Building Empirical Software Engineering Bodies of Knowledge with Systematic Knowledge Engineering

Stefan Biffl\textsuperscript{a}, Marcos Kalinowski\textsuperscript{b}, Fajar Ekaputra\textsuperscript{a}, Estefanía Serral\textsuperscript{a}, Dietmar Winkler\textsuperscript{a}

\textsuperscript{a} CDL-Flex, Inst. Software Technology and Interact.Sys.
Vienna University of Technology
Vienna, Austria
<firstname>.<lastname>@tuwien.ac.at

\textsuperscript{b} NEnC, Núcleo de Engenharia do Conhecimento
Federal University of Juiz de Fora
Juiz de Fora, Brazil
kalinowski@ice.ufjf.br

Abstract—[Context] Empirical software engineering (EMSE) researchers conduct systematic literature reviews (SLRs) to build bodies of knowledge (BoKs). Unfortunately, valuable knowledge collected in the SLR process is publicly available only to a limited extent, which considerably slows down building BoKs incrementally. [Objective] In this paper, we introduce the Systematic Knowledge Engineering (SKE) process to support building up BoKs from empirical studies efficiently. [Method] SKE is based on the SLR process and on Knowledge Engineering (KE) practices to provide a Knowledge Base (KB) with semantic technologies that enable reusing intermediate data extraction results and querying of empirical evidence. We evaluated SKE by building a software inspection EMSE BoK KB from knowledge acquired by controlled experiments. We elicited relevant queries from EMSE researchers and systematically integrated information from 30 representative research papers into the KB. [Results] The resulting KB was effective in answering the queries, enabling knowledge reuse for analyses beyond the results from the SLR process. [Conclusion] SKE showed promising results in the software inspection context and should be evaluated in other contexts for building EMSE BoKs faster.

Keywords—Empirical software engineering, systematic knowledge engineering, systematic review, software inspection.

I. INTRODUCTION

Researchers in Empirical Software Engineering (EMSE) collaborate on research topics, such as defect detection methods for software inspection [1], to build up a body of knowledge (BoK). An EMSE BoK includes theory models [2], hypotheses derived from the theory models, and results from empirical studies that test those hypotheses [3], to explain and/or predict EMSE phenomena.

A consequence of the growing number of EMSE studies is the need to adopt systematic approaches for aggregating research outcomes in order to provide an objective summary of evidence on a particular topic [4]. In this context, systematic literature reviews (SLRs) have become a widely used research method [5]. However, the main public result of a SLR is, in general, a specific research synthesis report [6], while the accumulated knowledge in the SLR working material, generated from heterogeneous empirical study data sources [7], is not available to other researchers. Therefore, each new SLR project has to overcome knowledge sharing issues and rebuild large parts of existing knowledge from previous SLRs, making building a BoK considerably less efficient than necessary. Moreover, meta-analyses are limited to the presented research synthesis, not allowing other researchers to explore the underlying extracted information in different ways in order to answer questions related to their specific research goals.

Fig. 1 illustrates EMSE BoK building research challenges with key stakeholders, artifacts, and technologies. EMSE BoK researchers produce SLR and empirical study reports, available to general readership through digital libraries. However: (1) data extracted during the SLR process usually stays in a local archive and (2) the SLR report contains a specific research synthesis and there is seldom a way for other researchers to access the extracted data to apply different analyses and research synthesis methods, or to extend the data in the BoK.

![Figure 1. EMSE BoK research challenges.](image-url)

In this paper, we address these challenges by introducing the Systematic Knowledge Engineering (SKE) process, which enables systematically building up EMSE BoKs from empirical studies. SKE builds on the SLR process [8], improving data management by storing knowledge on BoK domain concepts typically used for theory building [2] from EMSE studies [3] in a Knowledge Base (KB). The semantic technology used in SKE facilitates researchers to identify relevant knowledge in the BoK through semantic queries (e.g., on synonyms and related concepts). Thus, SKE allows truly building up knowledge incrementally as researchers can access and reuse knowledge from past studies and integrate new contributions. Also, the KB enables researchers to explore knowledge aggregated from EMSE study reports through an extensible set of queries.

For evaluation we instantiated SKE to build a “Software Inspection” EMSE BoK KB based on a specific type of empirical study: controlled experiments. First, we elicited a set of relevant
query candidates in a survey with EMSE BoK researchers. Then, we modeled BoK topics related to software inspection theory and a common data model for controlled experiment data on those topics, defining the relevant information to extract. Following SKE’s search process, 102 software inspection experiments were identified. An independent team extracted information from the 30 most recent papers for data import and query resolution by a knowledge engineer.

Major findings were: (1) SKE was effective in identifying and characterizing inspection experiments and their results, (2) it was possible to build the software inspection KB from published experiments, and (3) the KB was effective in providing answers to the required queries. Additionally, we compared the SKE and SLR processes, when applied to building EMSE BoKs. In this context, the SKE KB facilitates reusing and exploring the extracted data based on semantic search capabilities not available for SLR reports. While the data extraction effort is comparable, SKE requires a knowledge engineer for data modeling, mapping, and providing query facilities. This overhead is offset by benefits of the KB, which can be used among researchers in and beyond a work group to incrementally build EMSE BoKs.

The remainder of this paper is organized as follows. Section 2 summarizes related work. Section 3 motivates the research issues. Section 4 introduces SKE. Section 5 presents the evaluation. Section 6 discusses results and lessons learned. Section 7 concludes the paper.

II. RELATED WORK

The quality and speed of building up a body of knowledge (BoK) in an EMSE research area depend on the ability of researchers to discover the existing content in a BoK, e.g., empirical studies investigating a set of hypotheses or variables. Currently, online searching for content is supported by syntactic full-text search on specialized databases, such as digital libraries. However, support for semantic searching is limited and researchers may not discover all relevant content.

Although some effort has been spent on repositories for empirical studies (e.g., CeBASE [9] and ViSEK [10]), they did not show significant progress since their introduction. To our knowledge, there is no related work on using SLR-based study identification and integration of bottom-up EMSE BoK building to facilitate reuse and semantic search. Therefore, this section describes work related to the theoretical foundations of this research’s main constructs: Systematic Literature Reviews (Section II.A) and Knowledge Base Design & Population (Section II.B). Further, we describe the foundations on Software Inspections (Section II.C) as basis for the evaluation use case.

A. Systematic Literature Reviews

Kitchenham and Charters [8] developed guidelines for performing Systematic Literature Reviews (SLRs) in the software engineering (SE) domain. Those guidelines state that the main reasons for conducting SLRs are (a) summarizing existing evidence concerning a treatment or technology; (b) identifying gaps in current research; and (c) providing background to appropriately position new research activities. The first reason is directly related to building BoKs by gathering evidence-based knowledge. In the context of this research, the main advantage of using SLRs is allowing systematically summarizing knowledge on a specific BoK scope and enabling incremental updates on top of previous SLRs. An example of such updates is available in [11], where four independent SLR trials were conducted to incrementally build evidence-based defect causal analysis guidelines. Lessons learned from applying SLRs to the SE domain are reported by Brerton et al. [4].

The SLR guidelines [8] summarize three main phases of a systematic review: Planning the Review, Conducting the Review, and Reporting the Review. The stages related to each of those phases are: (a) Planning the Review: identification of the need for a review, commissioning a review, specifying the research questions, and developing a review protocol; (b) Conducting the Review: identification of research, selection of primary studies, study quality assessment, data extraction and monitoring, and data synthesis; and (c) Reporting the Review: specifying dissemination mechanisms, formatting the main report, and evaluating the report. The PICO (population, intervention, comparison, outcome) strategy [12] is suggested [8] for detailing the research question elements in order to support developing the review protocol.

SLRs have become a widely used and reliable research method [5]. However, the main public result of a SLR is, in general, a specific research synthesis report by the authors. Unfortunately, reusable SLR packages that include the working material, which holds the accumulated knowledge, are available only seldom. The working material includes the data extracted from the primary studies (commonly stored in spreadsheets). Therefore, new SLRs have to rebuild large parts of existing knowledge, making the addition of knowledge less efficient than necessary. Even if such knowledge were available, data integration mechanisms to enable making the knowledge available for further use by other EMSE BoK researchers have not been discussed in this context. In summary, the reuse value of SLR knowledge to help building EMSE BoKs is limited.

B. Knowledge Base Design and Population

The process of building a KB may be seen as a modeling activity [13]. For creating a KB, it is essential to capture domain knowledge through content-specific agreements, so both human and knowledge-based systems can access and use the information [13]. For this purpose, formal ontologies have been successfully used since the 1990s [14]. Ontologies can provide standard terminologies and rich semantics to facilitate knowledge sharing and reuse [13]. OWL DL (Web Ontology Language - Description Logic) is the most used language for ontologies as it has the capability of supporting semantic interoperability to exchange and share context knowledge between different systems, and keeps a balance between expressiveness and automated processing. In addition, ontology enhances searching mechanisms, which may refer to precise semantic concepts rather than simple syntactic keywords, facilitating the use of the knowledge stored in the ontology [15].

Many methodologies have been proposed to design ontologies [16]. However, only a few of them consider collaborative and distributed construction of ontologies, such as the Collaborative Design Approach [17], which addresses the issue of collaborative construction of the ontologies by identifying and involving a diverse panel of participants.
Once the ontology or the data model of the KB is defined, it is necessary to capture the extracted data from information resources in accordance to the KB. This process is called KB population, and involves the creation, transformation and integration of individuals (instances) into the KB. In our case, the information resources for creating the KB are empirical study reports. The KB population process may face integration problems if the information resources use heterogeneous representations of the same concepts. The Interchange Standard Approach has been recommended as one of the best solution options for semantic integration [18]. Currently available tools to manage ontologies usually require ontology experts. Therefore, ontology non-experts need effective and efficient interfaces for both, importing and exporting knowledge, and for querying.

C. Software Inspections

Software inspections (SI) improve software quality by the analysis of software artifacts, detecting defects for removal before these artifacts are delivered to following software life cycle activities [1]. The traditional SI process by Fagan [19] involves a moderator planning the inspection, inspectors reviewing the artifact, a team meeting to discuss and register defects, passing the defects to the author for rework, so they can be corrected, and a final follow-up evaluation by the moderator on the need of a new inspection.

Over the years, many contributions on SI have been proposed, including alternative SI processes, SI methods to improve the effectiveness and efficiency of inspectors in detecting defects, models and guidelines supporting tasks of the inspection process that involve decision making, and tool support [20]. Much knowledge on those contributions has been acquired by conducting empirical studies and can be considered part of a BoK in the SI area. However, such knowledge is currently not organized in the context of a BoK. Therefore, it still takes considerable expertise and effort to identify studies and study results relevant for a given topic in the scope of SI. To support the SI community in building up their BoK, specific BoK topics can help to define the scope of knowledge. The IEEE Software Engineering BoK (SWEBoK) [21] breaks down the Testing BoK into the following topics: Fundamentals, Test Levels, Test Techniques, Test-Related Measures, Test Process, and Software Testing Tools. SI relates to Test Techniques in the IEEE SWEBoK. Nevertheless, they can be seen as a similar topic of interest and a hierarchical structure for them could be useful to facilitate organizing knowledge.

A fixed BoK topic structure may be limiting, since it is possible to apply several variant options of SI. Laitenberger and DeBaud [22] provide some parameters that help define SI variant options based on an early literature review on SI experiments: SI artifacts (e.g., requirements, design or code), SI process (e.g., SI with or without a group meeting), and SI methods (e.g., reading techniques). Such a list of parameters could be used as a starting point for a SI EMSE BoK topic structure.

III. RESEARCH ISSUES

The overall SKE goal is to enable incrementally building up an EMSE BoK by providing a process for knowledge acquisition and querying. Therefrom we derive three research issues.

Research Issue RI-1: SKE Requirements Analysis. Which queries to knowledge on empirical studies are most relevant to EMSE BoK stakeholders?

EMSE researchers want to synthesize EMSE BoKs but tend to focus on conducting empirical studies and seem to spend much less effort on considering data management to provide their EMSE BoK community with suitable access to the created knowledge [7]. Building on typical hypothesis patterns reported by Sjøberg et al. [2] we conducted an informal survey with EMSE BoK researchers from six active work groups on queries to knowledge on empirical studies that they need for their work. We are aware of the limitations of this survey and see the survey outcome as a preliminary working result, which is still useful to drive the SKE research at this stage, and can be extended by a future more formal survey in a wider scope.

Research Issue RI-2: SKE Process and Data Modeling. How can the traditional SLR process be adapted to support incrementally building EMSE BoKs? What data elements are necessary to address the most relevant queries of EMSE BoK researchers?

The SKE process builds on the traditional SLR process [8] and on the Collaborative Design Approach (CDA) [17] for knowledge engineering. Key idea is to loosen the tight connection between SLR data extraction and data synthesis in order to allow collecting knowledge from EMSE studies in a KB, as input to a range of research synthesis methods in a BoK community. Second key goal is to enable incrementally building up knowledge in the context of an EMSE BoK. We evaluate the resulting SKE process regarding effort and added benefits.

SKE aims at designing a common KB data model that captures both, concepts from the BoK domain (e.g., software inspection), and from the selected types of EMSE studies (e.g., controlled experiments). We evaluate the SKE data model regarding the effectiveness to answer the queries of EMSE BoK researchers identified in RI-1.

Research Issue RI-3: SKE Tool Support. Which functions are necessary to automate key steps in the SKE process, i.e., efficient data integration and querying?

For automating SKE a knowledge base (KB) is a major element to provide the desired semantic capabilities. The SKE KB is based on semantic technology with ontologies. Using ontologies makes the KB model extensible and facilitates semantic search [15]. Based on the requirements coming from RI-1 and RI-2 and discussions with SLR researchers the user interface for data import should be based on spreadsheet technology and for querying based on web technologies. We evaluate the tool support regarding the effectiveness and efficiency of data import and querying when applying the SKE process.

IV. SYSTEMATIC KNOWLEDGE ENGINEERING

This section presents the SKE process and its application to build a Software Inspection EMSE BoK KB based on contributions from controlled experiments. The following subsections provide details on how each research issue (requirements analysis, SKE process and data modeling, and SKE tool support) was addressed.
A. SKE Requirements Analysis (RI-1)

Fig. 2 shows how SKE addresses the challenges posed in Fig. 1 by introducing a KB and the role of a knowledge engineer. In this context, researchers extract data from empirical studies published in digital libraries and have that extracted data integrated into the KB by the knowledge engineer. Therefore, the knowledge collected is then available for semantic querying from the KB also to the general readership, including other researchers and practitioners.

To identify the most relevant query candidates to be answered by the software inspection EMSE BoK KB, we focused on EMSE BoK researchers as main stakeholders, who conduct meta-analyses on study reports or conduct empirical studies and need to be aware on relevant research in their area.

Based on an informal survey with software inspection EMSE BoK researchers in six research groups (located in Austria, Brazil, Chile, Ecuador, and Spain), we identified a set of query candidates. The selection of most relevant queries was based on a limited budget of value points, which each stakeholder could spend on the query candidates. Then the query candidates were sorted in descending order by the total number of points of each query. Overall, 10 researchers from the 6 research groups contributed to this selection process. The six most relevant stakeholder queries were:

- Q1: Which inspection methods were effective (or efficient) in finding defects in requirements artifacts?
- Q2: What are the results of experiments that report on a given BoK Topic Parameter <BTP>, e.g., inspection method PBR?
- Q3: Which experiments were conducted with the response variable <RV>, e.g., number of defects?
- Q4: Which hypotheses include the domain concept <DC> (and its synonyms), e.g., effectiveness?
- Q5: Which synonyms have been used for domain concept <DC>, e.g., efficiency?
- Q6: Who are researchers working on topics with response variables in their experiments similar to the domain concept <DC>, e.g., efficiency?

B. SKE Process and Data Modeling (RI-2)

Fig. 3 compares the phases and stages of the SLR [8] and SKE processes. The key innovation of the SKE process comes from decoupling data extraction from data synthesis and integrating extracted data into a KB designed for EMSE BoK building, rather than using the data to apply a particular synthesis method for answering a specific research question in the format of a SLR report (keeping the extracted data from the BoK research community). Thus, the approach allows the community to extend the knowledge gathered during data extraction and reusing it, with the KB’s semantic search facilities, as building blocks for a variety of analyses.

Details on each of the three SKE process stages and how they could be applied to build the software inspection EMSE BoK KB follow.

1) Planning EMSE BoK Creation. Similar to conducting a SLR, the first SKE phase starts with identifying the need and commissioning the creation of the EMSE BoK. Since SKE has a pre-defined purpose of building an EMSE BoK, instead of specifying research questions, SKE just needs to specify the BoK (topics) and the types of empirical studies of interest. In the case of building the inspection EMSE BoK, the BoK was software inspection and the empirical studies of interest were controlled experiments. Then, the SKE protocol is built based on a specific configuration of the PICO (population, intervention, comparison, outcome) strategy [12] to derive search strings that can be applied to digital libraries in the “P and I and C and O” format. In this configuration, the population represents the specified BoK or some of its topics, in our case Software Inspection. The intervention represents the specified empirical study types, in our case Controlled Experiments. The comparison is blank and the outcome represents the elements to extract from the empirical studies (e.g., hypotheses, findings), in our case experimental study results.

As in SLRs, the protocol includes the search strategy with the definition of sources of primary studies (e.g., digital libraries), the study selection criteria and procedure, the quality assessment procedures, and the data extraction strategy. In our case a single digital library was chosen: Scopus, which claims to be the largest database of abstracts [8] and seems sufficient for our study purpose. The study selection and quality assessment criteria were: the study should be an experiment published in a peer-reviewed publication medium. The search string to be applied on Scopus was derived from the PICO synonyms, adding two specific operators: (i) TITLE-ABS-KEY, avoiding searching in the reference metadata; and (ii) W/2, allowing a distance
of up to two words between keywords. The resulting search string was:

`TITLE-ABS-KEY ((software W/2 (inspection OR "defect detection" OR "reading technique")) AND ("experimental study" OR "experimental evaluation" OR "experiment" OR "empirical study" OR "empirical evaluation") AND ("hypothesis" OR "evidence" OR "finding" OR "result"))`

If there is already a BoK KB to build on, the search concepts, including synonyms, can be derived from the BoK KB.

For data extraction, a spreadsheet was prepared to gather relevant experiment data, according to the information to be loaded into the KB common data model on inspection experiments. Once the protocol is defined, the next SKE phase, Conducting Data Extraction, can be accomplished.

2) **Conducting Data Extraction.** This phase consists of following the protocol’s search, selection, and assessment strategies for extracting relevant data. In our case, we executed the search string in Scopus. Overall 156 papers were retrieved. After filtering by title and abstract, a set of 102 papers containing experiments on software inspections were identified, ranging from 1985 to 2013.

A sample consisting of the 30 most recent papers (ranging from 2006 to 2013) was chosen as criteria for data extraction. Six independent local EMSE experts extracted information from those papers into spreadsheets, with an extra expert acting as a data checker. Data extraction took on average about 2 person hours per paper. Data checking took additional 0.5 person hours per paper. As data synthesis is in SKE decoupled from data extraction, the goal is integrating the data into a KB so the BoK community can reuse the knowledge. Thus, the next SKE phase concerns creating the EMSE BoK KB.

3) **Creating/Updating EMSE BoK KB.** In this phase, the knowledge engineer (KE) has to design (or update) the KB common data model and then integrate the extracted data. This role is also responsible for providing query facilities.

For building the software inspection EMSE BoK KB, EMSE experts and the KE designed the data model applying CDA [17] and derived the data extraction spreadsheets. Then the KE provided the KB query facilities to address the most relevant stakeholder queries. More details on the KB common data model for hosting information on software inspection experiments follow.

**Data Modeling for an EMSE BoK KB.** Fig. 4 shows a high-abstraction view on the context in which empirical studies are conducted. Empirical studies have data and artifacts, contribute to a Body of Knowledge (BoK) on specific topics, and are performed by researchers who provide publications. Fig. 5 shows the entities for hosting data on experiments in UML (see [23] for UML-to-OWL transformation), based on these areas and on experimental concepts described by Wohlin et al. [3].

To organize the aggregated knowledge in a flexible way and link knowledge from the empirical studies (in this case, experiments) to inspection BoK topics, each topic was designed as relating to a set of inspection parameters, extended from the list of parameters by Laitenberger and Debaud [22]. For instance, knowledge on requirements (artifact) inspections applying PBR (inspection method) during individual preparation (activity) when conducting Fagan inspections (process). This tailoring is shown in Fig. 6. Note that this final step has to be redone for BoKs in other research topics.

The resulting model allows querying knowledge acquired from experimental studies. Such queries can list hypotheses of experiments related to specific BoK topics parameters (or their synonyms), the results for each hypothesis in the available experiment runs (confirmed/rejected) and information on their statistical confidence. Moreover, measurements that led to each of those results can also be obtained.

**C. SKE Tool Support (RI-3)**

The KB was implemented using the Protégé framework and uses semantic technology with ontologies to facilitate semantic searches. Besides the KB, the tool support comprises a spreadsheet data contribution interface and a web prototype for querying. The data contribution interface was automated in Java by using a spreadsheet reader library (Apache POI) and an ontology library (Apache Jena). The Interchange Standard Approach [18]
can be applied for integrating heterogeneous data. The queries of the web prototype were implemented using SPARQL query language. Using ontology-specific features, the knowledge engineer enhanced the KB by implementing semantic search functions (e.g., searching on domain concepts, their synonyms and related concepts). Additionally, a glossary tool to facilitate identifying and defining domain concepts was also designed and implemented.

The material related to this Section, including the list of queries, the complete data model, the SKE online query prototype and the glossary tool is available online1.

V. EVALUATION RESULTS

This section reports on the evaluation of the results regarding RI-2 and RI-3 based on the required queries coming from RI-1.

A. SKE Process Evaluation

For evaluation, we applied and measured the SKE process steps with tool support to build a software inspection EMSE BoK KB based on knowledge acquired from controlled experiments. Information was extracted from 30 typical research papers and integrated into the KB. While the authors of this paper provided the SKE process and data model design, the data extraction was conducted by an independent expert team.

1) SKE Process Feasibility. The feasibility of applying each SKE phase (planning, data extraction, and building the EMSE BoK KB) was evaluated. The planned SKE PICO configuration effectively supported the identification of relevant experiments on software inspections. The accuracy of the derived search string was evaluated by comparing its results against the software inspection review described in [24]. The experiments reported in this review and indexed in Scopus were successfully retrieved. Therefore, we see the identified papers also representative for new inspection experiments.

Data extraction into spreadsheets containing information on software inspection experiments based on the common KB data model was successfully achieved. Thus, the common data model built for software inspection experiments was effective in characterizing the inspection experiments (modeling similarities and variations) and their results.

Finally, concerning EMSE BoK KB creation, it was possible to create the software inspection KB from published experiment reports. A knowledge engineer conducted the integration of the data from the local spreadsheets into the common KB data model. The resulting EMSE BoK KB was effective and efficient in providing answers to the required queries.

2) SKE Process Effort. Based on our previous SLR experiences, we found the overall effort of applying the SKE process comparable to the effort of conducting SLRs (around 90 person hours). However, the effort of extending the SKE results is likely to be considerably lower (especially if done by other researchers, by allowing directly reusing and extending the extracted data, currently in many SLRs not publicly available). Table 1 shows the effort spent on each SKE phase to build the software inspection KB and an informal effort comparison to the corresponding SLR phase. Planning takes slightly less effort, since SKE applies a predefined PICO configuration. Although in SKE data extraction from primary studies probably takes somewhat more effort than for the SLR, the overall conduct phase takes about the same effort, since SKE does not apply data synthesis in this phase. Finally, creating the EMSE BoK KB is not considered in SLRs.

TABLE I. SKE INSPECTION KB PROCESS EFFORT.

<table>
<thead>
<tr>
<th>SKE Effort (person hrs)</th>
<th>Effort Description</th>
<th>SKE vs. SLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning EMSE BoK Creation</td>
<td>8</td>
<td>Building the protocol.</td>
</tr>
<tr>
<td>Conducting Data Extraction</td>
<td>80</td>
<td>Filtering primary studies and data extraction.</td>
</tr>
<tr>
<td>Creating EMSE BoK KB (Population)</td>
<td>&lt; 0.05</td>
<td>KB data integration (automated).</td>
</tr>
</tbody>
</table>

3) SKE Process Added Value. We are aware that SKE and SLR processes have different purposes (SLRs usually seek for evidence-based answers to research questions, while SKE focuses on building EMSE BoKs) and that one does not replace the other. However, since SLRs are widely used [5], we ascertain that, given the similar effort, it makes sense to apply SKE for building EMSE BoKs. Moreover, in this specific context, when compared to SLRs, SKE presents the following benefits:

- SKE integrates extracted data into a KB and facilitates the reuse of the aggregated knowledge by other researchers according to their specific goals. SLRs usually focus on specific research syntheses and extracted data is mostly stored in local spreadsheets, seldom publicly available.
- SKE facilitates building up knowledge incrementally by integrating new extracted data into the KB.
- The SKE KB allows exploring the gathered empirical evidence using semantic search capabilities that cannot be performed on SLR reports. For instance: “Which results were obtained for hypotheses investigated in BoK topic <BT> using the response variable <RV> (or any of its synonyms)?”

Note that SKE required some setup effort by the knowledge engineer for creating the KB’s ontology model, creating the spreadsheet importer, providing the query facilities, and developing a suitable user interface. Table 2 shows this setup effort.

TABLE II. KNOWLEDGE ENGINEER EFFORT.

<table>
<thead>
<tr>
<th>Effort Description</th>
<th>Know. Eng. Effort (person hrs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating EMSE BoK KB ontology model</td>
<td>16</td>
</tr>
<tr>
<td>Spreadsheet Importer Creation</td>
<td>40</td>
</tr>
<tr>
<td>Query creation</td>
<td>32</td>
</tr>
<tr>
<td>UI Development</td>
<td>40</td>
</tr>
</tbody>
</table>

B. SKE Data Model Evaluation

Having evaluated the feasibility of applying the SKE process, interest sprouts in evaluating the resulting software inspection EMSE BoK KB and its underlying common data model against the required queries coming from RI-1. Therefore, the

1 SKE Prototype: http://cdllflex.org/prototypes/ske
knowledge engineer formulated the queries in the KB language so that results for them could be obtained. To evaluate these results, an independent researcher built a set of query test cases based on the 30 papers included in the KB. Queries Q2 to Q6 could be directly formulated, listing experiment results, experiment hypotheses, synonyms, and research groups. The related test cases passed successfully with the query results, which can be accessed in the online prototype.

However, query Q1: “Which inspection methods were effective (or efficient) in finding defects in requirements artifacts?,” addressing a practitioners’ need could not be answered directly and had to be translated into terms of the underlying data model. The first decision was to focus on experiments that reported on effectiveness or efficiency (or synonyms) in their hypotheses or response variables and that were related to BoK Topics associated to inspection methods and to the artifact type requirements. For allowing semantic search on synonyms the additional domain concepts KB technology and the glossary were used. The test case built for this query was related to identifying the right set of experiments and passed successfully. Then, to answer the query, it was split into two separate queries that would provide an overview on the knowledge of interest out of those experiments. Q1.1 was to list the hypotheses and their results in all experiment runs. Q1.2 was to show the findings of all papers related to them.

Fig. 7 shows results of Q1.1 enabling stakeholders to see which inspection methods were reported as effective. Stakeholders can also see other focused and interesting information. For instance, that defects detected with PBR are more evenly distributed over the document, when compared to using checklists. Regarding Q1.2, it provided an insightful overview on findings of the papers on effectiveness of inspection methods. Thus, the evaluation of all required queries passed successfully in the scope of the SKE research prototype.

C. SKE Tool Support Evaluation

This evaluation concerned the support for data importing and querying. The spreadsheet data contribution interface was effective in enabling contributions from researchers and efficient by requiring low effort and little time for such contributions to be imported. Data from 6 spreadsheets, containing data on 30 experiments and including over 6,000 data elements (cells) were integrated into the KB in less than 3 minutes. The simple web interface prototype allows stakeholders to easily retrieve query results within a few seconds.

VI. DISCUSSION

In this section, we discuss for each research issue the evaluation results, possible limitations, and lessons learned.

A. RI-1: SKE Requirements Analysis

The SKE process and its resulting KB support users well in locating empirical evidence and reusing it according to specific needs. However, it is important to state that support for meta-analyses and for applying specific research syntheses methods are beyond the scope of SKE. Moreover, the relevant queries of our evaluation were chosen focusing on a survey with a specific type of stakeholder, the EMSE BoK researchers. Other stakeholders may have different information needs. Jedlitschka et al. [25], for instance, gathered empirical study information needs from 175 managers from industry. According to their findings, there are three categories of information needs for these stakeholders from experiment reports: technology, context, and impact. We believe that these information needs could be provided by semantic queries applied to the SKE KB. However, addressing practitioners’ needs still has to be evaluated.

B. RI-2: SKE Process and Data Modeling

The SKE process evaluation showed the feasibility of building a software inspection EMSE BoK from controlled experiments. Moreover, the overall effort was comparable to the effort of conducting a SLR and, for the specific purpose of building EMSE BoKs, several advantages of using SKE could be confirmed. However, it is important to state that the SKE and SLR processes have different purposes. The suitability of applying one or the other will depend on the specific research goals. In fact, if an EMSE BoK is built following SKE, the resulting KB may be used as input to an SLR protocol to identify relevant research on a specific topic, exploring semantic search facilities usually not available in digital libraries. SLR data extraction sheets, on the other hand, can be used as input for extending the knowledge contained in the SKE KB.

C. RI-3: SKE Tool Support

The simple user interface for data importing and querying enables researchers to contribute building up additional knowledge, and to query for knowledge with low effort. Using ontologies facilitates extensions of the underlying KB common data model and semantic search. The querying capabilities were found efficient in the evaluation on answering the EMSE BoK researchers’ most relevant queries.

D. Threats to Validity and Lessons Learned

The performed evaluation faces some threats to validity. A major threat is related to the decision of applying the approach on building a software inspection EMSE BoK based on study reports from controlled experiments. Software inspections are widely spread in academia and industry and many empirical studies were conducted in this area, which may have facilitated building up the EMSE BoK. Additionally, SKE can also be applied to gather knowledge from other types of empirical studies but the feasibility of this process needs to be investigated. Besides, the relevant queries used to evaluate the KB were obtained from a limited and informal survey.

Main lessons learned relate to success factors for applying SKE. A major success factor is properly involving a knowledge
The overhead of this new role is likely to be offset soon by benefits obtained by the established KB. Another success factor is getting the EMSE BoK community involved for long-term collection and use of data. Therefore, incentives and benefits of contributing should be clarified.

VII. CONCLUSIONS

In this paper we introduced the Systematic Knowledge Engineering (SKE) process, which supports building up EMSE BoKs from empirical studies. SKE builds on the Systematic Literature Review (SLR) process and provides a Knowledge Base (KB) as storage for extracted data. By decoupling data extraction from data synthesis and providing a KB, SKE allows the community to extend gathered knowledge (even if the original authors are no longer available) and reusing it with semantic search facilities, as building blocks for a variety of analyses.

SKE was evaluated by building a software inspection EMSE BoK from knowledge acquired through controlled experiments. Information was extracted from 30 research papers and integrated into the KB. The resulting software inspection EMSE BoK is available online1. Main evaluation results were:

- SKE’s suggested PICO configuration was successful in identifying relevant experiments on software inspections.
- Data extraction of inspection experiments into spreadsheets based on the common data model was successful.
- It was possible to create the software inspection EMSE BoK KB from published experiment reports.
- The KB was effective and efficient in answering the most relevant stakeholder queries.
- SKE enables knowledge reuse (by applying queries) for analysis and meta-analysis purposes. Moreover, new knowledge, i.e., new data from literature, can be included in the KB as a foundation for a growing EMSE BoK.

The SKE process showed promising results in the software inspection context and should also be evaluated in other contexts. The overall effort of applying SKE is comparable to the effort of conducting SLRs and, for the specific purpose of building EMSE BoKs, several advantages of using SKE could be identified. As future work we propose: (a) investigating to address different EMSE BoK stakeholder needs; (b) extending the set of empirical studies on software inspections in the KB; (c) instantiating SKE in other contexts; and (d) setting up a platform to allow building on the collective intelligence of the research community for quality assurance & recommendation.

ACKNOWLEDGMENT

This work was supported by the Christian Doppler Forschungs-gesellschaft, the Federal Ministry of Economy, Family and Youth and the National Foundation for Research, Technology and Development - Austria, and CNPq - Brazil.

REFERENCES