




Specifying the Process Model for Systematic Reviews: An Augmented Proposal

Pablo Becker  [GIDIS_Web, Engineering School, Universidad Nacional de La Pampa | beckerp@ing.unlpam.edu.ar]

Luis Olsina  [GIDIS_Web, Engineering School, Universidad Nacional de La Pampa | olsinal@ing.unlpam.edu.ar]

Denis Peppino  [GIDIS_Web, Engineering School, Universidad Nacional de La Pampa | denispeppino92@gmail.com]

Guido Tebes  [GIDIS_Web, Engineering School, Universidad Nacional de La Pampa | guido.tebes92@gmail.com]

Abstract

Context: Systematic Literature Review (SLR) is a research methodology intended to obtain evidence from scientific articles stored in digital libraries. SLRs can be performed on primary and secondary studies. Although there are guidelines to the SLR process in Software Engineering, the SLR process is not fully and rigorously specified yet. Moreover, it can often be observed a lack of a clear separation of concerns between what to do (process) and how to do it (methods). **Objective:** To specify the SLR process in a more detailed and rigorous manner by considering different process modeling perspectives, such as functional, behavioral, organizational and informational. The main objective in this work is specifying the SLR activities rather than their methods. **Method:** The SPEM (Software & Systems Process Engineering Metamodel) language is used to model the SLR process from different perspectives. In addition, we illustrate aspects of the proposed process by using a recently conducted SLR on software testing ontologies. **Results:** Our SLR process model specifications favor a clear identification of what task/activities should be performed, in which order, by whom, and which are the consumed and produced artifacts as well as their inner structures. Also, we explicitly specify activities related to the SLR pilot test, analyzing the gains. **Conclusion:** The proposed SLR process considers with higher rigor the principles and benefits of process modeling backing SLRs to be more systematic, repeatable and auditable for researchers and practitioners. In fact, the rigor provided by process modeling, where several perspectives are combined, but can also be independently detached, provides a greater richness of expressiveness in sequences and decision flows, while representing different levels of granularity in the work definitions, such as activity, sub-activity and task.

Keywords: Systematic Literature Review; Systematic Mapping; Process Modeling Perspectives; SPEM; Process Improvement; Software Testing Ontology

1 Introduction

A Systematic Literature Review (SLR) aims at providing an exhaustive evidence of relevant literature for a set of research questions. Initially, SLRs were conducted in Clinical Medicine (Rosenberg *et al.* 1996). Since Kitchenham issued in 2004 a technical report (Kitchenham 2004) about SLRs, the use of SLR in different scientific communities of Software Engineering (SE) has become more and more frequent for gathering evidence mainly from primary studies and, to a lesser extent, from secondary studies. The output document yielded when applying the SLR process on primary studies is called secondary study, while applying it on secondary studies is called tertiary study. To quote just a few examples, authors in Sepúlveda *et al.* (2016), Tahir *et al.* (2016), and Torrecilla-Salinas *et al.* (2016) document secondary studies on diverse topics in SE, while the authors in Garousi & Mäntylä (2016) and Kitchenham *et al.* (2010b) report tertiary studies.

Very often researchers have reused the procedures and guidelines proposed in Kitchenham (2004), which first were reviewed by Biolchini *et al.* (2005), and later updated by Kitchenham and her colleagues in 2007 (Brereton *et al.* 2007, Kitchenham & Charters 2007). More recently, by conducting a SLR, Kitchenham & Brereton (2013) evaluated and synthesized studies published by SE researchers (including different types of studies, not only primary ones) that discuss about their experiences in conducting SLR and their proposals to

improve the SLR process.

Even though SLRs have become in an established methodology in SE research, and there exist guidelines that help researchers to conduct a SLR, the SLR process itself is not fully and rigorously specified yet. **Figure 1** depicts the process specification made by Brereton and Kitchenham *et al.*, which was totally adopted or slightly adapted by the rest of

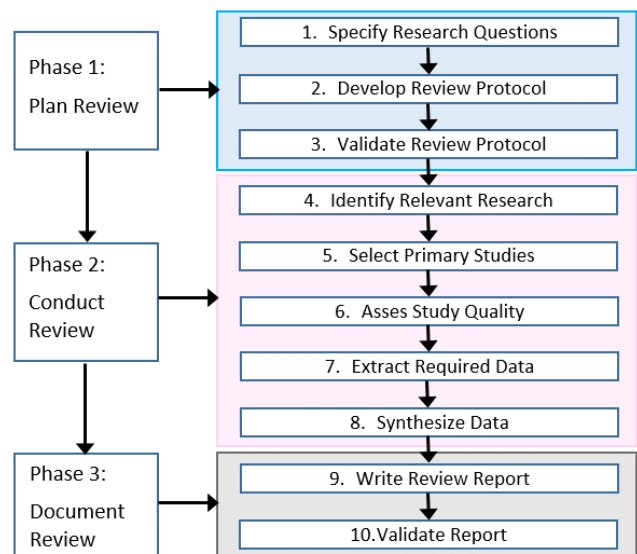


Figure 1. SLR process proposed by Kitchenham. Note that the figure's style is slightly adapted from Brereton *et al.* (2007).

the SE community up to the present time. This process specification shows ‘what’ to do through its phases and steps and in which order –or in other words, through its processes, activities and tasks as per Becker *et al.* (2015).

However, the process in **Figure 1** can be improved if we take into account the principles of process modeling proposed by Curtis *et al.* (1992), and used for instance in Becker *et al.* (2012). Curtis *et al.* describe four perspectives (views) for modeling a process:

- *Functional*: which describes what activities should be carried out and what flow of artifacts (e.g., documents) is necessary to perform the activities and tasks;
- *Behavioral*: which specifies when activities should be executed, including therefore the identification of sequences, parallelisms, iterations, etc.;
- *Organizational*: which aims at showing where and who are the agents (in compliance with roles) involved in carrying out the activities; and,
- *Informational*: which focuses on the structure of artifacts produced or consumed by activities, on their interrelations, etc.

Therefore, a full process specification considering different perspectives contributes to a clearer identification of what task/activities should be performed, in which order, by whom, and which are the consumed and produced artifacts as well as their inner structure.

In addition to these four views, a *methodological* perspective is defined in Olsina (1998), which specifies particularly what constructors (i.e., methods) are assigned to the activity descriptions. However, we detect that there is often a lack of a clear separation of concerns between what to do (process) and how to do it (methods). Consequently, sometimes methods are included as activities in the process, such as in Garousi & Mäntylä (2016), as we discuss later on.

Some benefits of using process modeling to strengthen the process specifications in general, and to strengthen the SLR process in particular, are: To *facilitate the understanding and the communication*, which it implies that the process model (with the richness that graphic representations provide) should be understandable for the target community; to *give support to the process improvement*, since all the fundamental perspectives of the process model are identified, which benefits the reutilization and the evaluation of impacts in front of potential changes in the process; to *give support to process management*, that is, to the planning, scheduling, and monitoring and control activities; to *allow the process automation*, which can help to provide supporting tools and to improve the performance; to *favor the verification and validation of the process*, fostering thus the consistency, repeatability and auditability in projects.

Additionally, in large-scale studies, like a SLR, a pilot or small-scale trial often precedes the main study in order to analyze its design validity. Therefore, it is very useful for researchers to conduct a SLR pilot to test whether aspects of the SLR design (such as the search string, selection criteria and data extraction form) are suitable. However, we observe that activities related to the pilot test are not explicitly speci-

fied in current SLR process models, as it happens in the process representation of **Figure 1**.

It is important to remark that the present paper is a significantly extended version of Tebes *et al.* (2019a). In this work, we include the organizational perspective for the SLR process, and new models from the functional and behavioral perspectives for some activities. Furthermore, in Tebes *et al.* (2019a) the study on software testing ontologies is illustrated just for the SLR pilot test. Here, we use the same study to illustrate fully the produced main artifacts throughout the SLR process. On the other hand, this work differs from Tebes *et al.* (2019b), whose focus is mainly on the analysis of the retrieved software testing ontologies but not on the process perspectives as we do in the next sections.

Summarizing, the main contribution of this work is to augment the existing SLR process specifications, considering the principles and benefits of process modeling such those described above. To this aim, we use the *functional*, *behavioral*, *informational* and *organizational* perspectives. Furthermore, we specify activities related to the SLR pilot test, which are often neglected in other SLR process specifications. As a result, the SLRs can be more systematic, repeatable and auditable for researchers and practitioners. It is worth noting that in the current work, regarding those quoted benefits of process modeling, our SLR process specifications aim primarily at facilitating the understanding and communication, as well as at giving support to the process improvement and process management. However, a thorough discussion and a detailed illustration of process modeling for supporting fully SLR process automation are out of the scope of this article.

This paper is structured as follows: Section 2 addresses related work. Section 3 specifies the proposed SLR process considering different process modeling perspectives. Section 4 illustrates a practical case applied to software testing ontologies from the process modeling perspectives standpoint. Section 5 discusses some benefits of our SLR process. Finally, Section 6 presents our conclusions and outlines future work.

2 Motivation and Related Work

One motivation for modeling the SLR process arose from certain difficulties that we faced (all the authors of this paper) when carrying out a SLR pilot study about software testing ontologies (Tebes *et al.* 2018, Tebes *et al.* 2019a). The general objective of this pilot study was to be able to refine and improve aspects of the protocol design such as the research questions, search protocol, selection and quality criteria, and/or data extraction forms. Analyzing several works about SLR, we have observed at least three main issues. First, activities related to the SLR pilot test are often omitted or not explicitly specified. Second, some aspects of the existing SLR processes are weakly specified from the point of view of the process modeling perspectives. Third, there is often a lack of a clear separation of concerns between what to do (process) and how to do it (methods). Next, we comment related work to SLR process specification where these issues were detected.

Table 1. Papers analyzed in this Related Work Section considering some features of SLR process specifications.

Paper	Features of the Process Specifications					Pilot Test activities
	Functional perspective	Behavioral perspective	Organizational perspective	Informational perspective	Used Notation	
Biolchini <i>et al.</i> 2005		✓		✓ (*)	UML	
Brereton <i>et al.</i> 2007		✓			Informal (using boxes and arrows)	
Garousi & Mäntylä 2016		✓			Informal (with legend of the used elements)	
Irshad <i>et al.</i> 2018 (**)						
Sepúlveda <i>et al.</i> 2016		✓			UML	✓
Tahir <i>et al.</i> 2016		✓			Informal (using boxes and arrows)	
Tebes <i>et al.</i> 2019a	✓	✓		✓	SPEM	✓
Torrecilla-Salinas <i>et al.</i> 2016		✓			Informal (using circles, boxes and arrows)	

* The specification of the informational perspective is represented by a text-based work-product breakdown structure.

** It has no graphical representation for the followed SLR process. However, authors adopt the Kitchenham & Charters (2007) process and the Wohlin (2014) guidelines for snowballing.

The first graphic representation of the SLR process proposed by Kitchenham (Brereton *et al.* 2007) was outlined in 2007 –taking into account previous works of the same authors (Kitchenham 2004) and other contributions such as Biolchini *et al.* (2005). It was totally adopted or slightly adapted by the rest of the SE community until to the present moment. Most of the works divide the process into three phases or stages: Plan Review, Conduct Review and Document Review. While at phase level the same three main activities are generally preserved (for example, in Sepúlveda *et al.* (2016), Tahir *et al.* (2016), Torrecilla-Salinas *et al.* (2016), to quote just a few works), at step level (sub-activities and tasks) they differ to some extent from each other. For example, in Tahir *et al.* (2016) three steps are modeled for phase 1: 1) Necessity of SLR; 2) Research questions formation; and 3) Review protocol formation. Note that these steps differ from those shown in **Figure 1**. Moreover, in Sepúlveda *et al.* (2016) five steps are presented for phase 1: 1) Goal and need of SLR; 2) Define research questions; 3) Define search string; 4) Define inclusion and exclusion criteria; and 5) Protocol Validation. The same lack of consensus in naming and including steps is observed in the abovementioned works for phase 2. Furthermore, in these works just a behavioral perspective is used to specify the process, so inputs, outputs and roles are not considered in these process models.

Although the SLR pilot test activity is usually neglected, in Sepúlveda *et al.* (2016) the “Pilot selection and extraction” step is included in phase 2. Nevertheless, the selection and pilot extraction step does not iterate into –or feedback to-phase 1, which may help to improve SLR design aspects, as we model in our proposed process specification in **Figure 2**.

In Garousi & Mäntylä (2016) and Irshad *et al.* (2018), we observe another adaptations or variations of the process documented in Brereton *et al.* (2007). In Irshad *et al.* (2018) the use of two methods called *backward* and *forward snowballing* is emphasized, while in Garousi & Mäntylä (2016) the

snowballing activity is included in the systematic review process.

Table 1 summarizes the analyzed features of the SLR processes considered in this Related Work Section.

On the other hand, it is important to remark that while SLRs are focused on gathering and summarizing evidence of primary or secondary studies, systematic mapping (SM) studies are used to structure (categorize) a research area. According to Marshall & Brereton (2013), a SM is a more ‘open’ form of SLR, which is often used to provide an overview of a research area by assessing the quantity of evidence that exists on a particular topic.

In Petersen *et al.* (2015), authors performed a SM study of systematic maps, to identify how the SM process is conducted and to identify improvement potentials in conducting the SM process. Although there are differences between SLRs and SMs regarding the aim of the research questions, search process, search strategy requirements, quality evaluation and results (Kitchenham *et al.* 2010a), the process followed in Petersen *et al.* (2015) is the same to that used for SLRs.

Therefore, we can envision that our proposed process can be used for both SLR and SM studies. What can differ, it is the use of different methods and techniques for some activities and tasks, mainly for the analysis, since as mentioned above the aim and scope for both are not the same, as also analyzed in the Napoleão *et al.* (2017) tertiary study.

In summary, as an underlying hypothesis, the existing gap in the lack of standardization of the SLR and SM processes currently used by the scientific communities can be minimized, if we would consider more appropriately the principles and benefits of process modeling enumerated in the Introduction Section.

3 Augmented Specification of the SLR Process

Considering that, there is no generalized consensus yet in the terminology used in the process domain, we introduce the meaning of some terms used in this work and then we focus on the SLR process specification. Note that the terms considered below (highlighted in *italic*) are taken from the Process Core Ontology (ProcessCO) documented in Becker *et al.* (2015).

In this work, a *process* is composed of activities. In turn, an *activity* can be decomposed into *tasks* and/or into activities of lower level of granularity called *sub-activities*. A *task* is considered an atomic element (i.e., it cannot be decomposed). Besides, process, activity and task are considered *work* (entity) *definitions*, which indicate ‘what’ to do. Every work definition (process/activity/task) consumes, and modifies and/or produces *work products*. A particular work product type is *artifact* (e.g., diagrams, documents, among others). Additionally, *methods* are *resources* that indicate ‘how’ to carry out the *description* of a work definition. In ProcessCO, many

methods may be applicable to a work description. Lastly, an *agent* is a performer assigned to a work definition in compliance with a *role*. In turn, the role term is defined as a set of skills (abilities, competencies and responsibilities) that an agent ought to own in order to perform a work definition.

Regarding the main aim of this section, **Figure 2** illustrates the proposed SLR process from the behavioral perspective using SPEM (OMG 2008). There are several process modeling languages such as BPMN (OMG 2011), SPEM and UML Activity Diagram (OMG 2017), which are the most popular in the academy and industry. Their notations are very similar considering different desirable features such as expressiveness (i.e., amount of supported workflow patterns), understandability, among others (Portela *et al.* 2012, Russel *et al.* 2006, White 2004). From the functional, behavioral and organizational perspectives, SPEM, UML and BPMN are suitable modeling languages that can be used. However, for the informational perspective, BPMN is not a suitable language. SPEM allows to use both BPMN Business Process Diagram and UML Activity Diagram, among other diagrams like the UML Class Diagram for specifying all process perspectives.

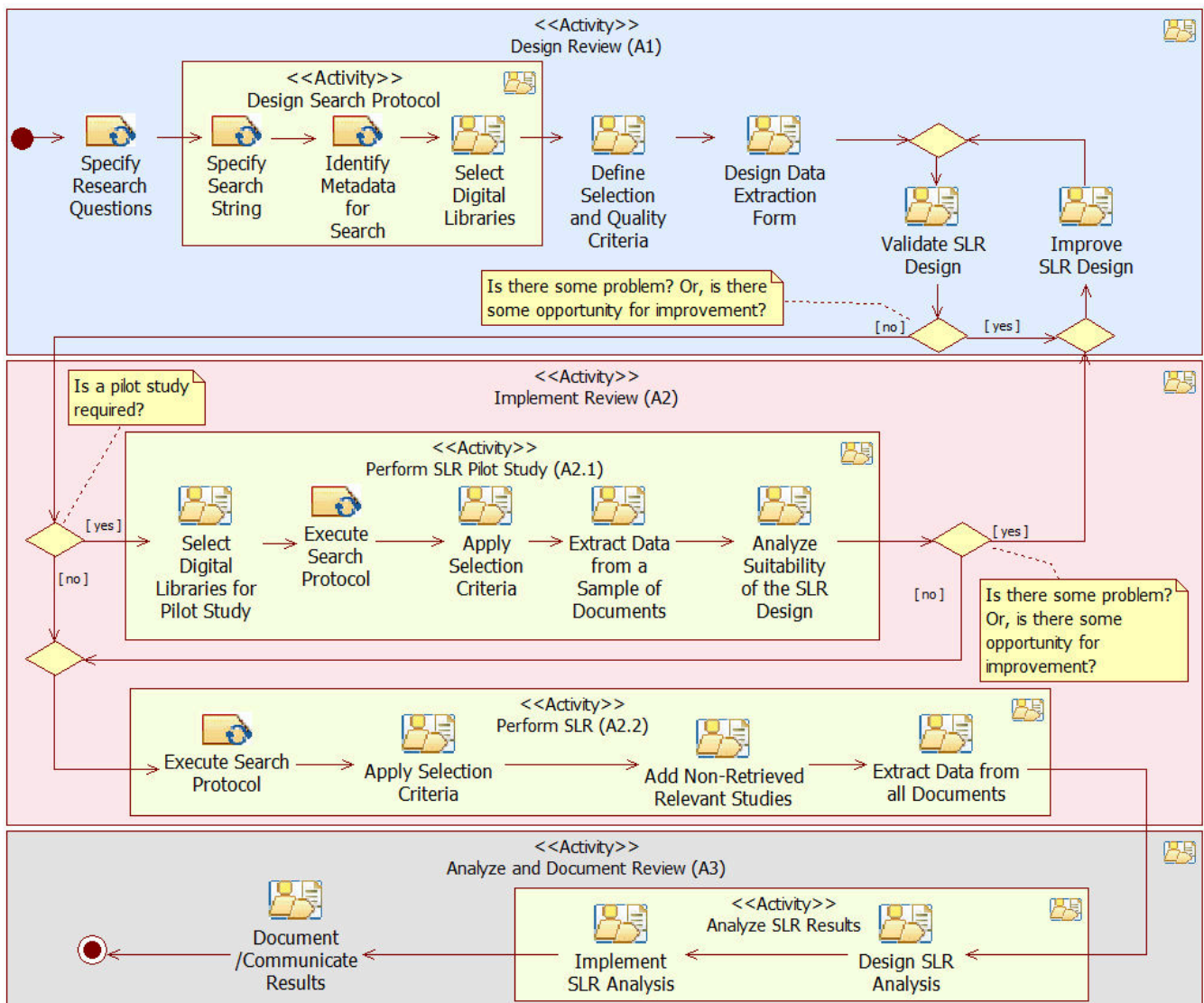


Figure 2. Behavioral perspective of the proposed SLR process.

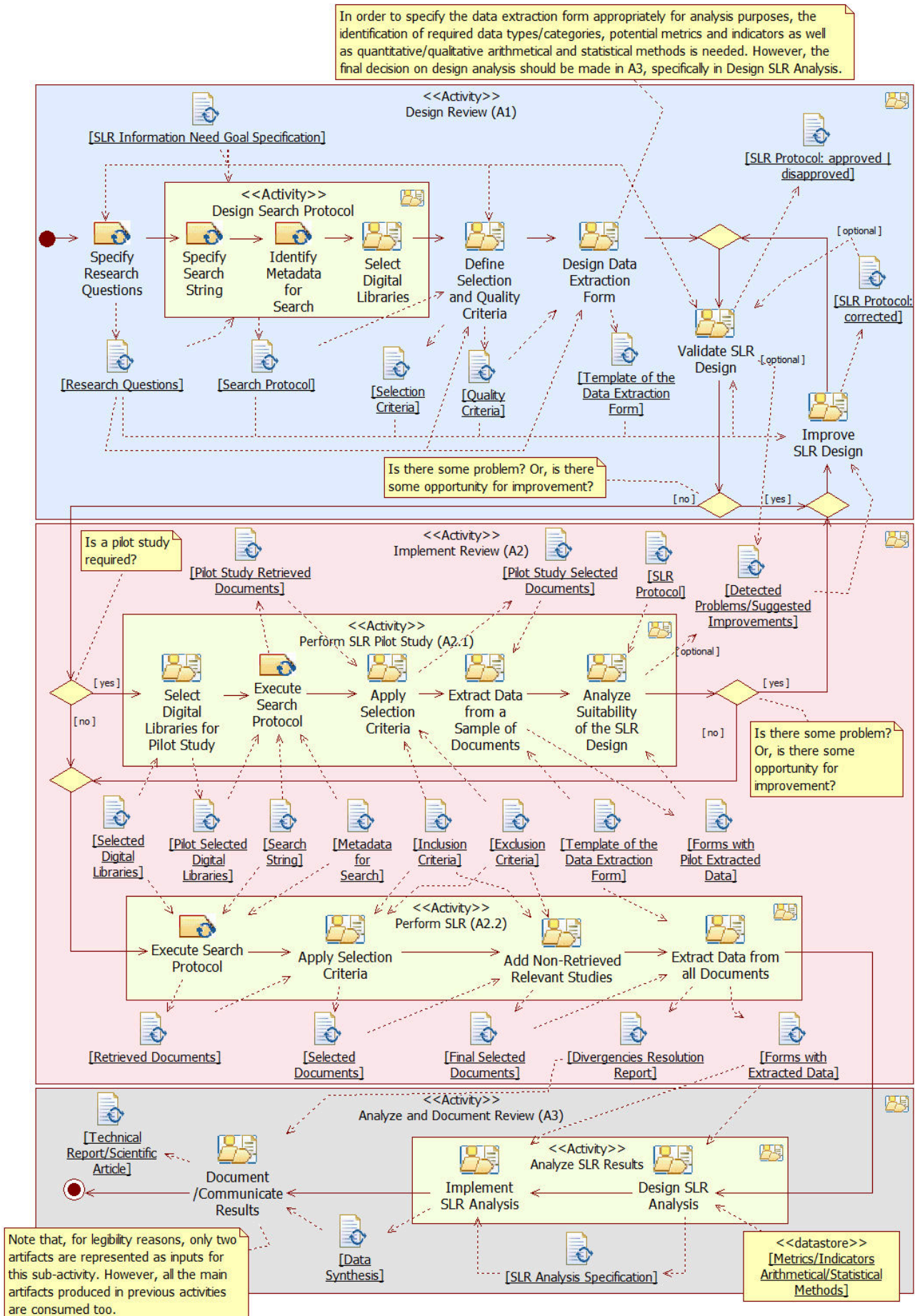


Figure 3. Functional and behavioral perspectives of the proposed SLR process.

As seen in **Figure 2**, our proposed process, like the original process (Brereton *et al.* 2007), has three main activities: (A1) Design Review, (A2) Implement Review and (A3) Analyze and Document Review. In turn, these activities group sub-activities and tasks. Note that for the Design Search Protocol sub-activity, the included tasks are shown as well, while not for the rest of the A1 sub-activities. It is done so intentionally to communicate that sub-activities have tasks, but at the same time for not giving all the details in order to preserve the diagram legibility.

As the reader can notice, our process specification has more details than other currently used models for SLRs, from the behavioral perspective standpoint. For example, we introduce decision nodes (diamonds in **Figure 2**) to represent iterations (e.g. between Validate SLR Design and Improve SLR Design in A1) and to convey that some activities/tasks could be not performed (e.g. the Perform SLR Pilot Study sub-activity in A2 is optional). Consequently, our process helps to indicate explicitly to researchers and practitioners that the SLR process is not totally sequential.

It is worth mentioning that **Figure 2** shows a recommended flow for the SLR process. In other words, it represents the "SLR-to-be" rather than the "SLR-as-is" process. We are aware that in a process instantiation there might be some variation points, including the parallelization of some tasks, and so on, as we discuss to some extent later on.

Furthermore, aimed at enriching the process specification, we consider in **Figure 3** the functional perspective joint to the behavioral perspective. Therefore, throughout the entire process, we can see the flow of activities and the work products consumed and produced in each activity/task. The functional perspective is very important to check out which documents are needed to perform a task and which documents should be produced, serving for verification purposes as well. Unfortunately, the functional perspective in the current SLR process proposals is often neglected.

Considering that a SLR is a very time-consuming endeavour that hardly can be faced by just one person, usually several researchers are involved playing different roles. The SLRs with the highest quality should have input from experts in the subject being reviewed, in the different methods for

search and retrieval, in qualitative and quantitative analysis methods, among many other aspects. Therefore, the organizational perspective to show the different roles involved in a SLR process can be used as represented in **Figure 4**.

Among these roles, A1 includes the SLR Designer (or Research Librarian) whose agent should develop comprehensive search strategies and identify appropriate libraries. Also, this role in conjunction with the Analysis Expert Designer are needed for the design of the data extraction form as well as for the definition of potential analysis methods. The SLR Validator role whose agent should have expertise in conducting systematic reviews, and the Domain Expert role, which should be played by an agent aimed at validating the protocol and clarifying issues related to the topic under investigation.

Table 2 describes the responsibilities and/or capabilities required by the different roles. Note that an agent can play different roles and, in turn, a role can be played by one or more agents (or even by a team). For example, in a given SLR the Data Collector role is frequently played by several researchers since Extract Data from a Sample of Documents and Extract Data from all Documents sub-activities are very time consuming and require a lot of effort.

In the following sub-sections, the three main activities are described considering their sub-activities and tasks, sequences, inputs and outputs, and roles, by considering the functional, behavioral and organizational perspectives. Additionally, to enrich the process specifications, in some cases, the informational perspective is used as illustrated later on.

3.1 To Design the Review (A1)

The main objective of the Design Review (A1) activity is to design the SLR protocol. To achieve this, the tasks and activities depicted in the light-blue box in **Figure 3** should be performed following the represented flow and the input and output artifacts.

As shown in **Figure 3**, the first task is to Specify Research Questions, which consumes the "SLR Information Need Goal Specification" artifact. It contains the goal purpose and the statement established by researchers, which guides the review design. Then, from the "Research Questions", the Design Search Protocol activity is carried out. This includes the

Table 2. Definitions of roles for the SLR process.

Role	Definition (in terms of Responsibility/Capability)
Analysis Expert Designer	A researcher responsible for identifying the suitable qualitative/quantitative data analysis methods and techniques to be used. The agent that plays this role should also be capable on managing documentation and visualization techniques.
Data Analyzer	Responsible for conducting the data analysis.
Data Collector	Responsible for extracting data from primary or secondary studies.
Domain Expert	A researcher or practitioner with knowledge, skills and expertise in a particular topic or domain of interest.
Expert Communicator	Researcher with rhetoric and oratory skills who communicates the SLR results to an intended community/audience.
SLR Designer	A researcher with knowledge and skills for designing and specifying SLR protocols.
SLR Expert Researcher	A researcher with knowledge and expertise in conducting SLR studies.
SLR Performer	A researcher with knowledge and skills for retrieving documents. The agent that plays this role should be expert in using search engines and document retrieval methods and techniques.
SLR Validator	A researcher with expertise in SLR for checking the suitability and validity of a SLR Design.

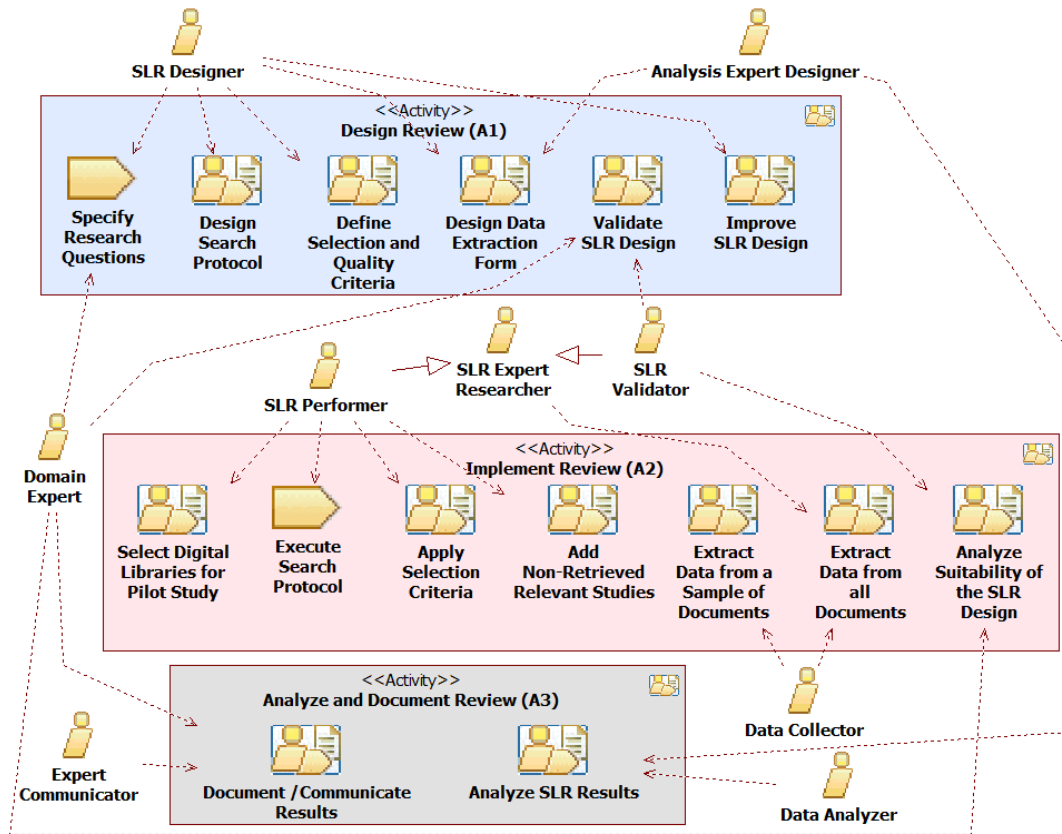


Figure 4. Organizational perspective of the proposed SLR process.

Specify Search String and Identify Metadata for Search tasks, as well as the Select Digital Libraries sub-activity. In turn, the latter includes the Define Digital Libraries Selection Criteria and Identify Digital Libraries tasks as represented in Figure 5. Examples of digital libraries selection criteria can be the target language and the library domain, among others. Digital libraries’ selection can determine the scope and validity of the reviewers’ conclusions. As a result of the Design Search Protocol activity, the “Search Protocol” is obtained, which includes a search string consisting of terms and logical operators, the metadata on which the search will be applied (e.g. title and abstract) and, the selected digital libraries (e.g. IEEE, ACM, Springer Link, Google Scholar, among others).

From the “Search Protocol” and “Research Questions” artifacts, it is possible to execute the Define Selection and Quality Criteria sub-activity. This produces the “Selection Crite-

ria” and “Quality Criteria” artifacts. The criteria can be indicated in a checklist with the different items to be considered. The “Selection Criteria” documents the set of inclusion and exclusion criteria, i.e., the guidelines that determine whether an article will be considered in the review or not. Reviewers should ask: Is the study relevant to the review’s purpose? Is the study acceptable for review? To answer these questions, reviewers formulate inclusion and exclusion criteria. Each systematic review has its own goal purpose and research questions, so its inclusion and exclusion criteria are usually unique (except for a replication). However, inclusion and exclusion criteria typically belong to one or more of the following categories: (a) study population, (b) nature of the intervention, (c) outcome variables, (d) time period, (e) cultural and linguistic range, and (f) methodological quality (Meline

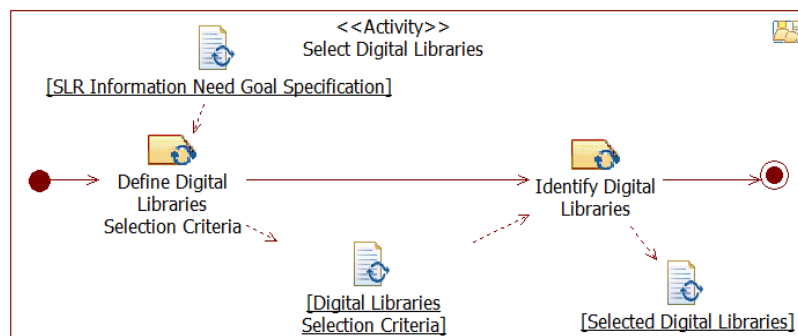


Figure 5. Functional and behavioral perspectives for the Select Digital Libraries sub-activity.

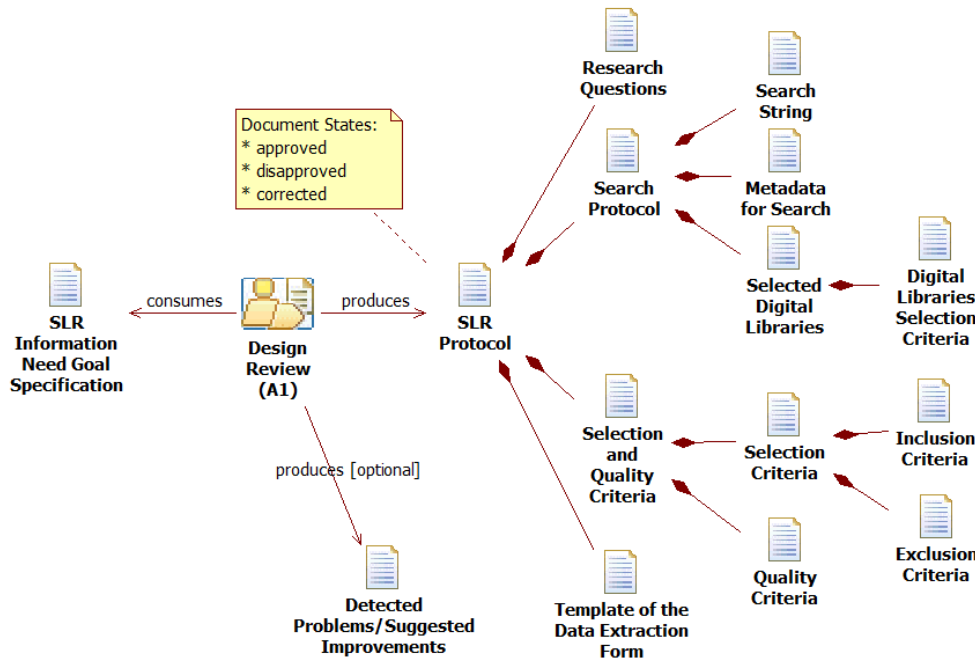


Figure 6. Informational perspective for the Design Review (A1) activity.

2006). Note that in **Figure 6** “Inclusion Criteria” and “Exclusion Criteria” artifacts are part of the “Selection Criteria” artifact.

The “Quality Criteria” documents features that allow to evaluate the quality of retrieved studies in A2, as well as to identify relevant or desirable aspects for the researchers. Sometimes, quality criteria are used like inclusion/exclusion criteria (or to build them) because are very important to select studies of high quality for deriving reliable results and conclusions (Kitchenham *et al.* 2004, Kitchenham & Charters 2007). In other cases, researchers did not plan to exclude any studies based on the quality criteria. To use “Quality Criteria” as “Selection Criteria” is a critical decision because if the inclusion criteria are too broad, poor quality studies may be included, lowering the confidence in the final result; but if the criteria are too strict, the results are based on fewer studies and may not be the yielded evidence generalizable (Lam & Kennedy 2005).

As shown in **Figure 3**, the next activity is Design Data Extraction Form. As output, the “Template of the Data Extraction Form” is yielded, whose fields are defined from the “Research Questions” and “Quality Criteria”. This will be used in A2 to collect information about each selected article. Note in **Figure 4** that this activity is performed by the SLR Designer and the Analysis Expert Designer. The former should has knowledge and skills to design and specify the data extraction form, while the latter expertise to identify required data types for analysis purposes (look at the annotation for the Design Data Extraction Form sub-activity, in Fig 3). Then, all the artifacts produced until this moment should be validated. To validate the SLR Design implies reviewing such documents in order to detect problems or opportunities for improvement. Usually, researchers with expertise in conducting SLRs perform this activity (see SLR Expert Researcher and SLR Validator definitions in **Table 2**). As out-

come, the “SLR Protocol” document is obtained, which contains all the artifacts previously produced, such as represented in the informational perspective in **Figure 6**.

Lastly, it is worth mentioning that the “SLR Protocol” document may be in an *approved*, *corrected* or *disapproved* state. In the latter case, a list of “Detected Problems/Suggested Improvements” should also be produced. This artifact will serve as input to the Improve SLR Design activity, which includes tasks such as Correct Research Questions, Correct Search String, among other tasks, in order to introduce changes in the “SLR Protocol”, i.e., to introduce corrections to improve it. Once the protocol has been corrected, the Validate SLR Design activity is performed again aimed at checking that the corrected protocol complies with the “SLR Information Need Goal Specification”. Ultimately, A1 activity ends when the “SLR Protocol” is approved.

3.2 To Implement the Review (A2)

The A2 main objective is to perform the SLR. Pink box in **Figure 3** shows the different sub-activities and tasks of A2 together with its input and output artifacts. Note that for first-time cases where a study is not a repeated or replicated one, performing first a pilot test is recommended, which is aimed at fitting the “SLR Protocol” produced in the A1 activity. Note that this concern is usually neglected or poorly specified in other existing SLR/SM processes.

When the realization of a SLR pilot study (A2.1) is taken into account, the first task to be enacted by the SLR Performer is Select Digital Libraries for Pilot Study (see **Figure 7**, which mainly emphasizes the flow of tasks and activities for the pilot study). This consists of choosing a library subset (usually one or two) from the “Selected Digital Libraries” artifact produced in A1. Then, the Execute Search Protocol task on selected libraries considering the “Search String” and the “Metadata for Search” is enacted. As outcome, a list of “Pilot Study Retrieved Documents” is produced. From this list, in

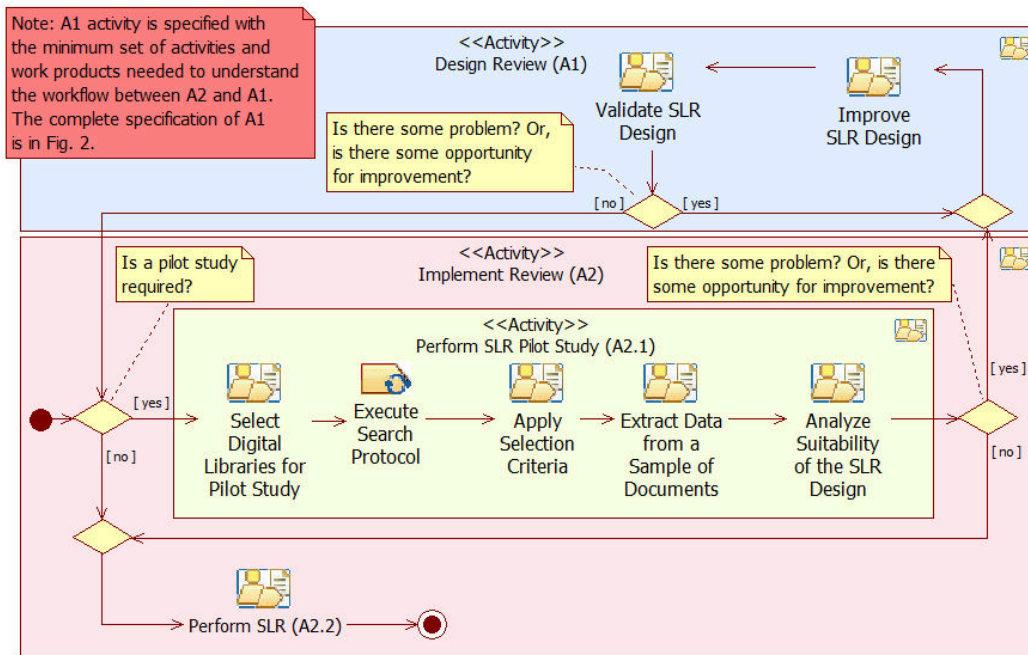


Figure 7. Behavioral perspective for the Perform SLR Pilot Study (A2.1) activity.

Apply Selection Criteria activity, the articles are downloaded and filtered out considering the “Inclusion Criteria” and “Exclusion Criteria”. This results in the “Pilot Study Selected Documents”.

From this subset of documents, the Extract Data from a Sample of Documents is done by Data Collectors (see Figure 8). This activity involves the Select Sample of Documents task, which can be done randomly (Kitchenham 2004). Then, for each document, the Extract Data from Sample task is performed by using the “Template of the Data Extraction Form”. Note that data is extracted from only one sample since the aim of the pilot test is just to analyze how suitable the protocol being followed is. If more than one Data Collector will use the forms in the final review, then it is recommended that more than one Data Collector participate in the pilot study data extraction. Testing the forms by different Data Collectors can be useful to find inconsistencies.

Finally, considering all the parts that integrate the “SLR Protocol” artifact (Figure 6) as well as the “Forms with Pilot Extracted Data”, the Analyze Suitability of the SLR Design activity is performed by the SLR Validator and the Expert Domain. This analysis permits to adjust the data extraction form in addition to other protocol aspects such as the research

questions, search string and/or selection criteria. For example, a method to validate the search string is checking if a set of known papers is recovered among the “Pilot Study Selected Documents”. When no problem is detected in the protocol, the Perform SLR activity (A2.2) is carried out. However, if a problem is detected or there is an opportunity for improvement, the Improve SLR Design and Validate SLR Design activities should be carried out again, as shown in the behavioral perspective specified in Figure 7. Once all the changes have been made and the “SLR Protocol” has been approved, the A2.2 sub-activity should be executed. Notice in Figure 7 that a new cycle of the pilot study could be performed, if were necessary.

The Perform SLR (A2.2) implies the Execute Search Protocol task taking now into account all the “Selected Digital Libraries”. The SLR Performer must Apply Selection Criteria on the “Retrieved Documents” in order to filter out those that do not meet the criteria defined in A1. As artifact, “Selected Documents” is yielded serving as input to the Add Non-Retrieved Relevant Studies sub-activity. This activity is usually performed using a citation-based searching method, for example, the forward snowballing method (i.e., finding papers that cited papers found by a search process) or the backward

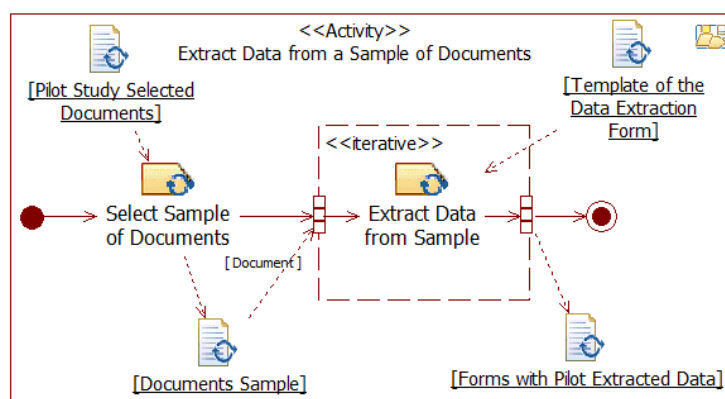


Figure 8. Functional and behavioral perspectives for the Extract Data from a Sample of Documents sub-activity.

snowballing method (i.e., looking at the references of the papers found by a search process).

At the end, the Extract Data from all Documents activity is done by using the “Template of the Data Extraction Form”. This activity is performed by one or more Data Collectors. Depending on Data Collector agents’ experience, the available resources, amount of articles, among other factors, a given article can be analyzed by one or two agents. In cases where the same article is read independently by several agents (as Data Collectors), the extracted data should be compared and disagreements be solved by consensus among them or by an additional researcher, maybe by an agent that plays the SLR Expert Researcher role. If each document is just reviewed by one Data Collector agent, for example, due to time or resource constraints, it is important to ensure that some method will be used for verifying consistency. Note in **Figure 3** that discrepancies should be recorded in the “Divergencies Resolution Report”, as also suggested by Biolchini *et al.* (2005). Once A2 is accomplished, the “Forms with Extracted Data” artifact is available for the A3 activity.

3.3 To Analyze and Document the Review (A3)

A3 is a central activity in the entire SLR/SM process. The main objective of this activity is to synthesize the analysis results based on the available scientific evidence in order to draw conclusions and communicate the findings. Furthermore, considering that a SLR should be systematic, reproducible and auditable, the continuous documentation of the followed process, applied methods and produced artifacts is a key issue. (Note that there are additional activities to those specified in **Figure 2** and **Figure 3**, in which the management of a SLR project –specifically, the project planning and scheduling- should also take into accounts such as the documentation of artifacts in all activities and control their changes and versions).

Figure 3 (in gray box) shows that Analyze SLR Results is the first sub-activity to be performed in A3. This implies in turn the Design SLR Analysis sub-activity, which is performed by an agent that plays the Analysis Expert Designer role, who is responsible for identifying the suitable data analysis methods and techniques to be used. As output, the “SLR Analysis Specification” is produced. Then, the Implement SLR Analysis sub-activity should be enacted by a Data Analyzer, who is responsible for conducting the data analysis. Analysis is carried out looking at the “Forms with Extracted Data” and “SLR Analysis Specification” artifacts. In analysis, diverse measurement, evaluation, categorization and aggregation methods of data as well as visualization means (such as tables, charts, word clouds, among others) can be used in order to give answer to the established research questions, e.g. to address the findings of similarities and differences between the studies, among many others. As a result, the “Data Synthesis” artifact is produced. The synthesis is usually descriptive. However, sometimes, it is possible to supplement a descriptive synthesis with quantitative summaries through meta-analysis, using arithmetical and statistical techniques appropriately.

Finally, an Expert Communicator carries out the Docu-

ment/Communicate Results sub-activity. To this end, dissemination mechanisms are first established, for example, technical reports, journal and conference papers, among others. Then, the documents that convey the results to the intended community are produced. In this way, the SLR process concludes. All the collected evidence and summarizations might be publicly available for auditability reasons.

4 Application of the proposed SLR Process on Primary Studies

In Olsina & Becker (2017), a family of evaluation strategies is presented. Any strategy in this approach, regardless if it is devoted to evaluation, testing, development or maintenance goal purposes, should integrate a well-established terminology, process and method specifications. To broaden the family, we are in the way of the development of software testing strategies. So first, our current aim is to build a software testing domain terminology. In this direction, in Tebes *et al.* (2019a) a SLR on software testing ontologies was designed and documented, i.e., the A1 and A2.1 activities were carried out. Then, in Tebes *et al.* (2019b), we documented the whole study including the A1-A3 activities. The result allows us to establish the grounds for developing a top-domain software testing ontology that should be integrated into the conceptual framework already developed for existing evaluation strategies.

In order to illustrate the proposed SLR process, next, we introduce the rationale for our research to contextualize the SLR study on software testing ontologies described later on.

4.1 Rationale for the SLR Study on Software Testing Ontologies

A strategy is a core resource of an organization that defines a specific course of action to follow, i.e., specifies what to do and how to do it. Consequently, strategies should integrate a process specification, a method specification, and a robust domain conceptual base (Becker *et al.* 2015). This principle of integratedness promotes, therefore, knowing what activities are involved, and how to carry them out by means of methods in the framework of a common domain terminology. In Olsina & Becker (2017), to achieve evaluation purposes, a family of strategies integrating the three-abovementioned capabilities is discussed. The conceptual framework for this family of evaluation strategies is called C-INCAMI v.2 (*Contextual-Information Need, Characteristic Model, Attribute, Metric and Indicator*) (Becker *et al.* 2015, Olsina & Becker 2018).

This conceptual framework was built on vocabularies or terminologies, which are structured in ontologies. **Figure 9** depicts the different C-INCAMI v.2 conceptual components or modules, where the gray-shaded ones are already developed. The ontologies for Non-Functional Requirements (NFRs), NFRs view, Functional Requirements (FRs), business goal, project, and context are defined in Olsina & Becker (2018), while for measurement and evaluation are in Becker *et al.* (2015). The remainder ontologies (for testing, development and maintenance) are not built yet.

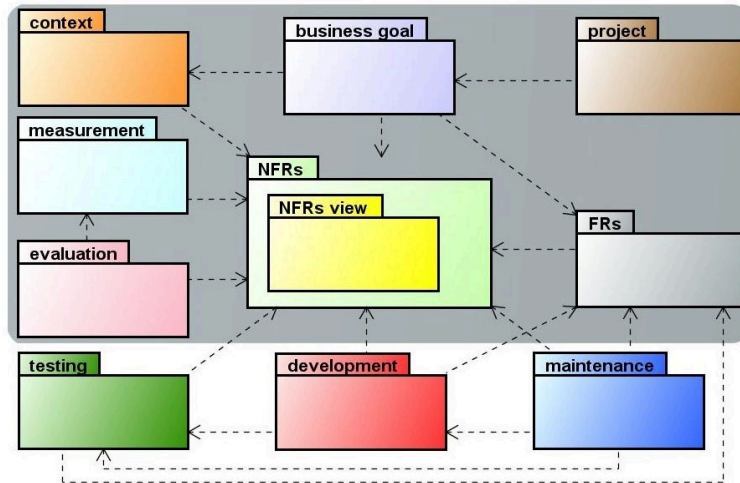


Figure 9. Conceptual components (modules) and their relationships for the C-INCAMI v.2 framework. Note that NFRs stands for Non-Functional Requirements while FRs stands for Functional Requirements.

Bearing in mind that there are already integrated strategies that provide support for achieving evaluation purposes, the reader can assume that strategies that provide support for achieving testing purposes are feasible to be developed as well. Given that a strategy should integrate a well-established domain terminology, therefore, a well-specified testing strategy should also have this capability for the testing domain. A benefit of having the suitable software testing ontology is that would minimize the heterogeneity and ambiguity problems that we currently observe in the different concepts dealing with testing methods and processes.

In this direction, we conducted the SLR study on software testing ontologies (Tebes et al. 2019b) in order to establish the suitable top-domain testing ontology to be integrated into the C-INCAMI v.2 conceptual framework. That is, we envision populating the testing conceptual component shown in **Figure 9** and linking it with the FRs and NFRs components. Next, we illustrate the A1-A3 activities presented in Section 3 using excerpts of the Tebes et al. (2019b) study. Members of the GIDIS_Web research group between the end of May and the beginning of August 2018 performed the A1 and A2.1 activities, while members of the ORT Uruguay research group helped in this stage to the validation of the SLR protocol. Then, members of both research groups between the end of October 2018 and the beginning of March 2019 performed the A2.2 activity.

4.2 To Design the Review (A1) for the Software Testing Ontologies Study

As observed in the functional and behavioral perspectives in **Figure 3**, to start A1, the “SLR Information Need Goal Specification” is required. In this case, the information need establishes that papers documenting software testing ontologies from digital libraries must be systematically analyzed.

From the main goal established for this SLR, two “Research Questions” were initially formulated, namely: (RQ1) What are the existing ontologies for the software testing domain? And, (RQ2) What are the relevant concepts, their relationships, attributes and constraints or axioms needed to describe the software testing domain? To answer RQ1 will allow us to identify and analyze the different existing software

testing ontologies. The RQ2 will serve us to know the terms (or concepts), their relationships, attributes or properties and restrictions needed to specify an ontology for the testing domain.

Then, the “Search Protocol” was designed. Taking into account RQ1, the following search string was initially proposed: “Software Testing” AND (“Ontology” OR “Conceptual Base”). For this particular study, the search string was applied on the three selected metadata, namely: *title*, *abstract* and *keywords*. (Note that the search string could also be applied to the full-text). Finally, the selected digital libraries included in the revision were *Scopus*, *IEEE Xplore*, *ACM Digital Library*, *Springer Link* and *Science Direct*.

For the “Selection and Quality Criteria” defined in this case study, see WP 1.3 in Table 3. Then, based on the research questions and the quality criteria, a set of fields for the data extraction was defined by the SLR Designer and the Analysis Expert Designer (see WP 1.4 in **Table 3**). For example, to extract terms, properties, relationships and axioms from each article, the “Relevant concepts used to describe software testing domain” field was specified as follows (not shown in WP 1.4 of **Table 3**):

Terms:

TermN: definition

Attributes:

TermN (AttributeN.1=definition, AttributeN.2=definition, ...)

Relationships:

is_a (TermX_type, TermY_subtype)
 part_of (TermX_whole, TermY_part)
 RelationM_name (TermX, TermY): definition
 RelationZ_without_name (TermX, TermY): definition

Axioms or restrictions:

Axiom: definition/specification

The “Data Extraction Form” allows obtaining homogeneity between the data extracted from each document and thus facilitating the task of analyzing them.

Once validated the produced artifacts, as result of A1, the “SLR Protocol” was obtained. **Table 3** shows all the artifacts

Table 3. “SLR Protocol” artifact produced in A1 for the software testing ontologies study.

Research Questions (WP 1.1) –Note that WP stands for <i>Work Product</i>	
RQ1: What are the existing ontologies for the software testing domain?	
RQ2: What are the relevant concepts, their relationships, attributes and constraints or axioms needed to describe the software testing domain?	
Search Protocol (WP 1.2)	
Search String (WP 1.2.1)	"Software Testing" AND ("Ontology" OR "Conceptual Base")
Metadata for Search (WP 1.2.2)	Title; Abstract; Keywords
Selected Digital Libraries (WP 1.2.3)	Scopus, IEEE Xplore, ACM Digital Library, Springer Link and Science Direct
Selection and Quality Criteria (WP 1.3)	
Inclusion Criteria (WP 1.3.1)	1) That the work be published in the last 15 years; 2) That the work belongs to the Computer Science area; 3) That the work documents a software testing ontology; 4) That the document is based on research (i.e., it is not simply a "lesson learned" or an expert opinion).
Exclusion Criteria (WP 1.3.2)	1) That the work be a prologue, article summary or review, interview, news, discussion, reader letter, or poster; 2) That the work is not a primary study; 3) That the work is not written in English.
Quality Criteria (WP 1.3.3)	1) Is/Are the research objective/s clearly identified? 2) Is the description of the context in which the research was carried out explicit? 3) Was the proposed ontology developed following a rigorous and/or formal methodology? 4) Was the proposed ontology developed considering also its linking with Functional and Non-Functional Requirements concepts?
Template of the Data Extraction Form (WP 1.4)	
Researcher name; Article title; Author/s of the article; Journal/Congress; Publication year; Digital library; Name of the proposed ontology; Relevant concepts used to describe software testing domain; Methodology used to develop the ontology; Research context; Research objective/s; Does the proposed ontology consider its linking with Functional and Non-Functional Requirements concepts?; Additional notes.	

that integrate this document, which correspond to those specified in the informational perspective of **Figure 6**.

4.3 To Implement the Review (A2) for the Software Testing Ontologies Study

4.3.1 To Perform the SLR Pilot Study (A2.1)

Looking at the activity flow shown in the behavioral perspective of **Figure 2**, we conducted a pilot study for analyzing the suitability of the “SLR Protocol”. As part of the A2.1 execution, from the “Selected Digital Libraries” in A1 (see WP 1.2.3 in **Table 3**), *Scopus* was selected for this pilot test because it contains digital resources from various sources such as Elsevier, IEEE Xplore and ACM Digital Library. As result of carrying out the Execute Search Protocol and Apply Selection Criteria activities, 19 documents were obtained (Tebes *et al.* 2018), which were reviewed by three Data Collectors of the GIDIS_Web research group to Extract Data from a Sample of Documents (recall **Figure 8**). Once A2.1 was completed, a list of “Detected Problems/Suggested Improvements” was produced and the Improve SLR Design activity was run (as prescribed in **Figure 3** through the functional and behavioral perspectives).

Table 4 shows the updates (highlighted in blue and underlined) that the “SLR Protocol” underwent after the pilot study. Next, some changes are described considering the

“Detected Problems/Suggested Improvements” document.

In the RQ1 research question, the “*existing*” term was replaced by the “*conceptualized*” term. The former term is broader than the latter including conceptualizations and implementations of software testing ontologies. However, our main goal is retrieving conceptualized ontologies regardless whether they are implemented or not.

On the other side, the “*relevant*” term in the RQ2 research question in **Table 3** influenced negatively the number of terms extracted by each Data Collector. Therefore, the research question was reformulated as observed in the WP 1.1 in **Table 4**. In addition, the “*Relevant concepts used [...]*” field in the form (see WP 1.4) was changed for “*Specified concepts [...]*”. This change made the extraction more objective and easier to interpret than with the initial design.

Moreover, the full reading of articles during the pilot study allowed us to detect that ontologies of various types were presented, such as foundational ontology, top domain ontology and domain ontology. Since the final aim after executing the SLR was to adopt, adapt or build a new top-domain ontology, this information turned out relevant. Consequently, a new research question (RQ3 in the WP 1.1 in **Table 4**) and the “*Classification of the proposed ontology*” field in the “Template of the Data Extraction Form” (see WP 1.4 in **Table 4**) were added.

Table 4. The new “SLR Protocol” version after the Pilot Study (A2.1) activity was performed. (Note that changes are indicated in blue and underlined w.r.t. that information shown in **Table 3**).

Research Questions (WP 1.1) –Note that WP stands for <i>Work Product</i>	
RQ1:	What are the <u>conceptualized</u> ontologies for the software testing domain?
RQ2:	What are the <u>most frequently included</u> concepts, their relationships, attributes and axioms needed to describe the software-testing domain?
RQ3:	<u>How are existing software testing ontologies classified?</u>
Search Protocol (WP 1.2)	
Search String (WP 1.2.1)	("Software Testing" <u>OR "Software Test"</u>) AND ("Ontology" <u>OR "Ontologies"</u>)
Metadata for Search (WP 1.2.2)	Title; Abstract; Keywords
Selected Digital Libraries (WP 1.2.3)	Scopus, IEEE Xplore, ACM Digital Library, Springer Link and Science Direct
Selection and Quality Criteria (WP 1.3)	
Inclusion Criteria (WP 1.3.1)	<ol style="list-style-type: none"> 1) That the work be published in the last 15 years (<u>from the beginning of 2003 until November 12, 2018</u>); 2) That the work belongs to the Computer Science area <u>or to the Software/System/Information Engineering areas</u>; 3) That the document <u>has the ontological conceptualization of the testing domain (i.e., it is not simply a "lesson learned or expert opinion" or just an implementation)</u>.
Exclusion Criteria (WP 1.3.2)	<ol style="list-style-type: none"> 1) That the work be a prologue, article summary or review, interview, news, discussion, reader letter, poster, <u>table of contents or short paper (a short paper is considered to that having up to 4 pages size)</u>; 2) That the work is not a primary study; 3) That the work is not written in English; 4) <u>That the work does not document a software testing ontology</u>; 5) <u>That the ontology presented in the document be an earlier version than the most recent and complete one published in another retrieved document</u>; 6) <u>That a same document be the result of more than one bibliographic source (i.e., it is duplicated)</u>; 7) <u>That the conceptualized ontology in the current document be a fragment of a conceptualized ontology in another retrieved document</u>.
Quality Criteria (WP 1.3.3)	<ol style="list-style-type: none"> 1) Is/Are the research objective/s clearly identified? 2) Is the description of the context in which the research was carried out explicit? 3) Was the proposed ontology developed following a rigorous and/or formal methodology? 4) Was the proposed ontology developed considering also its linking with Functional and Non-Functional Requirements concepts? 5) <u>What other terminologies of the software testing domain were taken into account to develop the proposed ontology?</u>
Template of the Data Extraction Form (WP 1.4)	
Researcher name; Article title; Author/s of the article; Journal/Congress; Publication year; Digital library; Name of the proposed ontology; <u>Specified</u> concepts used to describe software testing domain; Methodology used to develop the ontology; <u>Terminologies or Vocabularies taken into account to develop the proposed ontology</u> ; <u>Classification of the proposed ontology</u> ; Research context; Research objective/s <u>related to software testing ontologies</u> ; Does the proposed ontology consider its linking with Functional and Non-Functional Requirements concepts?; Additional notes.	

Also the search string was modified slightly (compare WP 1.2.1 in **Table 3** and **Table 4**) because not all search engines take into account variations or synonyms of the used words. The inclusion criterion 1 in the WP 1.3.1 (**Table 3**) is not very specific; therefore, it was modified as observed in the WP 1.3.1 of **Table 4**. The full reading of articles also permitted to detect that some of them were different versions (or fragments) of the same ontology. Therefore, exclusion criteria 5 and 7 were added (see WP 1.3.2 in **Table 4**).

On the other hand, since the searches in Scopus retrieve documents that belong to other digital libraries, exclusion criterion 6 of the WP 1.3.2 was added to eliminate duplicates.

Finally, we also observed that some ontologies were built

taking into account other terminologies, which may add a quality factor to the new proposal. For this reason, quality criterion 5 was added in the WP 1.3.3 of **Table 4**, which implies a new field in the “Template of the Data Extraction Form” (see “*Terminologies or Vocabularies taken into account [...]*” in WP 1.4). This new quality criterion may prove to be useful information in the construction process of any ontology. The reader can check the final “Template of the Data Extraction Form” artifact in Appendix A.

4.3.2 To Perform the SLR (A2.2)

Six agents (four researchers from GIDIS_Web and two researchers from ORT Uruguay) performed the A2.2 activity.

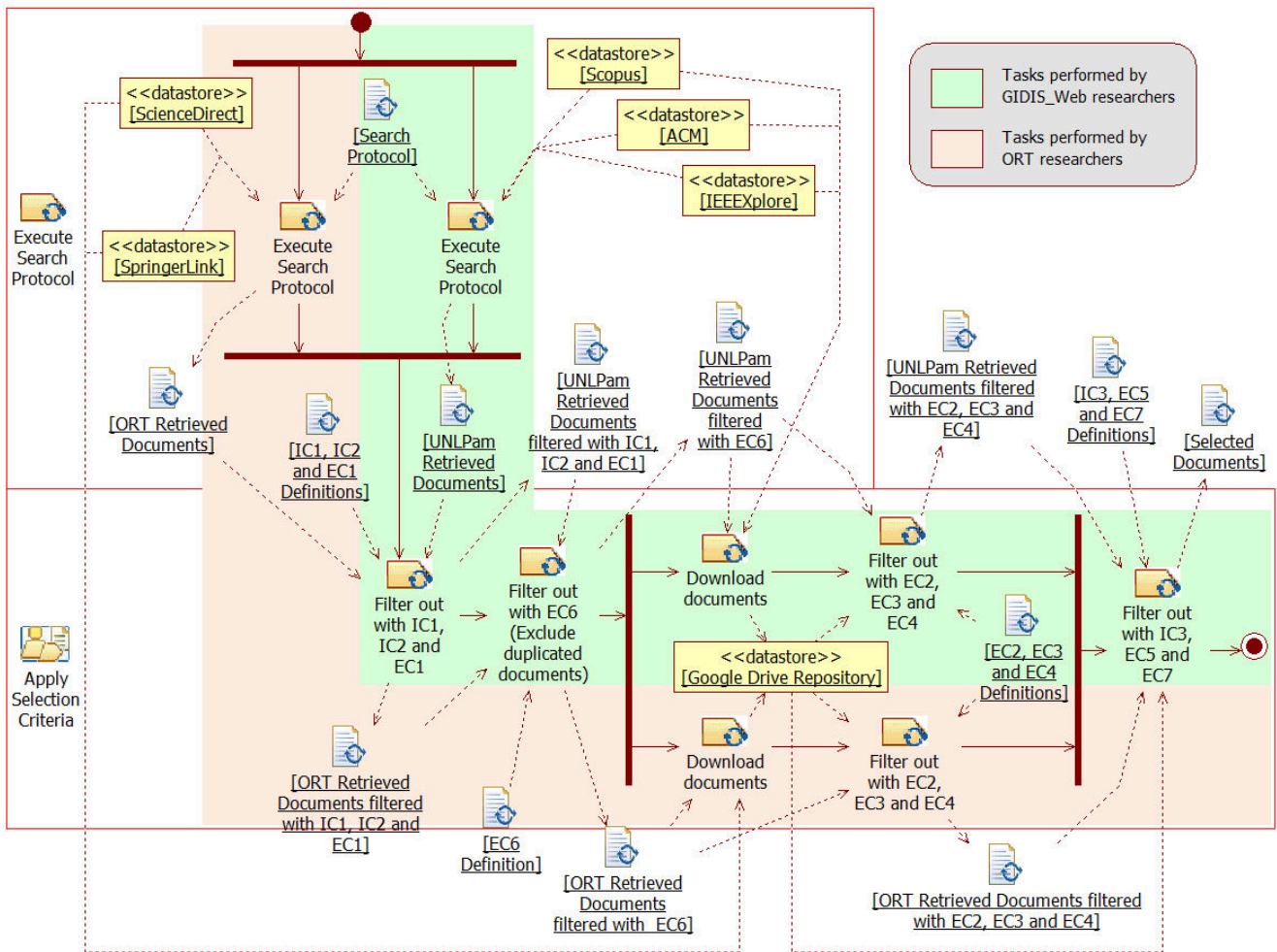


Figure 10. Instantiation of the tasks for the Execute Search Protocol and Apply Selection Criteria work definitions. Note that IC and EC stand for Inclusion Criteria, and Exclusion Criteria.

Figure 10 shows the Execute Search Protocol and Apply Selection Criteria work definitions instantiated for the SLR project on Software Testing Ontologies. The Execute Search Protocol task was performed for both research groups. Particularly, GIDIS_Web agents (green shadow in the figure) retrieved documents from the Scopus, ACM and IEEE Xplore digital libraries while, in parallel, ORT agents (orange shadow) retrieved documents from Springer Link and Science Direct digital libraries. The workload for carrying out the Execute Search Protocol tasks was balanced, i.e., two members per each group participated. The remainder two members of GIDIS_Web acted as Domain Expert and SLR Validator, who coordinated and guided to the other researchers throughout A2.2. Additionally, **Figure 10** shows the tasks instantiated in this SLR project in order to perform the Apply Selection Criteria sub-activity. Note that these tasks are scheduled taking into account the Include and Exclude Criteria artifacts of our project (see WP 1.3.1 and WP 1.3.2 artifacts in **Table 4**).

It is important to remark that from the project scheduling standpoint, for instance, the Execute Search Protocol task in the generic process in **Figure 3** was instantiated twice in **Figure 10**, considering the actual project particularities. Note also that the Apply Selection Criteria sub-activity is shown at task level in **Figure 10** considering the actual project particularities.

Therefore, the process models represented in Section 3 should be customized for any specific project life cycle and context.

As result of the Execute Search Protocol tasks, 731 documents were retrieved. **Figure 11** shows the amount of “Retrieved Documents” per digital library, following the specification of the “Search Protocol” artifact (see WP 1.2 in **Table 4**).

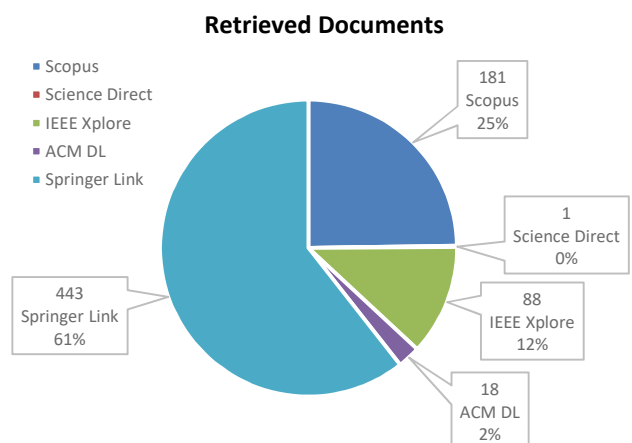


Figure 11. “Retrieved Documents” per Digital Library after performing Execute Search Protocol.

Table 5. “Selected Documents” after performing the Apply Selection Criteria sub-activity. Note that IC and EC stand for Inclusion Criteria, and Exclusion Criteria.

Criteria	Analyzed Content	Initial Studies	Eliminated	Selected Documents	Reduction
IC1, IC2, EC1	Title+Abstract	731	209	522	28.5%
EC6 (Duplicates)	Title+Abstract	522	47	475	6.4%
EC2, EC3, EC4	Full Text	475	456	19	62.3%
IC3, EC5, EC7	Full Text	19	9	10	1.2%
Total			721		98.6%

Table 6. “Final Selected Documents” after performing Add Non-Retrieved Relevant Studies.

PID	Title	Authors	Reference	Digital Library	Observations
118	An Ontology Based Approach for Test Scenario Management	Sapna P. G.; Mohanty H.	Sapna & Mohanty (2011)	Springer Link	
346	An Ontology for Guiding Performance Testing	Freitas A.; Vieira R.	Freitas & Vieira (2014)	ACM	Duplicated: Scopus, IEEE Xplore, ACM
347	Regression Tests Provenance Data in the Continuous Software Engineering Context	Campos H.; Acácio C.; Braga R.; Araújo M. A. P.; David J. M. N.; Campos F.	Campos <i>et al.</i> (2017)	ACM	
359	Ontology-Based Test Modeling and Partition Testing of Web Services	Bai X.; Lee S.; Tsai W. T.; Chen Y.	Bai <i>et al.</i> (2008)	IEEE Xplore	Duplicated: Scopus, IEEE Xplore
366	Test Case Reuse Based on Ontology	Cai L.; Tong W.; Liu Z.; Zhang J.	Cai <i>et al.</i> (2009)	IEEE Xplore	
383	An Ontology-Based Knowledge Sharing Portal for Software Testing	Vasanthapriyan S.; Tian J.; Zhao D.; Xiong S.; Xiang J.	Vasant-hapriyan <i>et al.</i> (2017b)	IEEE Xplore	Duplicated: Scopus, IEEE Xplore.
397	Ontology-based development of testing related tools	Barbosa E. F.; Nakagawa E. Y.; Riekstin A. C.; Maldonado J. C.	Barbosa <i>et al.</i> (2008)	Scopus	
424	Semi-automatic generation of a software testing lightweight ontology from a glossary based on the ONTO6 methodology	Arnicans G.; Romans D.; Straujums U.	Arnicans <i>et al.</i> (2013)	Scopus	
457	An ontology-based knowledge framework for software testing	Vasanthapriyan S.; Tian J.; Xiang J.	Vasant-hapriyan <i>et al.</i> (2017a)	Scopus	Duplicated: Scopus, SpringerLink
468	ROoST: Reference ontology on software testing	Souza É. F.; Falbo R. A.; Vijaykumar N. L.	Souza <i>et al.</i> (2017)	Scopus	
1000	Developing A Software Testing Ontology in UML for a Software Growth Environment of Web-Based Applications	Zhu H.; Huo Q.	Zhu & Huo (2005)		Retrieved from paper PID 468 using Backward Snowballing
1003	A Top Domain Ontology for Software Testing	Asman A.; Srikanth R. M.	Asman & Srikanth (2016)		Retrieved by prior knowledge

Table 5 records the number of “Selected Documents” produced after performing the Apply Selection Criteria sub-activity. This yielded 10 selected primary studies by applying the different inclusion/exclusion criteria over the analyzed content as presented in its 1st and 2nd columns. Considering the initial and end states the reached reduction rate is 98.6%.

The next activity carried out was the Add Non-Retrieved Relevant Studies sub-activity as depicted in **Figure 3**. This was performed just by GIDIS_Web members using two different methods, namely: Backward Snowballing and Prior

Knowledge of other Research Work. **Table 6** shows the “Final Selected Documents” after enacting Add Non-Retrieved Relevant Studies sub-activity. This totalized 12 research primary studies after surpassing all inclusion and exclusion criteria (see **Table 6**), namely: 10 Selected Documents plus 1 document (PID 1000) retrieved by Backward Snowballing, plus 1 document (PID 1003), a Master thesis of Asman & Srikanth (2016) retrieved by Prior Knowledge of other Research Work from the ResearchGate network by the end of 2017.

Table 7. Metrics for Terms (Tr), Attributes (At) and Relationships (Rh) for the Campos *et al.* (2017) ontology. Note that Df stands for Defined; Tx for Taxonomic; NoTx for Non-Taxonomic; and CptuaLvl for Conceptual Level.

PID	Terms (Tr)			Attributes (At)			Relationships (Rh)							
	#Tr	#DfTr	%DfTr	#At	#DfAt	%DfAt	#Rh	#TxRh	#Tx-is_a	#Tx-part_of	#NoTxRh	#DfNoTxRh	%DfNoTxRh	%TxCptuaLvl
347	14	14	100%	0	-	-	16	15	14	1	1	1	100%	93.75%

Finally, the Extract Data from all Documents sub-activity should be performed, which has as input the “Template of the Data Extraction Form” (Appendix A) and the “Final Selected Documents” (Table 6), and as output the “Forms with Extracted Data” artifact. Appendix B shows the filled form with extracted data for the PID 347 article. This sub-activity must be performed in a very disciplined and rigorous way, being also very time consuming.

The work distribution for the Extract Data from all Documents sub-activity was as follows. Two members of GIDIS_Web, as Data Collectors, gathered completely the required data of the 12 documents. At random, we selected 4 out of 12 documents which were made available (by Google Drive) for data collection by two members at Universidad ORT Uruguay, but not shared with each other while gathering the data in order to permit later a more objective checking of consistency. As result of this checking to the instantiated forms of both groups some minor issues were raised and discrepancies were consensuated via video chat. (Note that the data and quality extraction reliability could be evaluated. For example, in the study documented in Kitchenham & Brereton (2013), authors checked the level of agreement achieved for data and quality extraction. In the case of quality extraction, the Pearson correlation coefficient was found between the values for each assessor for each paper both for the number of appropriate questions and for the average quality score for each paper. In the case of data extraction, the agreement with respect to the study categories was assessed using the Kappa statistic.)

It is worth mentioning that thanks to looking for inconsistencies, we detected that into the collected concepts (i.e., terms, properties, etc.) included in the form for the RTE-Ontology (Campos *et al.* 2017) there were not only those software testing domain-specific terms and relationships but also those related to the core or high-level ontology so-called PROV-O (Lebo *et al.* 2013). Hence, we decided to document both in a differentiated way (by colors in the form) so, at analysis time, count only the domain related concepts accordingly. For instance, there are 17 terms in Campos *et al.* (2017), but just 14 are domain-specific software testing terms not taken from PROV-O. A similar counting situation happened with ROoST (Souza *et al.* 2017), but in this case, they took terms from UFO (a foundational ontology built by the same research group). Lastly, all these raised issues during the Extract Data from all Documents sub-activity were documented in the “Divergencies Resolution Record” artifact.

Ultimately, all the main SLR artifacts for this study including the filled forms with the extracted data for the 12 “Final Selected Documents” can be publicly accessed at <https://goo.gl/HxY3yL>.

4.4 To Analyze and Document the Review (A3) for the Software Testing Ontologies Study

A3 includes two sub-activities. The first sub-activity named Analyze SLR Results produces the “Data Synthesis” artifact. To produce this artifact, playing the Analysis Expert Designer role, we have designed and specified a set of direct and indirect metrics (Olsina *et al.* 2013). Below, we show just the formulas (not their whole specifications) for some indirect metrics:

$$\%DfTr = (\#DfTr / \#Tr) * 100 \tag{1}$$

$$\#Rh = \#TxRh + \#NoTxRh \tag{2}$$

$$\#TxRh = \#Tx-is_a + \#Tx-part_of \tag{3}$$

$$\%DfNoTxRh = (\#DfNoTxRh / \#NoTxRh) * 100 \tag{4}$$

$$\%TxCptuaLvl = (\#TxRh / \#Rh) * 100 \tag{5}$$

Briefly, metric’s formula (1) allows us to know the proportion of defined terms (#DfTr) with regard to the number of terms (#Tr) in the conceptualized ontology. The metric’s formula (2) is devoted to calculate the number of taxonomic (#TxRh) and non-taxonomic relationships (#NoTxRh). Note that a taxonomic relationship is the sum of the inheritance (#Tx-is_a) relationships and the whole-part (#Tx-part_of) relationships -see formula (3). The metric’s formula (4) permits to understand the proportion of defined non-taxonomic relationships. Finally, the metric’s formula (5) calculates the percentage of taxonomic conceptual-base level, which is very useful to determine if a conceptual base is actually an ontology or rather a taxonomy. (Note that, for brevity reasons, we did not include the metric’s formula for getting the ratio of specified axioms.)



Figure 12. Word cloud (<https://www.nubedepalabras.es/>) for the recorded Terms from the 12 data extraction forms. Note that Attributes and Relationships names are not included. Also, note that word size is related to the Term frequency, but Term colors have no meaning.

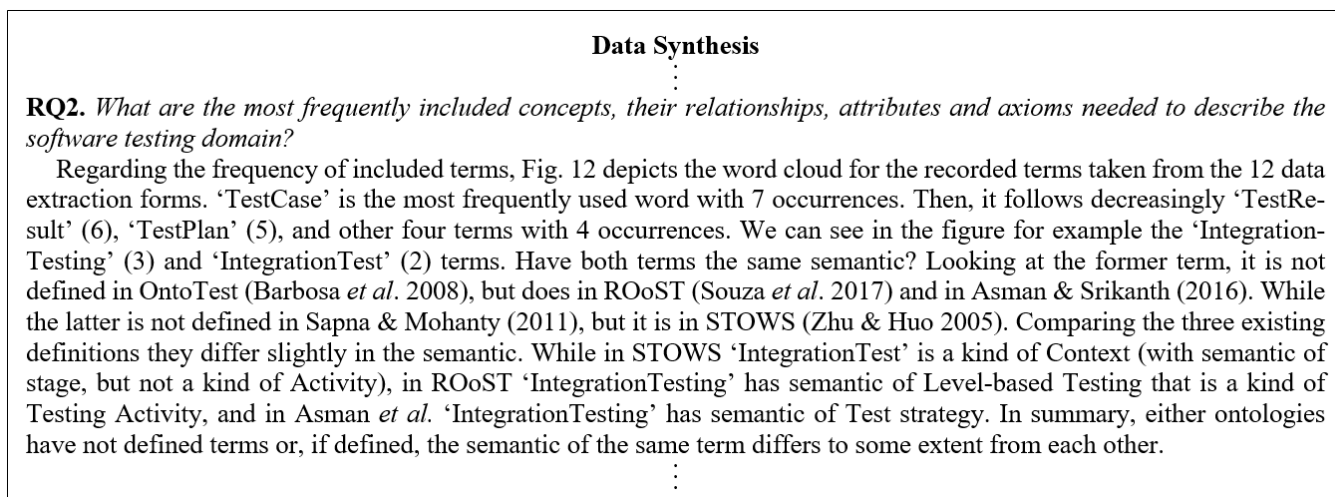


Figure 13. Fragment of the “Data Synthesis” produced in the Analyze SLR Results sub-activity.

These metrics are very easy to use considering that the “Template of the Data Extraction Form”, particularly the “Specified concepts used to describe software testing domain” field, has a suitable design for facilitating the subsequent counting (see the structure of this field in Appendix A, and one example of its instantiation in Appendix B). Additionally, we use a table to record the measure’s values. For example, **Table 7** shows the measures’ values for the paper PID 347 (Campos *et al.* 2017). Then, all these values are considered by the Data Analyzer to perform the analysis.

Moreover, in this activity we use other tools for analysis purposes. For example, a word cloud tool for the RQ2 (*What are the most frequently included concepts, their relationships, attributes and axioms needed to describe the software testing domain?*) is used. **Figure 12** shows the word cloud produced from the terms retrieved from the 12 conceptual bases.

All the tables, charts, word cloud and measures are included in the “Data Synthesis” and used by the Data Analyzer to perform the analysis. In **Figure 13**, we show an excerpt of the “Data Synthesis” produced for the software testing ontologies study. Then, using the “Data Synthesis” as input, a “Scientific Article” (Tebes *et al.* 2019b) was elaborated by the Expert Communicator and the Domain Expert in the Document/Communicate Results sub-activity. In Tebes *et al.* (2019b), the full analysis is documented, where all research questions are answered and other issues are considered, such as validity threats. However, due to the page size constraints that a conference paper has, we are currently extending it to a journal format where there is less page limits. This SLR on software testing ontologies will be then fully documented.

5 Discussion

The process proposed by Kitchenham *et al.*, which was so far adopted or slightly adapted by other researchers –for example in Sepúlveda *et al.* (2016), Tahir *et al.* (2016), Torrecilla-Salinas *et al.* (2016)- is rather coarse-grained represented using only the behavioral perspective from the process modeling standpoint. If we compare **Figure 1** with **Figure 2** –where both are representations of the behavioral perspective-, we

can observe, on one hand, a greater richness of expressiveness in sequences and decision flows in **Figure 2**, and, on the other hand, the possibility of representing different levels of granularity in work definitions such as activity, sub-activity and task. (Note that, for the reason of maintaining the simplicity and legibility of the diagram in **Figure 2**, we have not specified other aspects such as iterations and parallelisms – e.g., the parallelism that can be modeled between the Define Quality Criteria and Define Inclusion/Exclusion Criteria tasks, within the Define Selection and Quality Criteria sub-activity in **Figure 2**.)

Although the behavioral perspective is undoubtedly necessary, it is usually not sufficient since it does not represent inputs and outputs for the different activities and tasks. For this reason, the functional perspective in conjunction with the behavioral perspective enhance the model expressiveness as shown in **Figure 3**. Furthermore, the informational perspective also enriches the SLR process specification and favors the understanding of the process by showing the structure of a given artifact. This informational perspective is illustrated for the “SLR Protocol” artifact in **Figure 6**, which is produced by the A1 activity.

Additionally, the organizational perspective shown in **Figure 4** helps to understand that in order to carry out an SLR, it is also necessary to cover several roles that agents (both human and automated agents) require with different skills, i.e., the set of capabilities, competencies and responsibilities.

On the other hand, a key aspect to be highlighted in the present proposal is the modeling of the A2.1 sub-activity (Perform SLR Pilot Study), which gives feedback to the A1 activity (Design Review). This pilot test activity, in the adapted processes of Brereton *et al.* (2007) is very often neglected. Alternatively, if it is mentioned or considered such as in Brereton *et al.* (2007) and Sepúlveda *et al.* (2016), it is poorly specified. For example, in Sepúlveda *et al.* (2016), authors include the Selection and Pilot Extraction activity, which is represented simply as a step in the A2 activity – named phase in Sepúlveda *et al.* (2016)-, but it does not give feedback to the A1 activity (phase). So modeling the feedback loop is important due to it could help to improve aspects

of the SLR design, as we have proposed in **Figure 7**, and illustrated its usefulness in sub-section 4.3.1, particularly in the improvement of the Data Extraction Form. Additionally, notice that in our proposal we include both the Validate SLR Design sub-activity (which is represented in A1) and the Perform SLR Pilot Study sub-activity (which clearly must be represented in A2 since it contains tasks inherent to the execution of the SLR).

In short, the main contribution of this work is to augment the SLR process currently and widely used by the SE scientific communities. This is achieved by considering, on one hand, the principles and benefits of process modeling with greater rigor by using four modeling perspectives, namely: functional, behavioral, informational and organizational. And, on the other hand, the vision of decoupling the ‘what’ to do aspect, which is modeled by processes, activities, tasks, artifacts and behavioral aspects, from the ‘how’ to realize the description of work definitions, which is modeled by method specifications. It can be observed in the cited literature (and in general) a lack of separation of concerns between what to do (processes) and how to do it (methods).

Although throughout Sections 3 and 4 we have indicated some methods applicable to SLR tasks, the emphasis in this work is not on the specifications of methods. This separation of concerns can be seen, for example, in the description of the Add Non-Retrieved Relevant Studies task (recall sub-section 4.3.2), where it can be carried out by using two methods such as the forward snowballing and/or backward snowballing. However, in the process models related to this sub-activity (and to others) no reference is made to these methods, nor to any others.

6 Conclusion

In this work, we have documented the SLR process specification by using process-modeling perspectives and mainly the SPEM language. It is a recommended flow for the SLR process, since we are aware that in a process instantiation there might be some variation points, such as the parallelization of some tasks. It is worth noting that, regarding the quoted benefits of process modeling in the Introduction Section, the proposed SLR process specifications aim primarily at facilitating the understanding and communication, as well as at giving support to the process improvement and process management. However, a thorough discussion and a detailed illustration of process modeling for supporting fully SLR process automation have been out of the scope of this article.

Additionally, we have highlighted the benefits and strengths of the proposed SLR process model compared with others, which are more coarse-grained specified. Finally, we have illustrated aspects of it by exemplifying a SLR on software testing ontologies. One important contribution is the inclusion of the pilot test activity, which promotes the validation of the “SLR Protocol” artifact not only in the A1 activity but also in the A2.1 sub-activity. It is worthwhile to highlight that conducting a pilot study (not for a replicated study) can be very useful to improve the SLR Protocol and foster to some extent the quality of the evidence-based outcomes. Given that so far we have performed very few pilot studies,

we have not the practical evidence to state what is the more effective sample size for it. In fact, Kitchenham & Charters (2007) neither mention the appropriate sample size due to – they indicate- it depends on the available resources as well. In our humble opinion, we consider that in a pilot study, in addition to the sample size, it is important to select a set of documents (or the whole sample) and perform the data extraction of each document by more than one independent data collector. Having more than one filled data extraction form for the same document allows checkings for potential inconsistencies in the designed artifacts. Moreover, the Data Validator (SLR Expert Researcher) and the Domain Expert are very important roles that should be played at least by one person with expertise in order to check (jointly with the independent data collectors) the data extraction forms for inconsistencies, in addition to detect opportunities for improvement in the template metadata for the ulterior analysis endeavor, in the A3 activity. In a nutshell, we consider that several variables may be important for making a good/effective pilot test. However, we would need more informed evidence to tackle this issue. So it is an interesting aspect that could be further investigated by the community.

The proposed process model for the SLR provides a good baseline for understanding the details and discussing alternatives or customizations to this process. In fact, the rigor provided by process modeling, where several perspectives are combined (e.g. functional, informational, organizational and behavioral), but can also be independently detached, provides a greater richness of expressiveness in sequences and decision flows, while representing different levels of granularity in the work definitions, such as activity, sub-activity and task.

It is worth mentioning that the specified process contributes to one pillar of a well-established SLR strategy –knowing beforehand that a strategy should also integrate the method specification capability. Note that, for the same task, different method specifications could be applied. In consequence, the life cycle of a given SLR project should organize activities and tasks considering not only the prescribed SLR process but also the appropriate allocation of resources such as methods and tools, among others, for achieving the proposed goal. There are additional activities (to those specified in **Figure 2**) in which project planning must also take into account such as documenting artifacts in all activities and control their changes and versions. The continuous documentation and versioning of artifacts are key factors to guarantee consistency, repeatability and auditability of SLR projects.

As an ongoing work, we are currently developing a supporting tool for this SLR process since it can be very time consuming and also error prone remembering and using all the elements that this process provides. Although a variety of tools is available to assist the SLR process (Marshall & Brereton 2013), current tools obviously do not follow our SLR process or do not fit well. Consequently, we are developing a tool that can help to automate part or all of our SLR process and its documentation, in addition to favor its usefulness and wide adoption. Taking into account that the main objective in the present work is to provide models to facilitate the understanding and the communication of the SLR process to researchers and practitioners, rather than to give support

to the fully SLR process automation, we will need to augment for instance some activity specifications at task level in addition to provide a more flexible process flow for collaborative work.

On the other side, as ongoing work, we are currently finishing the development of the suitable top-domain testing ontology to be integrated into the C-INCAMI v.2 conceptual framework. To this end, we took into account those explicit terminologies coming not only from some of the existing primary studies, but also from official and *de facto* international standards (e.g. ISO/IEC/IEEE 29119-1:2013 <https://www.iso.org/standard/45142.html>, and ISTQB <https://www.istqb.org/downloads.html> respectively), which are widely adopted by professional testers.

Lastly, as a future work, we will perform a thorough checking if our SLR process specifications can suitably represent the SLR processes followed by other researchers. The outcome of this work will help us to validate our process specifications in a broader way.

Acknowledgments.

This work and line of research are supported by Science and Technology Agency of Argentina in the PICT 2014-1224 project, at Universidad Nacional de La Pampa.

References

- Arnicans, G., Romans, D., & Straujums, U. (2013). Semi-automatic Generation of a Software Testing Lightweight Ontology from a Glossary Based on the ONTO6 Methodology, *Frontiers in Artificial Intelligence and Applications*, V.249, pp. 263-276.
- Asman, A., & Srikanth, R. M. (2016). A Top Domain Ontology for Software Testing, Master Thesis, Jönköping University, Sweden, pp. 1-74.
- Bai, X., Lee, S., Tsai, W. T., & Chen, Y. (2008). Ontology-Based Test Modeling and Partition Testing of Web Services, *IEEE Int'l Conference on Web Services (ICWS'08)*, pp. 465-472.
- Barbosa, E. F., Nakagawa, E. Y., Riekstin, A. C., & Maldonado, J. C. (2008). Ontology-based Development of Testing Related Tools, *20th International Conference on Software Engineering and Knowledge Engineering (SEKE'08)*, pp. 697-702.
- Becker, P., Lew, P., & Olsina, L. (2012). Specifying Process Views for a Measurement, Evaluation, and Improvement Strategy, *Advances in Software Engineering Journal*, Academic Editor: Osamu Mizuno, Hindawi Publishing Corporation, USA, V.2012, 27 pgs.
- Becker, P., Papa, F., & Olsina, L. (2015). Process Ontology Specification for Enhancing the Process Compliance of a Measurement and Evaluation Strategy, *CLEI eJnal*, 18:(1), pp. 1-26.
- Biolchini, J., Mian, P.G., Natali A.C.C., & Travassos, G. (2005). Systematic Review in Software Engineering. Technical Report RT-ES 679-05. PESC, COPPE/UFRJ.
- Brereton, P., Kitchenham, B., Budgen, D., Turner, M., & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain, *Journal of Systems and Software*, 80:(4), pp. 571-583.
- Cai, L., Tong, W., Liu, Z., & Zhang, J. (2009). Test Case Reuse Based on Ontology, *15th IEEE Pacific Rim International Symposium on Dependable Computing*, pp. 103-108.
- Campos, H., Acácio, C., Braga, R., Araújo, M. A. P., David, J. M. N., & Campos, F. (2017). Regression Tests Provenance Data in the Continuous Software Engineering Context, *2nd Brazilian Symposium on Systematic and Automated Software Testing (SAST)*, Paper 10, pp. 1-6.
- Curtis, B., Kellner, M., & Over, J. (1992). Process Modelling, *Communications of ACM*, 35:(9), pp. 75-90.
- Freitas, A., & Vieira, R. (2014). An Ontology for Guiding Performance Testing, *IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), (WI-IAT '14)*, V.1, pp. 400-407.
- Garousi, V., & Mäntylä, M. (2016). A systematic literature review of literature reviews in software testing, *Information and Software Technology*, V.80, pp. 195-216.
- Irshad, M., Petersen, K., & Poulding, S. (2018). A systematic literature review of software requirements reuse approaches, *Information and Software Technology*, V.93, pp. 223-245.
- Kitchenham, B. (2004). Procedures for Undertaking Systematic Reviews, Joint TR, Comp. Science Dep., Keele University (TR/SE-0401) and National ICT Australia Ltd. (0400011T.1).
- Kitchenham, B., & Brereton, P. (2013). A systematic review of systematic review process research in software engineering, *Information and Software Technology*, 55:(12), pp. 2049-2075.
- Kitchenham, B., & Charters, S. (2007). Guidelines for Performing Systematic Literature Reviews in Software Engineering, EBSE Technical Report, Software Engineering Group, School of Computer Science and Mathematics Keele University and Department of Computer Science University of Durham, UK, V. 2.3.
- Kitchenham, B., Budgen, D., & Brereton, P. (2010a). The value of mapping studies—a participant-observer case study, *4th International Conference on Evaluation and Assessment in Software Engineering*, British Computer Society, pp. 25-33.
- Kitchenham, B., Pretorius, R., Budgen, D., Brereton, P., Turner, M., Niazi, M., & Linkman, S. (2010b). Systematic literature reviews in software engineering —A tertiary study, *Information and Software Technology*, 52:(8), pp. 792-805.
- Kitchenham, B.A., Dyba, T., & Jorgensen, M. (2004). Evidence-based software engineering. *26th International Conference on Software Engineering*, pp. 273-281.
- Lam, R. W., & Kennedy, S. H. (2005). Using metaanalysis to evaluate evidence: Practical tips and traps. *Canadian Journal of Psychiatry*, 50, pp. 167-174.
- Lebo, T., Sahoo, S., McGuinness, D., Belhajjame, K., Cheney, J., Corsar, D., Garijo, D., Soiland-Reyes, S.,

- Zednik, S., & Zhao, J. (2013). PROV-O: The PROV Ontology. Retrieved by July 1st, 2019, from <https://www.w3.org/TR/2013/REC-prov-o-20130430/>
- Marshall, C., & Brereton, P. (2013). Tools to Support Systematic Literature Reviews in Software Engineering: A Mapping Study. ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, Baltimore, MD, pp. 296-299. DOI: 10.1109/ESEM.2013.32
- Meline, T. (2006). Selecting studies for systematic review: Inclusion and exclusion criteria. *Contemporary issues in communication science and disorders*, 33, pp. 21-27.
- Napoleão, B.M., Felizardo, K.R., de Souza, E.F., & Vijaykumar, N. (2017). Practical similarities and differences between Systematic Literature Reviews and Systematic Mappings: a tertiary study, The 29th International Conference on Software Engineering and Knowledge Engineering, pp. 85-90, DOI: 10.18293/SEKE2017-069.
- Olsina, L. (1998). Functional View of the Hypermedia Process Model, 5th International Workshop on Engineering Hypertext Functionality, at ICSE'98, Kyoto, Japan, pp. 1-10.
- Olsina, L., & Becker, P. (2017). Family of Strategies for different Evaluation Purposes, XX CIBSE'17, CABA, Argentina, Published by Curran Associates, pp. 221-234.
- Olsina, L., & Becker, P. (2018). Linking Business and Information Need Goals with Functional and Non-functional Requirements, XXI CIBSE'18, Bogotá, Colombia, Published by Curran Associates, pp. 381-394.
- Olsina, L., Covella, G., & Dieser, A. (2013). Metrics and Indicators as Key Organizational Assets for ICT Security Assessment, Chapter 2, in the book titled "Emerging Trends in ICT Security", Elsevier (Morgan Kaufmann), 1st Edition, Akhgar & Arabnia (Eds.), pp. 25-44. ISBN: 9780124114746
- OMG (2008). Software & Systems Process Engineering Meta-Model (SPEM) Specification, Version 2.0.
- OMG (2011). Business Process Model and Notation (BPMN) Specification, Version 2.0.
- OMG (2017). Unified Modeling Language (UML) Specification, Version 2.5.1.
- Petersen, K., Vakkalanka, S., & Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update, *Information and Software Technology*, V.64, pp. 1-18.
- Portela, C., Vasconcelos, A., Silva, A., Sinimbu, A., Silva, E., Ronny, M., Lira, W., & Oliveira, S. (2012). A Comparative Analysis between BPMN and SPEM Modeling Standards in the Software Processes Context, *Journal of Software Engineering and Applications*, 5(5), pp. 330-339.
- Rosenberg, D.S.W., Gray, J., Hayes, R., & Richardson, W. (1996). Evidence-based medicine: what it is and what it isn't, *British Medical Journal*, Vol. 312, No. 7023, p. 71.
- Russel, N., van der Aalst, W., Hofstede, A., & Wohed, P. (2006). On the suitability of uml activity diagrams for business process modelling, Third Asia-Pacific Conference on Conceptual Modelling (APCCM), Hobart, V.53, pp. 195-204.
- Sapna, P. G., & Mohanty, H. (2011). An Ontology Based Approach for Test Scenario Management, 5th International Conference on Information Intelligence, Systems, Technology and Management (ICISTM'2011), V.141, pp. 91-100.
- Sepúlveda, S., Cravero, A., & Cachero, C. (2016). Requirements modeling languages for software product lines: A systematic literature review, *Information and Software Technology*, V.69, pp. 16-36.
- Souza, E. F., Falbo, R. A., & Vijaykumar, N. L. (2017). ROoST: Reference Ontology on Software Testing, *Applied Ontology Journal*, 12:(1), pp. 1-30.
- Tahir, T., Rasool, G., & Gencel, C. (2016). A systematic literature review on software measurement programs, *Information and Software Technology*, V.73, pp. 101-121.
- Tebes, G., Peppino, D., Dameno, J., Becker, P., & Olsina L. (2018). Diseñando una Revisión Sistemática de Literatura sobre Ontologías de Testing de Software. (In English: Designing a Systematic Literature Review on Software Testing Ontologies), VI Congreso Nacional de Ingeniería Informática/Sistemas de Información (CoNaIISI), Mar del Plata, Argentina, pp. 1-13.
- Tebes, G., Peppino, D., Becker, P., & Olsina, L. (2019a). Especificación del Modelo de Proceso para una Revisión Sistemática de Literatura. (In English: Specifying the Process Model for a Systematic Literature Review), XXII CIBSE'19, La Habana, Cuba, Published by Curran Associates 2019, pp. 391-404, ISBN 978-1-5108-8795-4.
- Tebes, G., Peppino, D., Becker, P., Matturro, G., Solari, M., & Olsina, L. (2019b). A Systematic Review on Software Testing Ontologies, 12th International Conference on the Quality of Information and Communications Technology, Springer book, CCIS V.1010, M. Piattini et al. (Eds.): QUATIC 2019, Ciudad Real, Spain, pp. 144-160, https://doi.org/10.1007/978-3-030-29238-6_11.
- Torrecilla-Salinas, C.J., Sedeño, J., Escalona, M.J., & Mejías, M. (2016). Agile, Web Engineering and Capability Maturity Model Integration: A systematic literature review, *Information and Software Technology*, V.71, pp. 92-107.
- Vasanthapriyan, S., Tian, J., & Xiang J. (2017a). An Ontology-Based Knowledge Framework for Software Testing, *Communications in Computer and Information Science*, V.780, pp. 212-226.
- Vasanthapriyan, S., Tian, J., Zhao, D., Xiong, S., & Xiang, J. (2017b). An Ontology-Based Knowledge Sharing Portal for Software Testing, IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C'17), pp. 472-479.
- White, S. A. (2004). Process modeling notations and workflow patterns, *Workflow Handbook 2004*, pp. 265-294.
- Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering, *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE '14)*, ACM, New York, NY, USA, Article 38, pp. 1-38.
- Zhu, H., & Huo, Q. (2005). Developing A Software Testing Ontology in UML for a Software Growth Environment of Web-Based Applications, *Software Evolution with UML and XML*, IDEA Group, pp. 263-295.

APPENDIX A

Data Extraction Form	
Researcher name	Last Name (Surname), First Name
PID - Article title	
Author/s of the article	Author1_Surname, Initials_of_the_Name; Author2_Surname, Initials_of_the_Name; ...
Journal/Congress/other	Name
Publication year	
Digital library	Name (Note: <i>If an article was repeated, indicate the name of all the libraries in which was found</i>)
Name of the proposed ontology	Name and/or Acronym
Specified concepts used to describe software testing domain	<u>Terms:</u> TermN: definition <u>Attributes:</u> TermN (AttributeN.1 = definition, AttributeN.2 = definition,...) <u>Relationships:</u> is_a (TermX_type, TermY_subtype) part_of (TermX_whole, TermY_part) RelationM_name (TermX, TermY): definition RelationZ_without_name (TermX, TermY): definition <u>Axioms or restrictions:</u> Axiom: definition/specification
Methodology used to develop the ontology	Name and/or Acronym
Terminologies or Vocabularies taken into account to develop the proposed ontology	<u>Ontologies:</u> name/s and/or reference/s <u>Taxonomies:</u> name/s and/or reference/s <u>Glossaries/Dictionaries:</u> name/s and/or reference/s
Classification of the proposed ontology	Select one category: 1- Foundational Ontology (top level) 2- Top-Domain Ontology 3- Domain Ontology 4- Instance/Application Ontology 5- Not determined by authors
Research context	Description
Research objective/s related to software testing ontologies	Description
Does the proposed ontology consider its linking with Functional and Non-Functional Requirements concepts?	1- Yes 0- No
Additional notes	<i>(This field is used for the researcher to make his/her observations or comments about something that is important to highlight)</i>

APPENDIX B

Data Extraction Form	
Researcher name	Peppino, Denis H.
PID - Article title	347 - Regression Tests Provenance Data in the Continuous Software Engineering Context
Author/s of the article	S. Campos Junior, H.; Paiva, C. A.; Braga, R.; Araújo, M. A. P.; David, J. M. N.; Campos, F.
Journal/Congress/other	Proceedings of the 2nd Brazilian Symposium on Systematic and Automated Software Testing (SAST), pp. 10:1 - 10:6
Publication year	2017
Digital library	ACM
Name of the proposed ontology	RTE-Ontology
Specified concepts used to describe software testing domain	<p><u>Important</u>: See additional note #2.</p> <p><u>Terms</u>:</p> <ol style="list-style-type: none"> 1. <u>Activity</u> 2. TestingActivity: Groups test execution activities. 3. TestSuiteExecution: Represents one test session. 4. TestCaseExecution: Represents one test case. 5. <u>Agent</u> 6. Person: Represents a person in the environment. 7. <u>Entity</u> 8. TestingEnvironment: Groups classes of the testing environment. 9. SoftwareBuild: Represents a specific build of the software. 10. Artifact: Groups software artifacts. 11. TestingArtifact: Groups artifacts related to testing. 12. TestingLog: Logs generated by a test session. 13. TestingSourceCode: Represents the source class of test cases. 14. SoftwareArtifact: Represents artifacts not directly related to test. 15. SourceCode: Represents software's source code. 16. SoftwareExecutable: Binaries used to execute the software. 17. TestingSuite: Groups test classes. <p><u>Attributes</u>:</p> <p>-</p> <p><u>Relationships</u>:</p> <p>is_a(Activity, TestingActivity) is_a(TestingActivity, TestSuiteExecution) is_a(TestingActivity, TestCaseExecution) is_a(Agent, Person) is_a(Entity, TestingEnvironment) is_a(TestingEnvironment, SoftwareBuild) is_a(Entity, Artifact) is_a(Artifact, TestingArtifact) is_a(TestingArtifact, TestingLog) is_a(TestingArtifact, TestingSourceCode) is_a(Artifact, SoftwareArtifact) is_a(SoftwareArtifact, SourceCode) is_a(SoftwareArtifact, SoftwareExecutable) is_a(Entity, TestingSuite) composedOf(TestingSuite, TestingSourceCode) covers(TestingSuite, SourceCode): represent which source code is covered by a test suite.</p> <p><u>wasInformedBy(Activity, Activity)</u> <u>wasAssociatedWith(Activity, Agent)</u> <u>actedOnBehalfOf(Agent, Agent)</u> <u>used(Activity, Entity)</u> <u>wasDerivedFrom(Entity, Entity)</u> <u>wasGeneratedBy(Entity, Activity)</u> <u>wasAttributedTo(Entity, Agent)</u></p> <p><u>Axioms or restrictions</u>:</p> <p>-</p>
Methodology used to develop the ontology	-
Terminologies or Vocabularies taken into account to develop the proposed ontology	<p><u>Ontologies</u>: PROV-O (Lebo et al. 2013)</p> <p><u>Taxonomies</u>: -</p> <p><u>Glossaries/Dictionaries</u>: -</p>

Classification of the proposed ontology	3- Domain Ontology
Research context	In this paper, they propose an architecture based on the use of an ontology and provenance model to capture and provide regression tests data to support the continuous improvement of software testing processes.
Research objective/s related to software testing ontologies	The main objective of this paper is to propose an approach capable of capturing and providing information about past executions of regression tests. To this, data provenance is used to achieve this goal.
Does the proposed ontology consider its linking with Functional and Non-Functional Requirements concepts?	0- No
Additional notes	Note 1: RTE: Regression Test Execution. Note 2: The terms/relationships highlighted in red and underlined were not counted and not taken into account in the analysis activity because they belong to the PROV-O, which has core or high-level concepts.