation with all the details in the "data warehouse requirements specification."

4.3. What are the activities for managing the RE process for DSS?

	Requirements Artifacts			Core Activities				
Investigated Literature	Goals	Scenarios	Solution-Oriented Requirements	Documentation	Elicitation	Negotiation	Management	Validation
(Garaibeh, 2012)	->	->	->	->	->	->	*Agile*	->
(O'Donnell et al., 2012)	->	->	->	->	->	->	*Agile*	->
(Ernst et al, 2012)	->	->	->	->	->	->	*Agile*	->
(Kimball, 1998)	->	->	->	->	->	->	*Program requirements*	->
	->	->	->	->	->	->	*Detailed project-specific requirements*	->
	✓	✓	✓	✓			**2-by-2 prioritization grid**	
(Winter & Strauch, 2003)				*Requirements evaluation criteria*			*Requirements prioritization*	
(Golfarelli & Rizzi, 2009)	*Collaboration of designers and end users*		✓		*Elicit goals and limitations*		*Requirements prioritization*	
							Integration with other systems	✓
(Stroh et al., 2011)	->	->	->	->	->	->	*Process perspective*	->
							Requirement prioritization	
				✓			*Traceability*	
(Atkins, 2009)			✓	✓			**-Prioritization matrix -Risk mgmnt plan -Change mngmt plan**	
(Atkins, 2009)				✓			*Requirements traceability*	
(Cui & Widom, 2003)				✓			*Lineage tracing for data warehouse*	
(Bębel et al., 2004).				✓			*Traceability of schema*	
(Vanhooff & Berbers, 2005)				✓			*Traceability of metadata*	
(Paim & Castro, 2003)			✓	✓			**Traceability matrix with cross-reference**	
			✓				*Traceability and change management*	
							Tools for change management	
				✓			*Requirements management control*	
			✓	**Data warehouse specification**	✓	✓	*Requirements iteration*	

Table 10: Management cross-sectional activity – proposed approaches in literature

Management is one of the two cross-sectional activities that expand through the complete RE process. The main goal of the management activity is to monitor the RE processes and detect changes, as well as managing and monitoring the execution of the requirements. Table 10 shows the literature study of the management activity in RE in DSS; crossed-lines across the table fields indicate a technique that expands throughout the complete cycle. According to the literature, in recent decades DSS projects were in "increasingly volatile business environments" (Garaibeh, 2012), meaning that organizations are demanding faster and more agile development processes that can cope with constant changes of requirements during the project lifecycle. Because of this demand, agile requirements evolution has been used due to the simplicity of changing the requirements and evaluating the consequences of the changes (Ernst et al., 2012). In recent years, according to O'Donnell et al. (2012), practitioners agree that agile style development is the most appropriate for BI systems and it has been the main approach used for many years. Table 10 shows that "agility," "process perspective" and "program requirements" expand across the complete RE life-cycle; and for this reason we have added crossed-lines across the requirements artifacts and core activities.

The management of the RE activities of DSS mainly focuses on the change management tools, traceability of requirements, prioritization, and defining the level of refinement. These are described below:

- **Control phase and management tools:** according to the work presented by Paim & Castro (2003), the "requirements management control phase" is executed when all requirements are traced, refined, and prioritized in accordance with the evaluation criteria. The authors emphasize that every change made in the requirements will affect the database model in the data mart and DW visions; and therefore most relational databases offer various tools (e.g., Microsoft SQL Server and Oracle 9i Developer Suite) to manage the affected attributes and discover the elements that were influenced by the change. These are then documented in the "change management plan" proposed by Atkins (2009).
- **Traceability of requirements:** traceability is recording the history of the tracked changes (Stroh et al., 2011). The literature states that in DW environments, traceability must be carried out starting from the requirements phase in order to measure the impact a change may have on the multidimensional design (Paim & Castro, 2003). One of the main models proposed in the literature to achieve traceability is the "traceability matrix" that shows the requirement's dependencies and provides cross-referencing between the different requirements (ex. functionalities vs. facts; facts vs. dimensional attributes) (Paim & Castro, 2003). Accordingly, Atkins (2009) proposes the use of keys for requirement traceability such as: unique numbering; key to testing; "referred to" during project; and cross referencing.
- **Provenance and lineage:** According to the literature, in addition to the traceability of requirements, DSS have specific traceability for provenance and lineage. These are the following:
    - *Traceability of sources***:** in addition to keeping the traceability of requirements, the literature mentions that in multi-source DWs, querying the sources for lineage information is sometimes impossible. Cui & Widom (2003) state that one of the reasons is that over time, sources become inaccessible, or hard to access because of changes, or that data is inconsistent. Hence, the same authors propose that a solution to improve lineage tracing of the sources is storing a copy of all source data in the warehouse; however, the storage cost may be too great.
    - *Traceability of the schema***:** this includes the traceability of the mappings to the integration schema (including schema versioning). There have been several approaches that focus on tracing changes and maintaining different versions of the schema before unification. One of these approaches is proposed by Bębel et al. (2004), in which alternative versions of the schema are stored and maintained to simulate different business scenarios.
    - *Traceability of metadata:* Vanhooff and Berbers (2005) propose storing traceability metadata to provide a history of changes caused by transformations.
- **Prioritization of requirements:** prioritization of information requirements is an essential feature. Since there are limited resources that allow only selected requirements to be covered by the DW system, an evaluation criterion is performed to assign priorities of what should be done first (Winter & Strauch, 2003). Kimball's (1998) work proposes using the two-by-two prioritization grid to rank requirements where the X-axis represents feasibility, and the Y-axis represents potential business impact. Moreover, the same work suggests that when prioritizing the requirements and the implementation process, the grid points for requirements on the upper-right should be tackled in the initial development phases.

4.4. *What are the activities to validate the RE process and its output for DSS?*

	Requirements artifacts			Core activities				
Investigated literature	Goals	Scenarios	Solution-oriented requirements	Documentation	Elicitation	Negotiation	Management	Validation
(Paim & Castro, 2003)			✓	**Data warehouse specification**				Requirements conformance
		Use cases		✓		*Reusability of the agreed knowledge*		Requirements conformance
				✓				Review sessions + Prototyping
				✓				Prototypes
				✓				OLAP interface prototypes
				✓				Requirement's checklist
				✓				*Guarantee Summaizability*
				✓			✓	*Validate derived data*
								Add external reviewers to validation process
				Create list of actions				*When problems detected return*
(Stroh et al., 2011)				✓				Prototype
				✓				Interviews
(Atkins, 2009)				✓				Quality review Peer review Customer review IT review Phase gate Project sponsor Requirement presentation
(Mazón et al., 2008)			✓	**Conceptual model**				**Normalization process**

Table 11: Validation cross-sectional activity – proposed approaches in literature

The objective of this cross-sectional activity is to avoid error propagation in the artifacts or activities, and detect these before release of the system. The literature provides two terms: a *validation* process is performed to determine whether input and output fulfill the quality criteria answering the question "Am I building the right system?" Whereas *verification* answers the question "Am I building the system right?" These two are described in Table 11 and we further elaborate below:

1. **Validation:** with the objective of validating the artifacts, Paim & Castro (2003) propose involving external reviewers in the validation process to give an unbiased point of view about the DW and its alignment with business objectives. In addition, diverse techniques are proposed by various authors to validate the activities made during the requirements analysis of a DSS (checkmarks indicate that all these activities must be documented):
   - **Prototypes and OLAP interface prototypes**: validating the requirements through prototypes is one of the most frequently cited approaches in the literature (Stroh et al., 2011). However, DSS includes several differences in comparison with operational systems: such as the validation of the multidimensional model (solution-oriented requirements). This may be carried out by an understandable prototype for the stakeholders (such as visualization graphs, dashboards, or an interface). In addition, Paim & Castro (2003) propose the use of OLAP interface prototypes to help recognize the architectural pieces and simulate OLAP behavior for stakeholders to corroborate their ideas before being allocated to the delivery of the final multidimensional solution.
   - **Review sessions:** Paim & Castro (2003) state that this activity is achieved by presenting the final requirements to all the involved parties and describing these in terms of functional, non-functional, and multidimensional aspects of the created design. According to the authors, review sessions combined with prototyping are very effective in identifying and removing design defects before becoming part of the final DW or data mart package delivered to stakeholders.
   - **Interviews:** validation of the documented information requirements may also be achieved

with interviews of IT and business staff (Stroh et al., 2011).

- **DW requirements checklist:** Paim & Castro (2003) defined a checklist for DW requirements, tailoring specific questions regarding aggregations, facts, documentation quality, completeness, etc. (see Table 12).
- **Other techniques include:** quality review; peer review; customer review; IT review; phase gate; project sponsor; and requirement presentation (Atkins, 2009).

2. **Verification:** the second approach is verifying that the multidimensional model follows the normalization process to ensure summarizability (Mazón, Lechtenbörger & Trujillo, 2008). According to Paim & Castro (2003), 'summarizability' refers to guaranteeing the correctness of the aggregated results when combining facts and dimensions. In addition, the same work that proposes the use of 'summarizability' also states that the DW application must verify the derived data. The output of the normalization process is a multidimensional model constrained to the business rules and relationships that do not violate summarizability (Mazón et al., 2008).

Validation checklist		
Item	**Validation**	**Met expectations?**
Automatic aggregation	Do all dimensional levels lead to a complete summarizability, in terms of the multidimensional model?	✓
Facts and dimensions representation	Are all stakeholder analytical needs represented in terms of a multidimensional schema?	✓
Facts and dimensions connection	Is the entire set of dimensional levels correctly associated in all levels to the basic set of facts being analyzed?	✓
Integration completeness	Are all integration requirements and procedures defined to correctly incorporate external information from source providers?	✓
Documentation quality	Do all defined documents serve as tools to accomplish all user needs under established quality standards?	✓
Requirements conformity	Can we truly "drill" across fact tables by navigating through conformed dimensions without incurring data loss or inconsistence?	✓

Table 12: Requirements checklist for data warehouse based on Paim & Castro (2003)

When problems are located during the validation process, Paim & Castro (2003) suggest that the validation team immediately attach a document with the 'list of actions' for each of the errors or defects identified. The development process then returns to the specification phase (e.g. core activities, requirements artifacts) to solve the issues.

5. Discussion

In the previous section, we provided a study of the literature of DSS from an RE perspective through a systematic mapping of the literature. The main findings worth highlighting are:

- Most of the differences found for the RE process for DSS in comparison with operational system projects are revealed during the elicitation activity (see Table 4). Some examples of elicitation practices carried out to create the DW and ETL designs include: creating a DW requirements workshop; eliciting the business rules; identifying fact dimensions and measures; collecting data extraction and procedures. However, there is a clear gap in the literature about how to generate these designs in a methodological manner from the business needs collected and documented during the RE process. Instead, designers are left to manually generate models that greatly depend on the expertise of the designers. Consequently, this manual process tends to be error-prone – and results in several rounds of elicitation and re-design to accommodate all the business needs identified. After thoroughly studying the literature, no clear solutions were found to facilitate the creation of such designs from the artefacts created during the RE process.

- The studied literature found no relevant differences between DSS and other kind of systems with regard to *system context* (environment: people, other systems, processes, etc.). Accordingly, we have not reported this building block in the previous sections.

- During the elicitation activity, three relevant sources are usually identified: *business users*, *data processing staff (IT),* and *data sources*. A closer look at the literature indicates that this last source may require integrating diverse heterogeneous sources; and this requires establishing several procedures (e.g., quality control procedures) to control how the data is collected and transformed.

- During the elicitation of DSS, it is not easy for stakeholders to express their requirements considering that business users tend to focus only on their business objectives without understanding the technology behind the system. In response to this, business needs must be extracted by asking, "*what do you do and why?*" rather than "*what do you want in your system?*". From these elicited requirements, we may start identifying the multidimensional concepts (e.g. facts, metrics, and dimensional hierarchies).

- The literature shows that the negotiation activity is addressed by relatively few approaches and usually consists of similar activities to those performed in traditional information systems. Nevertheless, the differences mentioned in the literature is the topic of the type of conflict that may appear during negotiation activity on DSS (i.e. stakeholders may not agree on the granularity of the data, facts, and dimensions needed).

- The documentation activity in DSS enables the achievement of requirements to be consistent and traceable over the entire project. In this context, plenty of models have been proposed to document goals for DSS, such as: tropos; softgoal-interdependency graph; function refinement tree; and goal classification. When documented, these models must be verified in accordance with the specified business rules. However, when creating the documentation in detail, users tend to lose the high-level vision of the DSS design.

- Several authors propose the use of *scenarios* with use cases (just as in any operational system). Other alternatives include sequence diagrams, process diagrams, workflows, etc. While use cases are traditionally used in any software system and not just in DSS, flow-oriented diagrams are crucial in DSS to represent ETL flows.

- One of the most addressed DSS failures presented in the literature is the poor existing communication between business and IT users. Given this, the literature proposes the use of a *vocabulary* from the start of the project aimed at homogenizing terms and providing better communication between IT and business users; whereas the *glossary* provides cross-references between the terms (e.g., synonyms). This process is mainly performed manually, although automation may be considered if the requirements are formalized in machine-readable format. Hence, the glossary may later be transformed into a *shared ontology* (super-set of vocabulary) to automate the creation of solution-oriented requirements by building the *multidimensional vocabulary*, which is illustrated with multidimensional concepts (e.g., facts, dimensions, and measures) and their relationships.

- During the requirements management activities in DSS, we must consider the *traceability* and *prioritization* of requirements. In addition, the literature proposes several techniques to validate the outcome (such as prototypes, OLAP prototyping, and interviews). Another way to validate the data is ensuring summarizability to guarantee the correctness of the aggregated results produced by combining facts and dimensions.

6. Threats to Validity

There are some threats to the validity of our study. They are described and detailed as follows:

- *Incompleteness of study search:* we cannot guarantee that all relevant primary studies were selected during our search. It is possible that some relevant studies were not chosen during the search process (for instance, the search keywords may have many synonyms). To mitigate this threat, we first searched the most popular electronic databases containing a large number of journals, conferences, and books in the software engineering field. Subsequently, we employed the "snowballing" technique to extend the search by considering the related literature using "cited-by" and the references from previously collected work.
- *Bias on study selection:* bias may exist on behalf of the researchers from the study selection. To mitigate this threat, we set clear inclusion/exclusion selection criteria (see subsection 3.4) for the final papers. We met various researchers from the DSS and RE fields and reached a consensus on selection criteria.
- *Inaccurate data extraction:* the extraction of data may affect the final classification of the results. To mitigate this threat, the models, techniques, and guidance extracted from the final set (27 documents) were discussed, resolved, and agreed among the researchers in workshop sessions.

7. Conclusions and Future Work

Mapping studies are being increasingly used in the area of software engineering to identify gaps in the field. This paper has presented a systematic mapping study that summarizes the existing literature regarding requirements engineering specifically for DSS. To pursue our objective, we first collected all the publications related to the topic found by querying the selected databases. From the initial set of 2284 papers, a total of 27 papers were included in this study based on a set of selection criteria. The results extracted from the selected literature were further classified and aggregated into models, techniques, and guidance according to the degree of formalisms they presented. The results obtained have enabled us to extract conclusions regarding the state-of-the-art in the field of DSS from an RE perspective. As a result, we have identified a gap in the literature on how to generate the DW design and ETL flows (the two main constructs of a DSS) in a methodological manner from the business needs.

As future work, we plan to fill the gap identified in this study. Currently, most DW and ETL designs are manually generated at the end of the RE process. However, this manual process tends to be error-prone, since it requires conducting several rounds of redesign to satisfy all the business needs. Hence, an object of further research is to automate the creation of such designs by systematically addressing the RE process and the creation of the artefacts in a structured manner so as to automatically create the respective designs in an incremental manner and considering all stakeholder needs. This will enable DW designers to perform precisely the same operation each time they follow the automated process – thereby reducing the manual re-design process with every new requirement and so decreasing human error. The interdependences and connections identified in this study during the creation and preparation of such artefacts represent an initial step towards the goal.

Acknowledgements

This work has been partially supported by the Secreteria d'Universitats i Recerca de la Generalitat de Catalunya under 2014 SGR 1534.

References

Atkins, N. (2009). Gathering and Documenting your BI Business Requirements. In *IBM Cognos Forum Asia Pacific*. Gold Coast: IBM.

Azvine, B., Cui, Z., Nauck, D. D. & Majeed, B. (2006). Real time business intelligence for the adaptive enterprise. In *Proceedings of the 8th IEEE International Conference on E-Commerce Technology and the 3rd IEEE International Conference on Enterprise Computing, E-Commerce and E-Services,* (pp. 29-29). San Francisco: IEEE.

Bębel, B., Eder, J., Koncilia, C., Morzy, T. & Wrembel, R. (2004). Creation and management of versions in multiversion Data Warehouse. In *Proceedings of the 2004 ACM symposium on Applied computing,* (pp. 717-723). Cyprus: ACM.

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F. & Mylopoulos, J. (2004). Tropos: An Agent-Oriented Software Development Methodology. In *Autonomous Agents and Multi-Agent Systems*, *8*(3), 203-236.

Britos, P., Dieste, O. and García-Martínez, R. (2008). Requirements elicitation in data mining for business intelligence projects. In *IFIP International Federation for Information Processing*, (pp. 139-150). Boston: Springer.

Cabibbo, L. & Torlone, R. (1998). A logical approach to multidimensional databases. In *Advances in Database Technology - EDBT'98,* (pp. 183-197). Berlin: Springer.

Cui, Y. & Widom, J. (2003). Lineage tracing for general Data Warehouse transformations. *The VLDB Journal—The International Journal on Very Large Data Base*s, *12(*1), 41-58.

Dyba, T., Dingsoyr, T. & Hanssen, G. (2007). Applying systematic reviews to diverse study types: An experience report. In *1st International Symposium on Empirical Software Engineering and Measurement (ESEM)*, (pp. 225-234). IEEE.

El Akkaoui, Z., Zimànyi, E., Mazón, J. N. & Trujillo, J. (2011). A model-driven framework for ETL process development. In *Proceedings of the ACM 14th International workshop on Data Warehousing and OLAP* (pp. 45-52). ACM.

Ernst, N., Borgida, A., Mylopoulos, J. & Jureta, I. (2012). Agile requirements evolution via paraconsistent reasoning. In *Advanced Information Systems Engineering* (pp. 382-397). Berlin: Springer.

Garaibeh, N. (2012). DSS Development and Agile Methods: Towards a new Framework for Software Development Methodology. *International Journal of Machine Learning and Computing, 2*(4), 438-442.

Giorgini, P., Rizzi, S. & Garzetti, M. (2008). GRAnD: A goal-oriented approach to requirement analysis in Data Warehouses. *Decision Support Systems*, *45*(1), 4-21.

Golfarelli, M. & Rizzi, S. (2009). *Data warehouse design: Modern principles and methodologies.* New York: McGraw-Hill.

Golfarelli, M., Rizzi, S. & Turricchia, E. (2011). Modern software engineering methodologies meet Data Warehouse design: 4WD. In *Data Warehousing and Knowledge Discovery,* (pp. 66-79). Berlin: Springer.

Greenspan, S., Mylopoulos, J. & Borgida, A. (1994). On formal requirements modeling languages: RML revisited. In *Proceedings of the 16th International Conference on Software Engineering*, (pp. 135-147). IEEE.

Inmon, W., Imhoff, C. & Sousa, R. (2001). *Corporate information factory*. Second Edition. New York: John Wiley & Sons.

Insfrán E., Molina P.J., Martí S. & Pelechano V. (2001). RE Applied to User Interface Conceptual Modeling. In *4to. Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes Software IDEAS 2001,* (pp. 181–192). San José: Centro de Información Tecnológica (CIT).

Kavakli, E. & Loucopoulos, P. (2003). Goal driven RE: evaluation of current methods. In *Proceedings of the 8th CAiSE/IFIP8* (pp. 16-17).

Kimball, R. (1998). *The Data Warehouse lifecycle toolkit: Expert methods for designing, developing, and deploying Data Warehouses*. New York: Wiley & Sons.

Kimball, R. & Caserta, J. (2004). *The Data Warehouse ETL toolkit.* New York: Wiley & Sons.

Kitchenham, B. & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. In *Technical report, 2.3 EBSE Technical Report*. EBSE.

Lee, B. & Bryant, B. (2002). Contextual natural language processing and DAML for understanding software requirements specifications. In *Proceedings of the 19th International Conference on Computational Linguistics*, (pp. 1-7). Stroudsburg: Association for Computational Linguistics.

Lehner, W. Albrecht, J. & Wedekind, H. (1998). Normal forms for multidimensional databases. In *Scientific and Statistical Database Management.* In *Proceedings. Tenth International Conference,* (pp. 63-72). IEEE.

Loucopoulos, P. & Karakostas, V. (1995). *System Requirements Engineering*. McGraw-Hill.

Luhn, H. (1958). A business intelligence system. *IBM Journal of Research and Development*, 2(4), 314-319.

Mathur, S., Sharma, G. & Soni, A. K. (2012). Requirement Elicitation Techniques for Data Warehouse Review Paper. *International Journal of Emerging Technologies and Advanced Engineering, 2*(11), 456-459.

Mazón, J., Lechtenbörger, J. & Trujillo, J. (2008). Solving summarizability problems in fact-dimension relationships for multidimensional models. In *Proceedings of the ACM International Workshop on Data Warehousing and OLAP (DOLAP),* (pp. 57-64), Napa Valley: ACM.

Mazón, J., Pardillo, J. & Trujillo, J. (2007). A model-driven goal-oriented requirement engineering approach for Data Warehouses. In *Advances in Conceptual Modeling–Foundations and Applications,* (pp. 255-264). Berlin: Springer.

Molina, J., Ortín, M. Moros, B., Nicolás, J. & Toval, A. (2000). Towards Use Case and conceptual models through business modeling. In *Proceedings of 19th International Conference on Conceptual Modeling ER2000,* (pp. 281-294). Salt Lake City: Springer

O'Donnell, P. A. Sipsma, S. & Watt, C. (2012). The "Hot" Issues in Business Intelligence: The View of Practitioners. *In 16th IFIP Working Group 8.3 International Conference on DSS,* (pp. 101-112). Amsterdam: IOS Press

Paim, F. R. & Castro, J. F. (2002). Enhancing Data Warehouse Design with the NFR Framework. In *5th Workshop on RE WER,* (pp. 40-57). Valencia.

Paim, F. R. & Castro, J. F. (2003). DWARF: An approach for requirements definition and management of Data Warehouse systems. In *11th IEEE International RE Conference,* (pp. 75-84). Monterey Bay, IEEE.

Perkins, A. (2000). Business rules = meta-data. In *Technology of Object-Oriented Languages and Systems, 2000,* (pp. 285-294). Washington: IEEE.

Petersen K. Feldt R., Mujtaba S. & Mattsson M. (2008). Systematic mapping studies in software engineering. In *12th international conference on Evaluation and Assessment in Software Engineering,* (pp.68-77). Italy

Pohl, K. (1994). The three dimensions of RE: a framework and its applications. *Information systems*, 19(3), 243-258.

Pohl, K. (2010). *RE: Fundamentals, principles, and techniques*. Heidelberg: Springer.

Raventós, R., García, S., Romero, O., Abelló, A. & Viñas, J. (2015). On the Complexity of RE for Decision-Support Systems: The CID Case Study. In *Business Intelligence e-Biss,* (pp. 1-38). Switzerland: Springer.

Sellis, T. K. & Simitsis, A. (2007). ETL workflows: From formal specification to optimization. In *Advances in Databases and Information Systems,* (pp. 1-11). Berlin: Springer.

Strauch, B. (2002). *Entwicklung einer Methode für die Informationsbedarfsanalyse im Data Warehousing*. (Doctoral dissertation). Universität St. Gallen, St. Gallen.

Stroh, D. I., Winter, R. & Wortmann, F. (2011). Method support of information requirements analysis for analytical information systems. *Business & Information Systems Engineering*, 3(1), 33-43.

Vaisman, A. & Zimányi, E. (2014). *Data Warehouse Systems: Data-Centric Systems and Applications*. Heidelberg: Springer.

Van Lamsweerde, A. (2001). Goal-oriented RE: A guided tour. In *RE Proceedings. Fifth IEEE International Symposium* (pp. 249-262). IEEE.

Vanhooff, B. & Berbers, Y. (2005). Supporting modular transformation units with precise transformation traceability metadata. In *ECMDA-TW Workshop, SINTEF,* (pp. 15-27). Nuremberg.

Vassiliadis, P. (2000). Gulliver in the land of data warehousing: practical experiences and observations of a researcher. In *Proceedings 2nd International Workshop on Design and Management of Data Warehouses* (p. 12). Stockholm.

Winter, R. & Strauch, B. (2003). A method for demand-driven information requirements analysis in data warehousing projects. In *Proceedings of the 36th Annual Hawaii International Conference* (pp. 231-239). Hawaii: IEEE.

Wohlin, C., (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering* (p. 38). London: ACM.

Wrembel, R. & Koncilia, C. (2007). *Data warehouses and OLAP: concepts, architectures, and solutions*. IGI Global.

Yu, E. (1997). Towards modelling and reasoning support for early-phase RE. In *RE '97* (pp. 226-235). Annapolis: IEEE.

Zave, P. (1997). Classification of research efforts in Requirements Engineering. *ACM Computing Surveys (CSUR)*, 29(4), 315-321.

Reference	Practice
(Atkins, 2009)	User Stories
	Process Models
	Domain Models
	Workflow Models
	Prioritization Matrix
	Risk Mgmnt Plan
	Change Mngmt Plan
	Analysis Interviews
	Group Interface
	Observation
	Reverse Engineering
	Engineering Surveys
	Workshops
	Quality Review
	Peer Review
	Customer Review
	IT Review
	Phase Gate
	Project Sponsor
	Requirement Presentation
	Start w/Business Requirements
(Azvine et al., 2006)	Business Vocabulary
	Vocabulary Shared Ontology
(Bębel et al., 2004).	Traceability of Schema
(Britos et al., 2008)	Data, technical resources and security limits
(Cui & Widom, 2003)	Lineage Tracing for Data Warehouse
(Garaibeh, 2012) (O'Donnell et al., 2012) (Ernst et al, 2012)	Agile
(Golfarelli & Rizzi, 2009)	Collaboration of designers and end users
	Investigate available sources
	Identify: measures, Dimensions, Facts
	Elicit goals and limitations
	Elicitation Source: business users
	Elicitation Source: Data Processing Staff
	Negotiation Stategy
	Integration with other systems
(Golfarelli & Rizzi, 2009) (Atkins, 2009)	Brainstorming
(Golfarelli & Rizzi, 2009) (Bresciani et al, 2004) (Mathur et al., 2012)	Tropos
(Golfarelli & Rizzi, 2009) (Kimball, 1998)	Facilitated Sessions
(Golfarelli & Rizzi, 2009) (Winter & Strauch, 2003) (Mazón et al., 2008)	Conceptual Model
(Insfrán et al., 2001)	Function Refinement Tree
(Kimball & Caserta, 2004)	CRUD Matrix
(Kimball, 1998)	Bus Matrix
	2-by-2 Prioritization Grid
	Facetoface Interviews
	Interviews by User
	Data Audit Interviews
	Elicitation Checklist
	Program Requirements
	Detailed project-specific requirements
(Kimball, 1998) (Golfarelli & Rizzi, 2009) (Stroh et al., 2011)	Interviews
(Kimball, 1998) (O'Donnell et al., 2012) (Britos et al., 2008)	Identify Customer Needs
(Kimball, 1998) (Paim & Castro, 2003)	Specialized Inteviews
(Lee & Bryant, 2002)	Natural Language to XML
(Mazón et al., 2007)	Goal's Classification
(Mazón et al., 2008)	Normalization Process
(Molina et al., 2000)	Business / System Use Case
	Sequence Diagram
	Process Diagram
	Workflow Models
(O'Donnell et al., 2012) (Stroh et al., 2011) (Paim & Castro, 2003)	Prototypes

Reference	Practice
(Paim & Castro, 2002)	Inter-dependency Graph
	Softgoal
(Paim & Castro, 2003)	Data Mart Vision
	Data Warehouse Vision
	Non-Functional Requirements Framework (NFR)
	Non-Functional Requirements Specification
	Business Rules Specification
	DW - ENF Framework
	Data Warehouse Specification
	Requirements Management Plan
	Use Case Specification
	Traceability Matrix with Cross-Reference
	DW Requirments Workshop
	Requirements Conformance
	Review Sessions +
	OLAP interface Prototypes
	Requirement's Checklist
	Customer Needs + Business Application Domain
	Determine the extent metrics can be operated
	Granularity in data marts, constraints for MD analysis, and rules for data exchange and load.
	High-level requirement vision
	Requirements Management Planning Phase
	Configurable Views
	MD restrictions and quality constraints
	Use pre-existing operational documenta-tion
	Checklist for Quality Requirements
	List of actions
	Project Objectives
	Collect Data from different sources
	Data Extraction & Integration Procedures
	Data Collection Procedures
	Promote fast agreement about the course of actions for system's delivery
	Overlapping Requirements
	Reusability of the agreed knowledge
	Agreement among all stakeholders
	Requirements Iteration
	Reuse of common behavior scenarios
	Identify facts, metric operations, and dimension hierarchies.
	Traceability and Change Management
	Tools for Change Management
	Requirements Management Control
	Revision Report
	Guarantee Summaizability
	Validate Derived Data
	Add external reviewers to validation process
	When problems detected return
	Elicit Business Rules
(Paim & Castro, 2003) (Insfrán et al., 2001) (Atkins, 2009)	Use Cases
(Sellis & Simitsis., 2007)	ETL Workflow
(Strauch, 2002)	MD Vocabulary
(Stroh et al., 2011)	Include middle management & decision-making competence
	Elicit Future Target Requirements
	Identify Overlapping Requirements
	Process Perspective
(Stroh et al., 2011) (Atkins, 2009)	Traceability
(Stroh et al., 2011) (Golfarelli & Rizzi, 2009) (Paim & Castro, 2003) (Britos et al., 2008) (Molina et al., 2000)	Glossary
(Vanhooff & Berbers, 2005)	Traceability of Metadata
(Winter & Strauch, 2003)	Information Map
	Evaluation Criteria
(Winter & Strauch, 2003) (Golfarelli & Rizzi, 2009) (Stroh et al., 2011)	Prioritization

Appendix Table 1: Practices (raw data) extracted from the literature