TOWARDS A COMMON BODY OF KNOWLEDGE FOR ENGINEERING SECURE SOFTWARE AND SERVICES*

Widura Schwittek, Holger Schmidt, Stefan Eicker and Maritta Heisel
paluno: The Ruhr Institute for Software Technology, University of Duisburg-Essen, Duisburg and Essen, Germany

Keywords: Common body of knowledge, Knowledge management, Software engineering, Security engineering, Services computing, Interdisciplinary.

Abstract: *Interdisciplinary* communities involve people and knowledge from different disciplines in addressing a common challenge. Differing perspectives, processes, methods, tools, vocabularies, and standards are problems that arise in this context. We present an approach to support bringing together disciplines based on a *common body of knowledge* (CBK), in which knowledge from different disciplines is collected, integrated, and structured. The novelty of our approach is twofold: first, it introduces a CBK ontology, which allows one to semantically enrich contents in order to be able to query the CBK in a more elaborate way afterwards. Second, it heavily relies on user participation in building up a CBK, making use of the Semantic MediaWiki as a platform to support collaborative writing. The CBK ontology is backed by a *conceptual framework*, consisting of concepts to structure the knowledge, to provide access options to it, and to build up a *common terminology*. To ensure a high quality of the provided contents and to sustain the community's commitment, we further present organizational means as part of our approach. We demonstrate our work using the example of a *Network of Excellence EU project*, which aims at bringing together researchers and practitioners from services computing, security and software engineering.

1 INTRODUCTION

Software engineering (SE) can be considered as an "umbrella discipline": typical SE tasks involve interdisciplinary *knowledge* about processes, methods, tools, and standards. Consequently, new types of SE sub-disciplines have emerged bringing together SE and other disciplines. For example, the field of *security engineering*, which "is about building systems to remain dependable in the face of malice, error, or mischance" (Anderson, 2001), has been combined with SE, and is referred to as *secure software engineering*.

Bringing together different disciplines harbors a number of problems, such as bringing together differing perspectives, vocabularies, and approaches. Moreover, these problems have to be considered with respect to multiple dimensions such as research and practice, which further complicates the situation.

We present in this paper an approach to overcome the aforementioned problems based on a *common body of knowledge (CBK)*. While existing *bodies of knowledge (BOKs)* like the *Software Engineering Body of Knowledge (SWEBOK)* (Bourque and Dupuis, 2005) solely rely on books or hypertext systems as a medium, our CBK provides several advantages such as improved flexibility and access possibilities for its users. In fact, the CBK introduces an *ontology* that allows users to semantically enrich content. The CBK ontology is backed by a *conceptual framework* consisting of three main pillars: The *structuring* of knowledge from different disciplines that the CBK collects and integrates, such as specific tools, methods, and notations constitutes the first pillar. To consolidate an interdisciplinary community, we need a common understanding of the key concepts as well as a common vocabulary of the different disciplines. The CBK introduces a *common terminology*, i.e., necessary basic notions and relations between them. The common terminology is the second pillar, and it allows us to create a mapping between discipline-specific terminology and the notions of the common terminology. The last pillar comprises means to *group* knowledge in order to provide a variety of access options to the knowledge for a wide

*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).

Schwittek W., Schmidt H., Eicker S. and Heisel M..
TOWARDS A COMMON BODY OF KNOWLEDGE FOR ENGINEERING SECURE SOFTWARE AND SERVICES.
DOI: 10.5220/0003666303690374
In *Proceedings of the International Conference on Knowledge Management and Information Sharing* (KMIS-2011), pages 369-374
ISBN: 978-989-8425-81-2

range of different target groups. Since the CBK's content is semantically enriched, it can be precisely queried in order to, e.g., find appropriate methods to analyze and solve a given problem. Moreover, the CBK can be used to contribute to identifying possible research gaps, weaknesses, and interesting directions for future research.

Another difference to existing BOKs is that our CBK heavily relies on *user participation* supported by the wiki platform Semantic MediaWiki[1] with additional extensions (SMW+). Consequently, the CBK supports *collaborative writing* and provides mechanisms to build up and update the CBK. Since the CBK will be opened for the public, our approach is complemented by organizational means considering aspects such as quality assurance to ensure a high quality of content.

We demonstrate our work using the example of the EU project *Network of Excellence (NoE) on Engineering Secure Future Internet Software Services and Systems (NESSoS)*[2], which aims at bringing together researchers and practitioners from security engineering, service computing, and SE. One of the main goals of the NESSoS project is to create a long-lasting research community on engineering secure software services and systems. Our approach for creating a CBK presented in this paper is our contribution to this goal.

The paper is organized as follows: we briefly present our case study, the NESSoS EU project, and outline use cases for the CBK in Sect. 2. In Sect. 3, we present the concepts and an ontology underlying our CBK. Section 4 introduces organizational measures of the CBK. We present related work in Sect. 5. Finally, we conclude and raise ideas for future work in Sect. 6.

2 SCOPE AND FUNCTIONALITY

The major goal of NESSoS is to lay the foundation for a long-lasting research community on engineering secure software-based Future Internet services and systems within a funding period of 42 months. Thus, partners from different fields coming from both academia and industry are re-addressing, harmonizing and integrating research activities. This interdisciplinary and international research setting has a high demand in transferring knowledge from research into practice and triggering research from practical challenges. An impact on training and education activities in Europe is expected as well. Within this overall effort of building a long-lasting research community the CBK plays an integral part. It supports the community to integrate and structure overlapping knowledge areas (e.g. SE, security engineering, services computing). Having in mind that the CBK should serve as a flexible computer-based handbook, we identified several roles and use cases in order to sketch the functionality and the scope of the CBK. We identified the researcher, practitioner, administrator and quality agent as typical roles. Each role has different aims when using the CBK, which have to be considered when defining the use cases. We finally came up with two groups of use cases. In the first group, use cases are defined concerning the management of contents such as adding and editing. In the second group, use cases define different views on the same contents for different target groups and purposes. Two examples for the second group of use cases are "Overview of a specific knowledge area" and "Comparison of different knowledge entities of the same type". Based on the uses cases sketched in this section, we have identified four key concepts for a CBK, which we present in the next section.

3 KNOWLEDGE BASE STRUCTURE

The basic idea behind the structural concept of the CBK is to be able to link arbitrary content classes with each other and to allow users to browse content along the links. Furthermore, the aim is to provide several access possibilities to the CBK, each customized to the target audiences and use cases the CBK addresses. In the following, we introduce a *conceptual framework* that consists of four basic concepts: *knowledge objects*, *knowledge areas*, *learning trails*, and the *common terminology*. All these concepts can be considered as the building blocks of the CBK.

We formalize these CBK concepts using a special CBK ontology. *Ontologies* are used to capture knowledge about some domain of interest. We use the *OWL (Web Ontology Language)*[3] terminology in the following. An ontology describes *concepts* and *relations* between them. In OWL, a concept is specified in terms of a *class*, i.e., a *set of individuals*. An individual represents a concrete object in the domain in which we are interested. In general, in OWL a relation is specified as a *property*, which represents a binary relation between individuals.

We partially present the current ontology under-

[1] http://www.semantic-mediawiki.org

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

[3] http://www.w3.org/TR/owl2-overview/

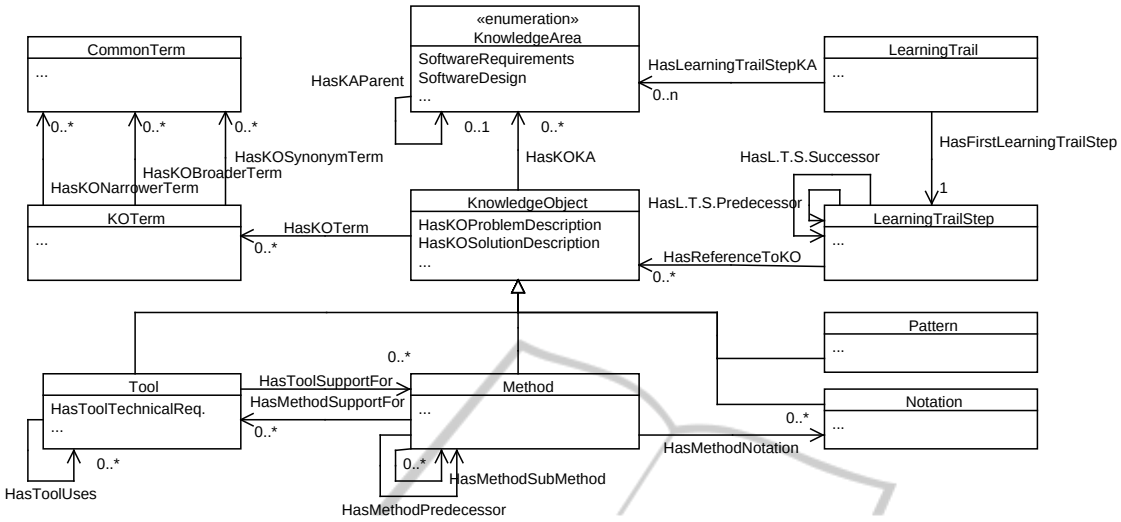

Figure 1: CBK ontology (excerpt).

lying the NESSoS CBK in Fig. 1. We describe the CBK concepts as well as their representation as part of the CBK ontology in more detail in the following subsections.

3.1 Knowledge Objects

A knowledge object (KO) is a fundamental entity of the CBK. The content of a KO is structured around the problem and solution description (see exemplary properties of KO class in Fig. 1) based on the pattern approach prevalent in SE, such as design and architectural patterns). Each KO can be linked to other KOs, resulting in a network of KOs, which as a whole can be considered as a representation of a body of knowledge of a certain discipline. For the initial version of the CBK, we derived four KO types, which we consider as typical types of contributions to a body of knowledge of an engineering discipline, engineering secure software and services in particular. These KO types are *methods*, *tools*, *patterns*, and *notations* (see Fig. 1). We consider them as a starting point, open for extensions in the future. Methods define a set of activities, which in combination with a notation or a number of notations are used to tackle problems in engineering secure software and services in a systematic way. Tools support a software engineer in achieving a development goal in an (at least partially) automated way. Patterns provide a form through which knowledge about recurring development tasks is codified. A notation defines symbols, a syntax, and semantics to express relevant artifacts.

3.2 Knowledge Areas

We adopt the concept of knowledge areas (KA) from the SWEBOK (Bourque and Dupuis, 2005) for our CBK. KAs span the research field as a whole, dividing it into smaller parts and providing an easier access to subjects of interest. The SWEBOK was created in a long process from 1998 to 2003, involving approximately 500 reviewers from 42 countries in a first phase and over 120 reviewers from 21 countries in a second phase. One main result is the worldwide accepted common understanding of what is today viewed as SE. This includes the differentiation of the field into a number of KAs on which we want to base our KAs, e.g., *software requirements* and *software design* (see exemplary properties of KA class in Fig. 1). We took this decision because we regard the field of engineering secure software and services as a supplement of SE and therefore concerning all SE KAs. In addition, we introduce KAs specific to the fields of security and services based on standard literature. For example, we introduce the KAs *risk analysis* and *privacy* as presented in Anderson's *Security Engineering* book (Anderson, 2001).

Each KA consists of a description providing an overview of the KA and its scope, as well as relationships to other KAs. KAs are detailed further into subareas, topics and sub-topics. Each topic or sub-topic contains the following three items: A short state-of-the-art description of the topic/sub-topic, links to KOs supporting the topic/sub-topic and a list of the most relevant publications for further reading.

3.3 Learning Trails

Learning trails are a structuring element meant to provide access to the common body of knowledge on engineering secure software and services for different target groups. This idea is based on the fact that content has to be prepared in accordance with the background of the reader. An expert in a topic area expects more detailed information, whereas a non-expert needs more contextual information in order to be able to understand. Learning trails are therefore written and categorized along different expert levels indicating, e.g., what prior knowledge is required to understand the content. Another differentiation is made concerning the reader's background: if (s)he is from research or from practice. Learning trails are realized by moderated tours, which guide the reader through a set of KOs, which are considered to be part of a certain topic. Each step builds upon the previous step and gives a successive introduction into a topic with respect to the reader's expert level and background (see classes *LearningTrail* and *LearningTrailStep* in Fig. 1). The overall aim of this approach is to provide access to the CBK for a broad spectrum of people, regardless of whether the reader is a student, an experienced expert, a practitioner, or a researcher.

3.4 Common Terminology

The aim of a common terminology is to enable a community to speak the same language; or at least to simplify the translation of a term to another domain with the help of a common reference or *common term* as we want to call it in the following (see Fig. 1). A common term is a term with a meaning on which an agreement was reached within the community. With the common terminology, we therefore introduce an instrument for defining a common term with a certain meaning and for relating different terms with the same or a similar meaning to this common term. In the opposite direction, the common terminology serves the purpose of a dictionary from which synonyms and translations can be queried. A term does not always have the same exact meaning of another similar term, so that deviations to the meaning of the common term must be made explicit. In the CBK, this is realized by three different relationship types. A term's meaning is either synonymical, broader or narrower in relation to another term's meaning (see relationships between *CommonTerm* and *KOTerm* in Fig. 1).

The core CBK team initially creates an ontology of terms of the domain "engineering secure software and services" on basis of the existing CBK content and term usage after a certain period of time. It is then proposed to the community and refined within regular feedback cycles.

4 ORGANIZATIONAL MEASURES

Formulating a body of knowledge for a new discipline is not a task which is accomplished by an individual. It is a highly collaborative effort with many people involved comprising many activities, such as having discussions about what the core of the discipline is, what common terminology to agree upon, and what the state-of-the-art is constituted by, to name just a few. It should also be realized collaboratively, because codifying the knowledge into words and sentences or at least referencing existing knowledge like books and papers means a lot of work. Since the work is never finished, regarding of all new research results contributing to the body of knowledge every day, collaboration is the only feasible way to keep the CBK up-to-date. We acknowledge this by choosing a collaborative approach backed by SMW+ to build up a CBK for engineering secure software and services relying explicitly on user participation.

A CBK has the greatest benefit, if it is complete, up-to-date, and valid. Especially in the beginning of such a project this is not the case, leading to low acceptance and low user participation, if launched for the public too early. We therefore conceived three phases, each with a different focus and participation style in order to work against this effect. Furthermore, the CBK content has to be revised on a regular basis to ensure a high quality, which can be summed up by the question: How is content provisioning and quality assurance supported best while relying on user participation?

We present the three phases in Sect. 4.1, and we introduce quality assurance means in Sect. 4.2.

4.1 Three Phases

The first phase is a *planning phase*, in which all discussed aspects are considered while preparing the initial CBK structure and planning.

During an *inception phase*, content is provided by a closed user group, consisting of experts from different areas within the secure software development field. These experts are mainly researchers from NESSoS, where we profit from the opportunity of having so many researchers linked together through the NoE. The writing process is managed by a central coordinator, who creates the initial CBK structure, defines clear writing responsibilities, watches deadlines,

and ensures quality (see Sect. 4.2). At the end of this phase, the result is a sound CBK content base providing a complete, up-to-date, and validated state-of-the-art of this interdisciplinary research field. A high benefit of this work is expected for researchers from service, security, and software engineering. But also practitioners will find it interesting to get a glimpse on what current research has to offer.

The *run phase* is marked by the launch of the CBK for the general public in terms of reading and writing. At this point in time, the CBK should provide a complete overview of the research field of secure software development. To launch the CBK with a sound content base, which has mostly been created by the community itself, increases the attraction of the CBK for other people that we considered in the use cases (see Sect. 2). Especially for practitioners and for stakeholders other than researchers, learning trails will guide through the vast amount of research results, with respect to their expert level (see Sect. 3.3).

4.2 Quality Assurance

The SMW+ supports quality assurance tasks in different ways. Authors are notified via e-mail, when other people have modified their KO. In the case of vandalism or wrong information, it is possible to revert the changes back to a previous state, making use of the versioning functionality of SMW+. If provided information is controversial, the system allows users to have discussions for each knowledge object on the same page. If new attributes are introduced into the ontology, it is usually the case that these attributes lack of values for existing individuals. SMW+ provides a mechanism to gather information about missing attribute values and allows us to notify the respective author. Furthermore, SMW+ provides an elaborate access control mechanism, which makes it possible to define groups and assign read and write access rights. We make use of this mechanism in order to introduce roles, each with different access rights for, e.g., KOs, KAs or administrative functions of SMW+.

Depending on the project phase, quality is assured in different ways.

In the *inception phase*, quality is assured by a restrictive access control, allowing only partners of the network to have full access to the CBK. Additionally, a central quality assurance (QA) team will start their work having a regular qualitative review on the contents of the CBK, flagging them with a marker indicating when a KO needs to be revised due to a low content quality. But not only the QA team is able to flag KOs. Everyone is allowed to flag an article if vandalism is detected.

While the inception phase is characterized by a controlled environment through a closed user group, the *run phase* takes a more decentralized and community-driven approach. Since we assume that we will reach a critical mass of users during a short period after going public, content contribution will increase and self-regulation will become realistic. Thus, quality assurance is incrementally shifted over to the user, because the QA task is no more feasible to be exercised by a few experts. Instead, experts will rather be assigned responsibilities along the knowledge areas, taking a more moderating role.

As already mentioned, the underlying SMW+ platform supports both approaches, providing adequate collaboration functionality such as feedback and access control mechanisms.

5 RELATED WORK

The concept of a codified BOK is not new and can be found in many different disciplines. Compared to our CBK, they all differ in how they were created and in how knowledge is codified.

All of the BOKs presented in the following were created top-down. By this we mean that an expert team was formed or authors were chosen to write articles. Our approach comprises a top-down phase, but also a bottom-up phase in which the CBK is opened to the public in terms of reading and writing (see Sect. 4). This is comparable to the shift from the creation of the Encyclopedia Britannica to the creation of Wikipedia, acknowledging the fact that new knowledge is generated very fast and by many people these days.

A BOK mentioned before is the “Software Engineering Body of Knowledge” aka SWEBOK (Bourque and Dupuis, 2005), the most prominent among all other BOKs within the SE discipline. The *Computer Engineering Body of Knowledge (Computing Curricula 2005)* (Div. Auth., 2006) and the *Software Engineering Education Knowledge (SEEK)* (part of (Div. Auth., 2004)) have a special focus on SE education. The *Project Management Body of Knowledge (PMBOK)* (Project Management Institute, 2008) is also well-known and covers project management knowledge in general. In the security field, BOKs do exist with different focuses promoted by both industry and governments such as the *Information Technology Security Essential Body of Knowledge* (U.S. Department of Homeland SecurityOffice of Cybersecurity and Communications National Cyber Security Division, 2008).

A more collaborative approach is taken by the two

BOK projects *Usability BOK*[4] and *Build Security In*[5] by the U.S. Department of Homeland Security, both also fostering user participation to provide content following a bottom-up approach.

All of the BOKs presented so far do not allow to be queried elaborately. Many BOKs only exist as a book with access possibilities given by the table of contents or a key word index, while others also provide a hypertext system, allowing one to browse content along links, such as the online version of the SWEBOK[6], the *IEEE Body of Knowledge on Services Computing*[7] or the *Guide to the Systems Engineering Body of Knowledge (G2SEBoK)*[8]. We go one step further and allow more elaborate queries through our CBK ontology with which we are able to semantically enrich all CBK contents.

The need for defining a common terminology for different interdisciplinary communities led to a large number of publications in this area, e.g., the work by Fabian et al. (Fabian et al., 2010) for SE and security. Similar to our common terminology concept, their approach defines a taxonomy relating fundamental notions across the different disciplines, and they specify to what extent notions from one discipline can be translated into notions of the other discipline. The main difference to our work is that Fabian et al. do not complement their results by further concepts such as KAs, learning trails, etc. to create a CBK.

6 CONCLUSIONS

In this paper, we presented a conceptual approach to support bringing together disciplines based on a CBK. We demonstrated our approach using the NESSoS EU project and the interdisciplinary field of engineering secure software and services.

Our approach comprises the following main contributions to *consolidate interdisciplinary communities*:

- KOs allows users to *structure knowledge* such as best practices and research results according to their type. Provided content is semantically enriched in an automated manner. This allows users to browse, compare, and run complex queries on the CBK.
- The CBK introduces a mechanism to *group knowledge* into KAs. This provides access to the CBK via a hierarchical taxonomy and represents a valuable instrument to discover gaps in practice and research.
- The *common terminology* helps the community to find a common language of the different disciplines, and to define and use translations.
- *Learning trails* provide access to the CBK for a broader audience, practitioners and researchers in particular.
- The *ontology* which underlies the whole CBK supports several representations of the CBK, including known ones like books and hyper texts. It provides the flexibility to create customized representations.
- *User participation* is supported by adequate processes and by the chosen SMW+, which might lead to a more up-to-date, a more comprehensive, and a sustainable CBK.
- The realization of the CBK through SMW+ provides a smart means to allow collaborative creation and editing since SMW+ is fully integrated with the design of the ontology.

In the future, we plan to elaborate more on using the ontology properties for specific use cases such as identifying gaps in research areas. Moreover, we want to include user rating mechanisms in our CBK concept.

[4]http://www.usabilitybok.org
[5]https://buildsecurityin.us-cert.gov
[6]http://www.computer.org/portal/web/swebok
[7]http://www.servicescomputing.tv
[8]http://g2sebok.incose.org

REFERENCES

Anderson, R. (2001). *Security Engineering*. Wiley.

Bourque, P. and Dupuis, R., editors (2005). *SWEBOK – Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society.

Div. Auth. (2004). Software engineering 2004: Curriculum guidelines for undergraduate degree programs in software engineering.

Div. Auth. (2006). *Computing Curricula 2005: The Overview Report: A volume of the Computing Curricula Series*. Computing Curricula Series. Association for Computing Machinery and Association for Information Systems and IEEE Computer Society.

Fabian, B., Gürses, S., Heisel, M., Santen, T., and Schmidt, H. (2010). A comparison of security requirements engineering methods. *Special Issue on Security Requirements Engineering*, 15(1):7–40.

Project Management Institute (2008). *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*. Project Management Institute, 4th edition.

U.S. Department of Homeland SecurityOffice of Cybersecurity and Communications National Cyber Security Division (2008). Information technology (IT) security essential body of knowledge (EBK): A competency and functional framework for it security workforce development.