Ontology-Based Identification of Research Gaps and Immature Research Areas*

Kristian Beckers[1], Stefan Eicker[1], Stephan Faßbender[1], Maritta Heisel[1], Holger Schmidt[2], and Widura Schwittek[1]

[1] University of Duisburg-Essen, paluno - The Ruhr Institute for Software Technology
{firstname.lastname}@paluno.uni-due.de
[2] ITESYS - Institut für technische Systeme GmbH, Germany
h.schmidt@itesys.de

Abstract. Researchers often have to understand new knowledge areas, and identify research gaps and immature areas in them. They have to understand and link numerous publications to achieve this goal. This is difficult, because natural language has to be analyzed in the publications, and implicit relations between them have to be discovered. We propose to utilize the structuring possibilities of ontologies to make the relations between publications, knowledge objects (e.g., methods, tools, notations), and knowledge areas explicit. Furthermore, we use Kitchenham's work on structured literature reviews and apply it to the ontology. We formalize relations between objects in the ontology using Codd's relational algebra to support different kinds of literature research. These formal expressions are implemented as ontology queries. Thus, we implement an immature research area analysis and research gap identification mechanism. The ontology and its relations are implemented based on the Semantic MediaWiki+ platform.

Keywords: ontologies, research gaps, knowledge management, facetted search.

1 Introduction

Getting an overview of existing engineering methods, tools and notations (referred to as Knowledge Objects – KOs) for specific fields (referred to as Knowledge Areas – KAs) is of major importance for software engineering researchers. This knowledge is the basis for finding research gaps and problems in this field, which require their attention. Our objective is to develop a technique for finding missing methods, notations and tools in specific knowledge areas.

Researchers usually have to rely on their experience during a research area analysis, which includes the activities of finding research gaps and identifying research areas. This can lead to a biased outcome of a research area analysis. Hence, research gaps or immature research areas might be overlooked repeatedly. In addition, researchers have to find relations between publications, which are sometimes implicit.

* This research was partially supported by the EU project Network of Excellence on Engineering Secure Future Internet Software Services and Systems (NESSoS, ICT-2009.1.4 Trustworthy ICT, Grant No. 256980).

G. Quirchmayr et al. (Eds.): CD-ARES 2012, LNCS 7465, pp. 1–16, 2012.

In order to ameliorate this situation, we propose a structured approach for research area analysis. This approach utilizes the extensive research of Kitchenham et al. [1–8] for structured literature reviews. We apply Kitchenham's methods to a special *ontology*, the Common Body of Knowledge (CBK) of the EU project *Network of Excellence (NoE) on Engineering Secure Future Internet Software Services and Systems (NESSoS)*[1]. One of the major goals of this NoE is the integration of the disciplines of software, service, and security engineering. Hence, the CBK ontology contains information about these areas in numerous KOs that refer to KAs.

Our approach is threefold: We carry over Kitchenham's research for structured literature reviews to informal queries for the CBK, and we also extend the CBK to support these informal queries. In the next step we refine these informal queries into formal CBK relations using Codd's relational algebra [9]. For this purpose, we apply the *DOOR* method by Allocca et al. [10] for capturing the semantics of relations in ontologies and to formally specify these relations. The technical realization of the CBK is a *Semantic MediaWiki+* platform, and we implemented the relational algebra expressions as CBK queries.[2]

The queries result in tables that show the relations between KAs and KOs. The tables also contain the information of *how many* KOs are in a KA and *what kind of* KOs exist in it, e.g., methods, tools, techniques, and notations. These compact results of a query are more effective than analyzing the natural language in publications. Moreover, the creation and execution of a query in the CBK is less time consuming than finding relevant literature for a KA and analyzing it.

The paper is organized as follows: we explain background about structured literature surveys and the NESSoS CBK in Sect. 2. We present in Sect. 3 a structured research area analysis method, which contains research gap analysis and immature research area identification. We show in Sect. 4 our realization of the approach for the field of engineering secure software and services using the NESSoS CBK. Section 5 presents related work. Finally, we conclude and raise ideas for future work in Sect. 6.

2 Background

We explain Kitchenham's structured approach to structured literature reviews in Sect. 2.1 and the basic structure of the NESSoS Common Body of Knowledge in Sect. 2.2.

2.1 Literature Research According to Kitchenham

To gain a structured overview of the state of the art and existing literature before starting new research is one fundamental element of scientific work. For the area of software engineering, Kitchenham was one of the first, who described a structured literature review process [1]. Over the years this inital process was extended and improved by Kitchenham herself and others [1–8].

There are several reasons and goals why researchers might want to perform a literature review. And there are also different types of literature reviews, which can serve

[1] http://www.nessos-project.eu/

[2] http://www.nessos-cbk.org

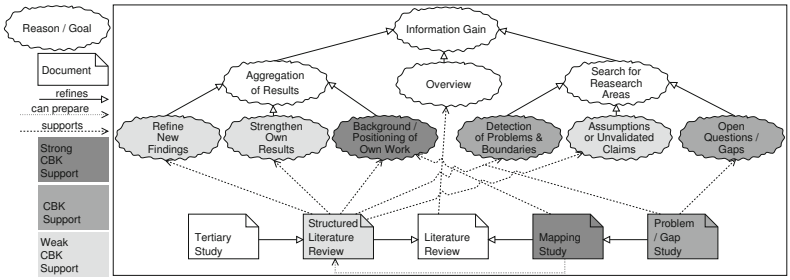

Fig. 1. Types of literature researches and reasons and goals to perform them [1–8]

the different goals. Fig. 1 shows a condensed view of findings and statements from different publications in the field of systematic literature reviews in software engineering [1–8]. The overall reason to do any kind of literature research is Information Gain. This top-level goal can be refined into the goal to get a mere Overview without a specific motivation [1]. In contrast, the goals of Aggregation of Results and Search for Research Areas have a well-founded motivation.

When aggregating results, one might want to Refine New Findings based an the aggregated data [1, 2]. Or the findings and data of other publications are used to Strengthen Own Results [1, 2]. A last reason for aggregation is to give a Background / Positioning of Own Work [4, 5, 7, 8]

When searching for research areas, a Detection of Problems & Boundaries [1, 2, 5, 6] of a certain method or set of methods can be the goal. Another option is to search for Assumptions or Unvalidated claims [1, 2, 5, 6]. These two sub-goals aim at finding immature research areas and improve them with further research. In contrast, finding Open Questions and Gaps aims at research fields, where no publications about solutions exist [2, 5, 8].

All types of Literature Reviews support the goal of obtaining an overview. The quality of the overview differs in how structured and planned the literature review was performed. A special type is the Structured Literature Review (SLR) [1–3, 6]. A SLR is a comprehensive literature review considering a specific research question. Kitchenham's method to perform a SLR was developed to find empirical primary studies considering a specific question and to aggregate the data in the first place. Additionally it turned out later that SLRs also make it possible for researchers to find immature research areas [5, 6]. A special kind of the SLR is the Tertiary Study [2, 4]. Such a study aggregates the results of other SLRs, hence it relies on secondary studies. Another type of literature review is the Mapping Study [2, 5, 7, 8]. Here it is not the aim to extract any data, but to map studies to research fields or problems. When also the problems and gaps discussed in the studies are obtained while doing the mapping, a Problem / Gap Study as a special kind of mapping study is the result [2, 8, 7]. This kind of study serves to find real gaps or to find immature methods. For assessing immaturity, a gap study should be combined with a SLR.

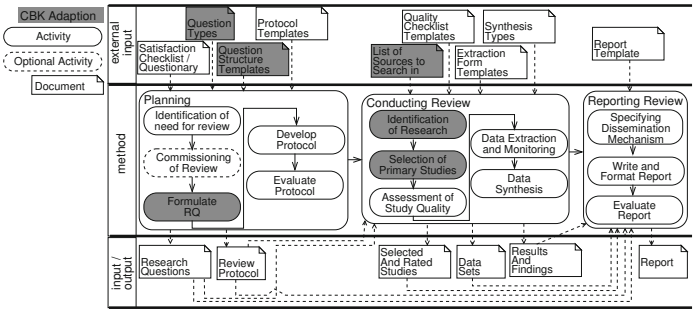

Fig. 2. Process proposed by Kitchenham for undertaking a SLR [1, 2, 6, 3]

Kitchenham et al also propose a process to conduct SLRs [1–3, 6]. It is shown in Fig. 2. This process can be also used for mapping studies with some slight adaptions [2, 5, 8]. The process is split up in three major phases. First the Planning phase takes place, followed by the Conducting Review phase, and finally the Reporting Review phase ends the review process.

The Planning starts with an Identification of need for review step The next step is optional as in Commissioning of Review the SLR is tendered to other research groups. The first real step towards a SLR is to Formulate RQ (Research Questions). The research questions are the core of an SLR. All later decisions and results are checked against the RQ later on. In Develop Protocol the review itself is planned. The step Evaluate Protocol is performed to detect misunderstandings, ambiguities, and insufficient definitions. To ease the planning there are predefined Satisfaction Checklists / Questionaries, Question Types, Question Structure Templates, and Protocol Templates. For Question Structure Templates, Kitchenham and Chaters proposes to use the PICOC criteria framework to structure research questions [2]. PICOC stands for the criteria *population*, e.g. application area or specific groups of people, *intervention*, e.g. the method which is of interest, *comparison*, e.g. the benchmark, *outcomes*, what is the improvement to be shown, and *context*, a description of the setting in which the comparison takes place. All these documents serve as an input and guide for certain planning steps. The result of the planning phase are the Research Questions and the Review Protocol. They serve as input to the Conducting Review phase.

The review starts with the Identification of Research, which results in a set of studies which might be relevant. According to defined inclusion and exclusion criteria the step Selection of Primary Studies is performed. The selected studies are then rated in the step Assessment of Study Quality. For those studies with a satisfying quality level the data contained in the studies is extracted in the step Data Extraction and Monitoring. Afterwards the Data Synthesis is performed. The input to this phase are a List of Sources to Search In, Quality Checklist Templates, Extraction Form Templates, and Synthesis Types. Outputs produced in the conducting review phase are the Selected And Rated Studies, the Data Sets extracted form these studies, and the Results and Findings of the data synthesis.

All previously generated outputs serve as an input for the last phase Reporting Review. As external input, Report Templates are given. Based on the inputs, the step Specifying Dissemination Mechanism is executed Then the report is actually written in the step Write and Format Report. As last activity a Evaluate Report step is performed The Report is the output of the entired SLR process.

2.2 NESSoS Common Body of Knowledge

Ontologies are used to capture knowledge about some domain of interest. In our case, that domain is the field of engineering secure software and services. An ontology provides a catalogue of the classes of objects that exist in the domain. Moreover, an ontology consists of relations between these classes, and of the objects contained in the classes. We present the ontology we use in this paper in Fig. 3 as a Unified Modeling Language (UML) class diagram.[3], this ontology presents the subset of the CBK, which is relevant for this work. The classes in light grey represent the most relevant classes in our ontology for this work, and the classes in dark grey are classes that inherit from the most relevant classes.

The class KnowledgeArea divides the field of secure software and services into knowledge areas (KA). The central class in our ontology is the class KnowledgeObject, which represents all types of knowledge objects (KO) we want to capture. As examples, we consider the KOs of the types Tool, Method, and Notation. The equally named classes inherit general properties from the class KnowledgeObject. In general, the properties that are inside of a class box are simple properties, e.g., of type String or Boolean, while there also exist structured properties connected to class boxes via associations. Simple properties are, for instance, contextDescription, problemDescription, and solutionDescription, which represent textual descriptions of the context, the tackled problem, and the solution for tools, methods, and notations. These properties are part of the class KnowledgeObject. An example for a structured property is the association publications, which connects the class KnowledgeObject and the class Publication. This property is structured, because every publication consists of a BibTeX entry or links to DBLP[4] (bibtexEntriesOrLinksToDBLP), and a flag indicating the importance of a publication (isPrimaryLiterature).

The class CommonTerm has several defined terms, and these can be related to terms of KOs. Moreover, some structured properties refer to enumeration types labeled with the UML ≪enumeration≫ stereotype, e.g., the association maturityLevel that connects the class KnowledgeObject and the class MaturityLevel. This enumeration type allows us to rate every tool, method, and notation according to its maturity.[6]

Multiplicities at the association ends specify constraints on the number of elements contained in an association end. For instance, the 1 at the association end of the association maturityLevel describes that each KO has exactly one maturity level. KOs have

[3] http://www.uml.org/

[4] http://www.informatik.uni-trier.de/~ley/db/

[5] The UML stereotype ≪enumeration≫ is used for classes that have a fixed set of attributes, which are referred to by other classes. This use differs from the specification in the UML standard.

[6] In general, enumeration types allow us to pre-define values a property might have.

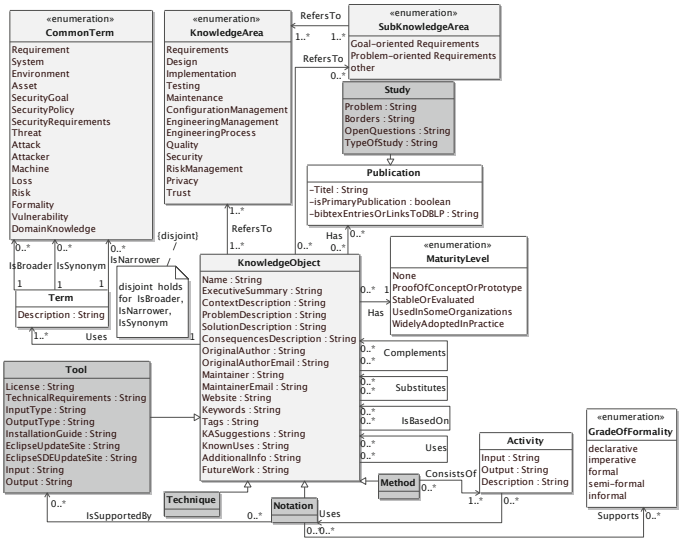

Fig. 3. An Ontology for the Common Body of Knowledge for Secure Software and Services[5]

several relations between each other, e.g., that one KO Uses another. However, these were extended for the contribution of this work and, hence, we explain them in Sect. 3.2.

Specific types of KOs are Notations, which can be supported by Tools (IsSupportedBy). A Notation supports a grade of formality (GradeOfFormality). Methods can be divided into Activities. Activities (Activity) help to structure Methods and to describe workflows based on Inputs, Outputs, and a Description. This means that an activity can use the output of another activity as input. Techniques have just one action, which makes them less complex than Methods.

Using the presented ontology structure, we can adequately capture and process knowledge in the field of engineering secure software and services. In addition, we evolved the ontology for this work with the elements Term, Study, and SubknowledgeArea and we also extended the attributes of KnowledgeObject. We explain these new elements in Sect. 3.2.

3 Structured Research Area Analysis

This section describes the main scientific contribution of this paper. We present in Sect. 3.1 how our work integrates into the approach of Kitchenham described in Sect. 2.1. We show the preparation of our ontology in order to support research area analysis in Sect. 3.2. We describe how to support research area analysis in Sect. 3.3. In Sect. 3.4, we specify the relations between the different ontology parts in detail. These relations enable us to substantiate the envisaged research area analysis by the semantics of the knowledge stored in the ontology.

3.1 Extension/Integration of Kitchenham and CBK

There are several points of integration for the CBK and the literature review process introduced in Sec. 2.1. These points are the inputs Question Types, Question Structure Templates, and the List of Sources to Search In, and the process steps Formulate RQ, Identification of Research, and Selection of Primary Studies (see Fig. 2). To improve the integration for some of these points, the CBK and / or the literature review process have to be adapted.

Question Types. The original question types defined by Kitchenham at al are formulated for SLRs [1, 2]. For mapping studies, a selection of questions and their generalization can be found in the works of Kitcheham et al [5], and Petersen et al. [8].These insights combined with the structure of the CBK (see Fig. 3) result in some new question types:

- How many different KOs exist for the KA in question?
- Which KA(s) are covered by a certain KO?
- Which are the problems and future work mentioned for (a) given KA(s)?
- What is the maturity of KOs for (a) given KA(s)?
- What are the main publications for a given KO or KA?

Question types help to formulate research questions. They give evidence which questions can be of interest and how to formulate them. Moreover, whenever a research question within an actual review maps to one of the questions types given above, this question can be answered by the CBK directly.

Question Structure Templates. For the question structure template, Kitchenham and Chaters proposes to use the PICOC criteria framework to structure research questions [2] as we already described in Sect. 2.1. But when investigating the CBK meta-model (see Fig. 3) it seems to be reasonable to add some criteria.

The main addition is to define the knowledge area(s) explicitly, unlike having them implicit in the context. In most cases of conducting a SLR or mapping study, there is a very specific focus on a special part of software engineering. An overview given by Kitchenham et al. shows that evidence [4]. This focus should be captured within the criteria, because some electronic sources support to select knowledge areas [6, 2]. Moreover, for mapping studies these knowledge areas are often used for structuring the report [5, 7, 8]. In the case of the CBK the knowledge areas are one of the core concepts, and searching the CBK utilizes the knowledge areas.

A minor addition is to distinguish between general terms of the population and special terms of the knowledge area(s). The special terms have a great weight when searching and can help to structure the results [6, 5].

Using this new question structure makes important parts of the questions more explicit. And they ease the use of the CBK, because the separation between general search terms,common terms and knowledge areas are directly reflected in the formulation of the search queries (We will see in Sect. 3.4).

List of Sources to Search In. The CBK and its searching capabilities has to be added to the list of sources to search in [2, 6]. Adding the CBK itself is trivial, but for the capabilities it has also to be checked which new capabilities the CBK introduces. For example, the missing relations between KOs and between publications, which

has been found as issue for all existing search sources [6], is explicitly addressed in the CBK. A detailed discussion is skipped at this point due to the lack of space.
Having the CBK in the source list with an explanation of its capabilities helps to plan the research and clarifies in which cases the CBK is of superior use in comparison to other sources.

Formulate RQ. Besides the two inputs, the Question Types and the Question Structure Templates, the CBK can directly support the formulation of research questions. The CBK already defines an ontology of knowledge areas. With an extension, it will support sub-knowledge areas (see Sec. 3.2). Those areas can help to focus the research questions, because it gives an orientation how to refine knowledge areas and how knowledge areas are related. To find the focus for the own research is considered challenging without such a support [3, 4, 7]. Additionally the adapted CBK presented in Sec. 3.2 provides the common terms used in these knowledge areas. These terms help to sharpen the questions and to avoid ambiguities.
The integration of the CBK and Formulate RQ improves the outcome of the whole literature review process, because it helps to avoid the formulation of imprecise search questions with respect to missing / wrong focus and wrong / ambiguous wording.

Identification of Research. The ontology of common terms and related synonyms contained in the CBK also helps to formulate the search queries, not only for the CBK, but also for other search sources.
This formulation of search terms is a crucial step and the knowledge about relations between terms and the existence of a synonym list improves this step of finding research a lot [1–8].

Selection of Primary Studies. For the selection of primary studies the CBK provides information for some comprehensive and sophisticated selection criteria. The CBK already contains information, which is very specific and useful to rate KOs, like the maturity level. And this information is available for all results obtained from the CBK. These criteria are hard to evaluate for results from other search sources [2, 3, 6]. Examples for inclusion criteria:

- Only include KOs and related publications with a certain maturity level
- Only include KOs, which are a core concept of a KA and therefore many other KOs are based on these KOs
- Only include publications, which are considered as most significant by the editors of KOs

Having a set of precise selection criteria, which can be evaluated for all results, improves the outcome of the whole review process. The probability of excluding relevant studies and the bias caused by including literature of low quality can be reduced. And the whole review process speeds up when using the CBK, because the information for evaluating the criteria is available explicitly.

3.2 Preparing the Ontology

We prepared the NESSoS CBK ontology (see Sect. 2.2) in order to support the concepts of the structured literature reviews from Kitchenham explained in Sects. 2.1 and 3.1. Hence, we extended the CBK ontology (see Fig. 3) with the classes SubKnowledgeArea, Publication, and Study.

We consider keywords and tags in knowledge objects. Keywords are given according to the guidelines of a specific system, e.g., the ACM-keyword-system [11]. Tags are chosen without any restriction and provide the possibility to choose any possible word.

We want to use the CBK ontology for finding research gaps. This requires KnowledgeObjects that do not have a solution description yet, but only problem descriptions. We also included the MaturityLevel None for these KnowledgeObjects. In addition, KnowledgeObjects now have a FutureWork attribute that states research not yet carried out. The difference between these two is that KnowledgeObjects that only have a problem description represent a research area that needs a significant amount of research for providing a solution. The FutureWork attribute in KnowledgeObjects represents possible research that can build upon an existing solution. Researchers that use our approach might look for one or the other.

KnowledgeObjects have relations between each other. These relations are relevant for the investigation of research areas. KOs can be based on other KOs (IsBasedOn), and they can use each other (Uses). In addition, KOs can be used in combination. In this case, one KO Complements another. If KOs can be exchanged, these have a Substitutes relation.

We include the class Term that holds the terms a specific KnowledgeObject uses. The class CommonTerm contains a set of well-defined terms. A Term of a KnowledgeObject is either broader, synonymous, or narrower than a CommonTerm. This allows a comparison of KnowledgeObjects using the CommonTerms. Without these any comparison would lack precision, because terms and notions differ in KnowledgeObjects.

3.3 Identifing Research Gaps Using the CBK

While a Mapping Study and a Problem/Gap Study is supported best by the CBK, it at least partially supports a Structured Literature Review and a Tertiary Study (see Fig. 1). In the following, the CBK support for all different kinds of studies, as defined by Kitchenham, is described in more detail. The support, as described in this section, sketches only a first idea, which will be refined in Sect. 3.4, where we only consider the well supported Mapping Study and Problem/Gap Study.

Mapping Study. For a Mapping Study researchers have to specify one or more KAs, one or more sub-KAs or one or more common terms . Additionally, they can constrain the search by providing further search terms. The CBK returns KOs grouped by KAs. Depending on the scope of the search, 20 results,for example , for one KA can be interpreted as a sign for maturity of a field or immaturity of a field. For example, 20 results for the KA Requirements has a different meaning than 20 results for the sub-KA Goal-oriented Requirements including the search terms *Cloud* and *Law*.

Problem/Gap Study. Conducting a Problem/Gap Study is also well supported by the CBK. The researcher specifies one or more KAs, one or more sub-KAs or one or more common terms and provides search terms. In this case, the CBK not only groups KOs along the specified KAs, sub-KAs or common terms as described for the Mapping Study, but extends the search to the following classes and fields of the CBK ontology: ProblemDescription and FutureWork of the class KnowledgeObject,

Problem and Border in the class Study. These results support the creation of a Problem/Gap Study because all relevant information is presented to the researcher in a structured way.

Structured Literature Review. While the researcher is able to retrieve KOs for the selected KAs, sub-KAs or common terms and, thus, all relevant literature references, can only be regarded as a starting point to conduct a full-fledged Structured Literature Review. A SLR involves an in-depth analysis of the actual literature, which is out of the CBK's scope.

Tertiary Study. Conducting a Tertiary Study is only supported in theory, because it requires all relevant secondary studies to be part of the CBK. If this is not the case, there is no support for this kind of study using the CBK.

3.4 Formalizing Research Area Analysis

We now identify and specify relevant relations for the identification of research gaps, making use of the ontology and the knowledge it stores and structures. For the analysis, the relations between different tools, methods, and notations, i.e., different KO types, and KAs are of particular relevance.

Allocca et al. [10] present the *DOOR* method to capture the semantics of relations between different ontologies and to formally specify these relations. While we partly adopt the DOOR steps to support our approach to identify and specify relations between different KOs, we abstain from building an ontology of these relations. We use the ontology structure presented in Sects. 2.2 and 3.2 on the one hand for typing the relations and, more importantly, on the other hand to refine the semantics of the relations. We divide our approach into the following three steps:

1. Identify and specify top-level relations
2. Identify and specify variants and sub-relations, and characterize their algebraic properties
3. Compose relations

We will use the following abbreviations in the formalization: KA = Knowledge Area, SKA = SubKnowledgeArea KO = Knowledge Object P = Publication, ST = Study, CT = CommonTerm, T = Term, and ML = MaturityLevel.

Top-Level Relations: The following relations are abstracted top-level relations that support the kinds of queries sketched in Sect. 3.3.

MappingStudy_KAxKO. Describes a mapping study as a relation between knowledge areas and knowledge objects.

MappingStudy_SKAxKO. Describes a more fine granular mapping study as a relation between sub-knowledge areas and knowledge objects.

MappingStudy_CTxKO. Describes a mapping study as a relation between common terms and knowledge objects.

ProblemGapStudy_KAxKOxST. Describes a problem or gap study as a relation between knowledge areas, knowledge objects, and studies.

ProblemGapStudy_SKAxKOxST. Describes a more fine granular problem or gap study as a relation between sub-knowledge areas, knowledge objects, and studies.

ProblemGapStudy_CTxKOxST. Describes a problem or gap study as a relation between common terms and knowledge objects and studies.

Variants, Sub-relations, and Algebraic Properties: Variants and sub-relations shed light on various facets of the top-level relations with regard to the structured design of the ontology. We express the relations using relational algebra based upon the work of Codd [9]. We use an extention of the relational algebra [12, 13] that offers aggregation and grouping functionalites. The symbol ξ groups the output according to specifed atribute(s). π projects only specified columns of a table. σ selects rows in a table for which specified boolean expression(s) hold. $\bowtie$ joins tables according to common attributes. All rows that do not have these attributes are left out. $\rtimes$ joins tables, but also displays rows of the left table that do not have all the common attributes. We use relational algebra, because the algebra expressions can be translated to SMW+ queries in a straightforward way, see Sect. 4.

For the specifications of the relations, we assume that the structural design of the ontology presented in Sect. 2.2 and 3.2 is given as tables. Classes that have 1..* cardinalities on both ends of the relation in our ontology require connection tables. Otherwise we would require multiple relations between tables, which is to be avoided during database design. For example, we want to create a table for knowledge objects. One row in the table is allowed to have multiple relations to rows in the knowledge area table. Instead, we would have to create numerous columns for these relations in the knowledge area, because we do not know how many relations we need. Hence, we create a further table for these relations. We denote these tables, which we add in the formalization, with "Connect" and append the names of the classes this table connects. For example, the connection table for the tables KnowledgeObject and KnowledgeArea is stated as: ConnectKAtoKO. These connection tables have two columns, which contain the the primary keys of each of the tables they connect.

Inheritance in the ontology is translated into one main table for the superclass and one table for each class that inherits from this class. These classes have a relation to the table that represents the superclass and have only the additional attributes of the inherited class. For example, the superclass KnowledgeObject has the class Tool that inherits from it, and one of the additional attributes is Input. Hence, we create a table KnowledgeObject and a table Tool, which has the attribute Input.

Searchterm: We define an algebraic expression ST, which represents a boolean expression for one or more searchterms.

$$ST ::= ST \diamond ST \mid \kappa = String \mid (ST) \mid \neg ST$$
$$\diamond ::= \wedge \mid \vee$$
$$\kappa ::= Tag \mid Keywords \mid ExecutiveSummary \mid Name$$

For example, the expression " Tags='cloud' $\vee$ Tags='law' " can be used with σ for the table KnowledgeObject. This results in a table with all KOs that have 'cloud' or 'law' as tags.

KnowledgeArea: We define KAB to be a boolean expression for the selection of one or more knowledge areas.

$$KAB ::= KAB \vee KAB \mid KnowledgeArea = KAS$$
$$KAS ::= Requirements \mid Design \mid Implementation \mid Maintenance \mid$$
$$ConfigurationManagemen \mid EngineeringManagement \mid EngineeringProcess \mid$$
$$Quality \mid Security \mid RiskManagement \mid Privacy \mid Trust$$

We defined Maturity Level (MLB), Sub-KnowledgeArea (SKAB), and Common Terms (CTB) in a similar manner.

MappingStudy_KAxKO: We specify queries for mapping studies as database relations. We explain the query in detail starting with the σ_{ST}, where we join the tables KA, ConntectKAtoKA, and KO and select rows accordng to ST. The query filters the resulting table for rows that have the required MLB and KAB. The result is projected onto the columns KnowledgeArea, MaturityLevel, Name, ExecutiveSummary, Tags, and Keywords. The query groups the results according to KnowledgeArea, MaturityLevel, Name, ExecutiveSummary, Tags, and Keywords.

$$
\begin{aligned}
&\mathit{MappingStudy\_KAxKO} = \\
&\xi_{\mathit{KnowledgeArea,MaturityLevel,Name,ExecutiveSummary,Tags,Keywords}}(\\
&\quad \pi_{\mathit{KnowledgeArea,MaturityLevel,Name,ExecutiveSummary,Tags,Keywords}}(\\
&\quad\quad \sigma_{KAB}(\sigma_{MLB}(\\
&\quad\quad\quad \sigma_{ST}(\mathit{KA} \bowtie \mathit{ConnectKAtoKO} \bowtie \mathit{KO})))))
\end{aligned}
$$

We define MappingStudy_SKAxKO, and MappingStudy_CTxKO in a similar manner.

Extending Searchterm: For a problem gap study we extend the fields of the CBK that can be searched as follows.

$$
\begin{aligned}
&\ldots \\
&\kappa ::= \mathit{Tag} \mid \mathit{Keywords} \mid \mathit{ExecutiveSummary} \mid \mathit{Name} \mid \mathit{FutureWork} \mid \\
&\mathit{Title} \mid \mathit{Problems} \mid \mathit{Borders} \mid \mathit{OpenQuestions}
\end{aligned}
$$

ProblemGapStudy_KAxKOxST: To perform problem gap studies, we include existing studies in the search relation and enriches the output with problem and future work descriptions. We formalize this relation as an variant of the relation MappingStudy_KAxKO. The symbol $\rtimes$ between KO and $(\mathit{ConnectKOtoP} \bowtie P \bowtie \mathit{ST})$ causes that also KOs are selected that do not have a publication or study.

$$
\begin{aligned}
&\mathit{ProblemGapStudy\_KAxKOxST} = \\
&\xi_{\mathit{KnowledgeArea,Name,ExecutiveSummary,FutureWork,Tags,Title,Problem,Borders,OpenQuestions}}(\\
&\quad \pi_{\mathit{KnowledgeArea,Name,ExecutiveSummary,FutureWork,Tags,Title,Problem,Borders,}} \\
&\quad\quad {\scriptstyle \mathit{OpenQuestions}}(\\
&\quad\quad\quad \mathit{MappingStudy\_KAxKO} \bowtie \mathit{KO} \rtimes (\mathit{ConnectKOtoP} \bowtie P \bowtie \mathit{ST})))
\end{aligned}
$$

We define ProblemGapStudy_SKAxKOxST, ProblemGapStudy_CTxKOxST in a similar manner.

Compose Relations. Complex relations can be composed from simple ones, as shown in the following example.

MappingStudy_KAxSKAxKOxCT: We merge the different mappings for knowledge area, sub-knowledge area, and the common term and define the following relation that returns the name of the knowledge object, maturity level, the executive summary, the tags, and the keywords, grouped by knowledge area, sub-knowledge area and common terms.

$$MappingStudy\_KAxSKAxKOxCT = \xi_{KnowledgeArea,Sub-KnowledgeArea,CommonTerm,MaturityLevel,Name,ExecutiveSummary,Tags,Keywords}(\pi_{KnowledgeArea,Sub-KnowledgeArea,CommonTerm,Name,MaturityLevel,ExecutiveSummary,Tags,Keywords}(MappingStudy\_KAxKO \rtimes MappingStudy\_SKAxKO \rtimes MappingStudy\_CTxKO))$$

ProblemGapStudy_KAxSKAxKOxCTxST: Finally we merge the different problem or gap study relations for knowledge area, sub knowledge area, and the common terms and define the following relation that returns the name of the knowledge object, common term, the executive summary, future work, the tags the keywords, title of the study, problem, borders and open questions grouped by knowledge area, sub-knowledge area and common terms. We specified this relation in the following.

$$ProblemGapStudy\_KAxSKAxKOxCTxST = \xi_{KnowledgeArea,Sub-KnowledgeArea,CommonTerm,Name,ExecutiveSummary,FutureWork,Tags,Keywords,Title,Problem,Borders,OpenQuestions}(\pi_{KnowledgeArea,Sub-KnowledgeArea,CommonTerm,Name,ExecutiveSummary,FutureWork,Tags,Keywords,Title,Problem,Borders,OpenQuestions}(ProblemGapStudy\_KAxKOxST \rtimes ProblemGapStudy\_SKAxKOxST \rtimes ProblemGapStudy\_CTxKOxST))$$

To sum up, we applied the DOOR method for the structured creation of ontology relations to the CBK ontology for implementing the Kitchenham structured research area analysis. First, we defined the top-level relations for mapping studies and problem/gap studies. We formalized these relations, using relational algebra, and we derived further relations from these. In addition, we have shown an example for a composed relation of the previously defined relations. We checked all the relational algebra expressions using the *relational* tool.[7] For future semi-automatic use of the relations and in the light of the technical realization (see Sect. 4), the composition of relations can be left to the users. For example, the Semantic MediaWiki+ allows its users to easily switch the predicates of the relations on and off to generate a result set as required.

4 Realization

Our ontology behind the CBK allows us to specify various queries realizing the relations presented in Sect. 3.4 using the SMW query language (SMW-QL). SMW-QL was introduced as a comfortable means to query the SMW [14, 15]. The SMW+ platform provides an inline syntax to integrate queries into a wiki page and a graphical query builder to support the creation of such queries (see Fig. 4). In the following, some of the queries specified previously in relational algebra will be translated into SMW-QL. We start with a simple query referring to the relation MappingStudy_KAxKO, followed by a complex query referring to the relation ProblemGapStudy_KAxKOxST. The query given in Listing 1.1 is read like this: retrieve all KOs that belong to the

[7] http://galileo.dmi.unict.it/wiki/relational/doku.php

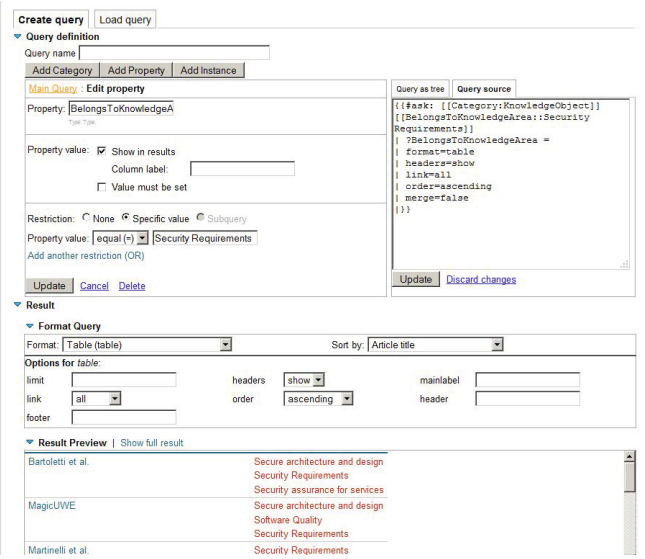

Fig. 4. Mapping study support realized as SMW-QL query

KA Security Requirements and which contain the search term attacker and/or invader in the executive summary. The search term can be further specified using comparator operators and wildcards. The result is returned as a table. Each row represents one knowledge object, whereas each column represents an attribute specified in the query indicated by the question mark. In our case, the table contains the columns RefersToKnowledgeArea, HasMaturityLevel, ExecutiveSummary, Tags and Keywords. The table is sorted along the KAs and MaturityLevel. The user is able to customize sorting by clicking on the table's header. The SMW-QL query given in Listing 1.2 refers to the relation ProblemGapStudy_KAxKOxST, thus supporting a Problem/Gap Study. Therefore it is necessary to additionally output the attributes ProblemDescription and FutureWork from the KO class and the attributes Problem and Border from the Study class.

In contrast to definition of ProblemGapStudy_KAxKOxST in Sect. 3.4, the first SMW-QL query is not reused in this query. While subqueries are in principal possible with SMW-QL, it is recommended to express sub-queries as queries where possible. In this case, it is realized as a flat query, not only because of performance advantages, but also for the sake of simplicity.

```
1 {#ask: [[Category:KnowledgeObject]]
2 [[RefersToKnowledgeArea::Security Requirements]]
3 [[ExecutiveSummary::~*attacker*]] OR [[ExecutiveSummary::~*invader*]]
4 | ?RefersToKnowledgeArea | ?HasMaturityLevel | ?ExecutiveSummary | ?Tags
5 | ?Keywords | sort=RefersToKnowledgeArea,HasMaturityLevel |}
```

Listing 1.1. Query for supporting a Mapping Study

```
{#ask: [[Category:KnowledgeObject]] [[Category:Study]]
[[RefersToKnowledgeArea::Security Requirements]]
[[ExecutiveSummary::~*attacker*]] OR [[ExecutiveSummary::~*invader*]]
| ?RefersToKnowledgeArea | ?HasMaturityLevel | ?ExecutiveSummary | ?FutureWork
| ?Problem | ?Title | ?Borders | ?OpenQuestions | ?Tags | ?Keywords
| sort=RefersToKnowledgeArea,HasMaturityLevel |}
```

Listing 1.2. Query for supporting a Problem/Gap Study

5 Related Work

Tools for structured literature reviews that are regularly used by the software engineering research community are major search engines and digital libraries such as ACM, CiteSeer, IEEE Xplore, Google Scholar, Science Direct and Web of Science [11]. All of these work similarly by specifying boolean search expressions. While they differ in evaluating search expressions and ranking the results, it can be stated that none of these search engines and digital libraries was created to support structured literature reviews [6], as our dedicated approach does.

6 Conclusion and Outlook

We have formalized the Kitchenham approach for structured literature reviews in relational algebra. Furthermore, we implemented these queries in an SMW+ ontology. Thus, we provide a semi-automatic support for the Kitchenham approach that eases the burden of manual literature reviews.

Our approach offers the following main benefits:

- Systematic execution of mapping and problem/gap studies according to Kitchenham based upon ontologies for specific domains (here: secure software and service engineering)
- A structured approach to analyze a research area
- Improving the outcome of literature studies via structured processing of knowledge using ontologies
- Further analysis of research domains can be executed with little effort

Our approach has the limitation that it cannot detect research gaps that are not part of the content of the CKB. Hence, the quality of the outcome of our work is dependent on the quality and quantity of CBK content. However, the possibility also exists that publications might be overlooked when manually executing a literature review according to Kitchenham. Moreover, research gaps, ideas for future work, etc. that only exist in the heads of researchers also cannot be found by any of these approaches.

The work presented here will be extended to support further, more extensive research questions in the future. Examples are the refinement of new findings or the strengthening of own results. We will also work on further automating our approach. We envision an extension of the approach towards other existing ontologies.

References

1. Kitchenham, B.: Procedures for performing systematic reviews. Technical report, Keele University and NICTA (2004)
2. Kitchenham, B., Charters, S.: Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report (2007)
3. Kitchenham, B.A., Brereton, P., Turner, M., Niazi, M., Linkman, S.G., Pretorius, R., Budgen, D.: Refining the systematic literature review process - two participant-observer case studies. Empirical Software Engineering 15(6), 618–653 (2010)
4. Kitchenham, B., Pretorius, R., Budgen, D., Brereton, P., Turner, M., Niazi, M., Linkman, S.G.: Systematic literature reviews in software engineering - a tertiary study. Information & Software Technology 52(8), 792–805 (2010)
5. Kitchenham, B.A., Budgen, D., Brereton, O.P.: Using mapping studies as the basis for further research - a participant-observer case study. Information & Software Technology 53(6), 638–651 (2011)
6. Brereton, P., Kitchenham, B.A., Budgen, D., Turner, M., Khalil, M.: Lessons from applying the systematic literature review process within the software engineering domain. Journal of Systems and Software 80(4), 571–583 (2007)
7. Budgen, D., Turner, M., Brereton, P., Kitchenham, B.: Using Mapping Studies in Software Engineering. In: Proceedings of PPIG 2008, pp. 195–204. Lancaster University (2008)
8. Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic Mapping Studies in Software engineering. In: EASE 2008: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (2008)
9. Codd, E.F.: A relational model of data for large shared data banks. Commun. ACM 13(6), 377–387 (1970)
10. Allocca, C., d'Aquin, M., Motta, E.: DOOR - towards a formalization of ontology relations. In: Proceedings of the International Conference on Knowledge Engineering and Ontology Development (KEOD), pp. 13–20. INSTICC Press (2009)
11. Turner, M.: Digital libraries and search engines for software engineering research: An overview. Technical report, Keele University (2010)
12. Klug, A.: Equivalence of relational algebra and relational calculus query languages having aggregate functions. J. ACM 29(3), 699–717 (1982)
13. Agrawal, R.: Alpha: an extension of relational algebra to express a class of recursive queries. IEEE Transactions on Software Engineering 14(7), 879–885 (1988)
14. Krötzsch, M., Vrandecic, D., Völkel, M., Haller, H., Studer, R.: Semantic wikipedia. J. Web Sem. 5(4), 251–261 (2007)
15. Bao, J., Li Ding, J.A.H.: Knowledge representation and query in semantic mediawiki: A formal study. Technical report, Computer Science Department at RPI (2008)