A Usability Evaluation of the NESSoS Common Body of Knowledge

Kristian Beckers
University of Duisburg-Essen
paluno
Germany
Email: Kristian.Beckers@paluno.uni-due.de

Maritta Heisel
University of Duisburg-Essen
paluno
Germany
Email: Maritta.Heisel@paluno.uni-due.de

Abstract—The common body of knowledge (CBK) of the Network of Excellence on Engineering Secure Future Internet Software Services and Systems (NESSoS) is a ontology that contains knowledge objects (methods, tools, notations, etc.) for secure systems engineering. The CBK is intended to support one of the main goals of the NESSoS NoE, namely to create a long-lasting research community on engineering secure software services and systems and to bring together researchers and practitioners from security engineering, service computing, and software engineering. Hence, the usability of the CBK is of utmost importance to stimulate participations in the effort of collecting and distributing knowledge about secure systems engineering.

This paper is devoted to identifying and ameliorating usability deficiencies in the initial version of the CBK and its current implementation in the SMW+ framework. We report on usability tests that we performed on the initial version of the CBK and the suggestions for improvement that resulted from the usability tests. We also show some exemplary solutions, which we already implemented. We discuss our experiences so that other researchers can benefit from them.

Index Terms—ontologies, research gaps, knowledge management, faceted search

I. INTRODUCTION

Getting an overview of existing engineering methods, tools, techniques, standards, and notations (referred to as Knowledge Objects – KOs) for specific fields (referred to as Knowledge Areas – KAs) is of major importance for software and security engineering researchers, practitioners, and students. Usually they have to rely on literature in digital or physical libraries to gain knowledge about KOs in the field of security engineering. We present our common body of knowledge (CBK\(^1\)) and its current implementation in the SMW+ framework to ease this effort. The CBK is open for browsing for the general public. We also invite everyone to contribute knowledge to the CBK and request a specific account\(^2\).

The ontology underlying the CBK helps to acquire knowledge in a structured way and we provide the following core use cases in our CBK. Users can browse KOs and get an overview of existing work. The browsing can be done for a specific kind of KO, e.g., method or tool, with specific search for names, keywords or KAs that are related to KOs. Our implementation allows the comparing of KOs by selecting two or more specific KOs and showing their attributes next to each other. This provides a quick overview of their keywords, KAs etc. and gets an idea of their differences. We also provide a research gap analysis, which can query the CBK for specific criteria. For example, we can conduct a mapping study that shows the KOs related to a specific KA. We can refine this search further, by searching for specific search terms in the KOs like UML. We explained the research gap analysis in detail in our previous work [1].

The CBK is implemented using the following technology. The Semantic MediaWiki is an extension of the MediaWiki platform, which is based upon the popular Wikipedia encyclopaedia. The Semantic MediaWiki has been funded in part by projects of the EU Framework Programmes (FP6 & FP7), SEKT\(^3\) and ACTIVE\(^4\), and by the project Halo\(^5\). The Halo core extension in turn is an extension to the Semantic MediaWiki featuring more options for configuring the UI. The whole framework consists of the MediaWiki, the SMW extension, and the Halo core extension is called SMW+. In the following, when we speak about SMW+ we will refer to the whole framework consisting of the MediaWiki, the Semantic MediaWiki extension and the Halo core extension.

The CBK was closed to the general public in the past. The reason was that only the project partners of the Network of Excellence on Engineering Secure Future Internet Software Services and Systems (NESSoS) should access the CBK during the development and testing. In addition, the partners provided an initial set of KOs and discussed the structure (meaning the attributes) of KOs, while doing so. Hence, in this period the CBK went through several changes that would have confused an external audience. One of the steps for preparing the opening of the CBK for the general public was an usability analysis. The usability of the CBK implementation must be of a very high quality in order to gain the acceptance and participation of the general public. We assessed and improved the usability of the initial version of the CBK by conducting two usability tests. The first usability test concerned querying the CBK and resulted in a satisfactory level of usability. The

\(^1\)http://www.anness-project.eu/cbk
\(^2\)http://www.anness-cbk.org/index.php/Feedback
\(^3\)http://www.sekt-project.com
\(^4\)http://www.active-project.eu
\(^5\)http://www.projecthalo.com
second usability test concerned the entry of new knowledge objects to the CBK. This test revealed numerous points for improvement. We discuss how these suggestions can be taken into account and present several exemplary solutions that we already implemented.

This paper is organised as follows: we provide an overview of the CBK in Sect. II. We report on our usability study in Sect. III and discuss the results and lessons learned in Sect. VII. Section VIII presents related work and Sect. IX concludes and raises ideas for future work.

II. The NESSoS CBK

Ontologies are used to capture knowledge about a certain domain. 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. 1 as a Unified Modeling Language (UML) class diagram\(^6\), 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\(^7\) (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 ≤enuneration≥ 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.\(^9\)

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 several relations between each other, e.g., that one KO uses another.

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. A Notation has at least one grade of formality (gradesOfFormality) according to the enumeration class GradeOfFormality.

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.

III. Usability Study

To manage that third parties contribute to and use the CBK, its usability is of utmost importance. Practical experience and feedback from the NESSoS partners suggested that the usability of the CBK should be improved. To assess the usability of the initial version of the CBK and to identify parts for improvements of the CBK, we conducted usability tests according to the ISO 9241 [2] standard, the Isometrics method [3], and the Thinking Aloud method [4].

In Sect. IV we evaluated the usability of discovering KOs using only the Isometrics method. We evaluated the usability adding KOs to the CBK using the Isometrics method and the Thinking Aloud method in Sect. V. We show suggestions for improvements for the CBK in Sect. V.

IV. Usability Testing of Knowledge Discovery

The first test covers the Browse KOs use case. We conducted this test with master students participating in a course on the development of secure and safe software at the University of Duisburg-Essen\(^10\) in Germany. An initial challenge for participants is that they had to understand the basic principles of the ontologies (e.g. relations between KAs and KOs). They had to describe the basic structure of the CBK. The participants had to learn about this structure of the CBK only by using the system. In addition, participants had to find the knowledge objects Security Adaptation Contract and VeriFast in the CBK.

\(^{6}\)http://www.uml.org/
\(^{7}\)http://www.informatik.uni-trier.de/~ley/db/
\(^{9}\)In general, enumeration types allow us to pre-define values a property might have.
\(^{10}\)http://www.uni-due.de
In this test they had to describe two specific attributes of the Security Adaptation Contract. These are the fields context and the problem description. They also had to read the entry of VeriFast and describe it. The tasks should be done in-between fifteen and thirty minutes.

We based our evaluation upon the ISO 9241 [2] standard that covers ergonomics of human-computer interaction. The standard describes general ergonomic principles, which apply to the design of dialogues between humans and information systems, namely:

- Suitability for the task
- Suitability for learning
- Suitability for individualization
- Conformity with user expectations
- Self descriptiveness
- Controllability
- Error tolerance

However, the standard lacks a detailed description of how to assess these general ergonomic principles. Hence, we use the Isometrics method [3], which proposes two questionnaires that are compliant to the ISO 9241. The difference between these questionnaires is that one requires a shorter time to fill out and to evaluate. In addition, the Isometrics method contains a manual for evaluating the results of the questionnaires. The results of these questionnaires can be evaluated according to the ISO 9241 general ergonomic principles. The Isometric method is available free of charge for scientific purposes.11

We asked the authors for permission to use the method for the evaluation of the CBK during the NESSoS project, and the authors granted permission.

The Isometrics questionnaires contain 76 individual questions, where each question belongs to a specific ergonomic principle. The participants have to rate the quality of specific characteristics of the principles. The rating ranges from 1 to 5, where 5 is the best grade. The participants can also choose not

11The Isometrics Project: http://www.isometrics.uni-osnabrueck.de
to give a grade and reply that they do not have an opinion on this specific question. The Isometrics method prescribes that questionnaires with more than 20 percent of the answers are that the participant has no opinion have to be excluded from the evaluation. We raised this level to 32 percent, so we had only to eliminate one participant from the evaluation. Hence, the results of the experiment have a threat to validity, because of this increase in allowed no opinion answers. Another threat to the validity of this experiment is that the students might have wanted to impress their teacher. However, the evaluation results are shown in Figs. 2 and 3. The first figure shows that, except for the principles suitability for individualization, the CBK was evaluated with grades of 4 and above. The overall mean of the results is also above 4. Fig. 3 shows additionally that the standard deviation of the participants ranged between 1 and 1.5. We conclude from these results that the usability of the CBK for this use case can be regarded as sufficient.

V. Usability Testing of Adding Knowledge Objects

We conducted a second usability evaluation on the CBK for the use case of entering a knowledge object into the CBK. The test persons were a senior researcher and a PhD student. The persons were already familiar with the concept of the CBK. However, the persons were not familiar with the implementation of the CBK.

The experiments were conducted on a 40 inch screen and the test persons were each trying to enter a method into the CBK. This first stage of the experiment was conducted using the so-called Thinking Aloud [4] method. The method captures participants’ conscious cognitions and emotions that are in relation to their actions during the usability testing. The test person speaks aloud every emotion and cognition during the execution of the test. This method can produce insightful results with only few participants. The examiner documented the spoken words during the usability testing. The use of logging instead of video and sound recordings of the test is called Discount Usability Testing [5]. After the experiment a short interview was conducted, and the test persons answered the long version of the Isometrics method [3]. We present the results of the Isometrics questionnaires in Figs. 4 and 5. The results in Fig. 4 are between 2.5 and around 3.5 except for the category suitability for individualization. The controllability and the conformity of user expectations are the highest values with 4 points. The other values are considerably lower. The standard deviation is between less than 1 and 2 points, depicted in Fig. 5. The Isometrics method prescribes that questionnaires with more than 20 percent of the answers are that the participant has no opinion have to be excluded from the evaluation. We raised this level to 25 percent, otherwise we would have had to eliminate one participant from the evaluation. The deviation was around 1 for the controllability and the conformance with user expectation principles. However, the error tolerance deviation was significantly higher with 2 points.

TABLE I

<table>
<thead>
<tr>
<th>Usability principle</th>
<th>Number of usability issues found</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suitability for the task</td>
<td>37</td>
</tr>
<tr>
<td>Self-descriptiveness</td>
<td>31</td>
</tr>
<tr>
<td>Controllability</td>
<td>7</td>
</tr>
<tr>
<td>Error tolerance</td>
<td>7</td>
</tr>
<tr>
<td>Conformity with user expectations</td>
<td>3</td>
</tr>
</tbody>
</table>

a) Suggestions for Improvements: The Isometrics questionnaires offer the possibility to write a comment of why a specific rating was given, and the questions also suggest the test persons to state examples for these problems. Furthermore, an importance rating between 1 and 5 was given for each of the answers. The Isometrics manual suggests that only suggestions with a rating of 3 or above should be considered as improvement suggestion. We collected all the suggestions.
that matched this principles. Furthermore, we collected all the suggestions that were made during the interview and the Thinking Aloud test. The evaluation of the entire usability tests resulted in 85 suggestions for usability improvements for the CBK. Tab. I and Fig. 6 show the principles that have to be improved. The highest numbers of suggestions are in the criteria suitability for the task and self-descriptiveness. The issues in self-descriptiveness concern often manuals and help texts, which can be added with little effort. However, the issues in the suitability for the task principles concern conceptual problems, and these require a considerably higher effort in addressing. A few issues were also discovered in the controllability, error tolerance, and conformity with user expectations.

We mapped the improvements to several functional areas, shown in Tab. II. The start page area represents the main page of the CBK. The add method area is the entire dialoge add method. The remaining areas refer to specific pages of parts of the add method dialoge. The add method: general is one specific page that is part of the add method dialoge. This area contains e.g. the webpages of KOs. The functional areas are also depicted in Fig. 7.

We show the lists of usability issues in Tabs. III. In this table, we present several mappings from found usability issues to functional areas of the CBK and usability principles. We sorted the list according to functional groups. We explain how we plan to address these in the next section. The description column contains citations from the usability test.

VI. REALISING IMPROVEMENTS

We provide an overview of improvements for the usability of the CBK that we will address. We ordered the received suggestions according to the functional areas used in the previous section.

**start page** We work on providing self-descriptiveness of the CBK. For this purpose, we added a long description of
the CBK’s background and goals of the NESSoS project. This long text can be reached via a About the CBK entry in the main menu. In addition, we provide a brief description of the CBK’s directly on the main page. We cleaned the start page of several entries that are not needed, e.g., the myDashboard button. Furthermore, we will increase the number of available help texts, e.g., the change view button will get an explanation next to it. The login is required for every action in the CBK, and so far the login button is difficult to find. The user is not told on the start page to login. We will improve this via a message on the start screen and a bigger login button on the front page.

**add method: save** The CBK shall allow to save the changes made on a knowledge object at any point in time. So far, this was not well explained and the save function would not work, if the address and email of a KO were not entered. We decided to change this behavior. In addition, a click on the save button changed the view of the user from the screen method adding to the start page. However, the change of this behaviour of the system would require an in-depth change of the code of the SMW+ platform. Thus, we will not implement this improvement at this time.

**add method: help** We started adding numerous help texts to the CBK before executing the usability tests. However, the tests showed that the texts require further improvements and corrections. In addition, the help texts are not everywhere in the system, and we have to strive for a complete coverage of the CBK.

**add method** We are working on a help entry that will contain detailed manuals providing guidance on how to enter knowledge objects into the CBK. The manuals already exists, which include many screenshots for a better understanding, and are part of the Population Plan document (available in the NESSoS Web page). These were also used during the requirements tests. The manuals have to be revised according to the discovered problems during the usability test, e.g., the explanation of a role in a method has to be revised. Moreover, after the results from the usability test, the usage of the CBK will be different from what it is now. Hence, we have to adapt the manuals accordingly. Another problem of the **add method** dialogue is that the participants of the test found it difficult to enter a method. We aim to improve this by reducing the complexity of the entry, meaning to reduce the number of fields in the dialog. For instance, the email and address field caused problems for the test participants. We will remove these entries and allow the entries of steps of a method earlier in the dialog. Moreover, we also increase the number of help texts in this dialog and reduce the number of buttons and texts on the screen that are not directly useful for the task. In addition, we have addressed some malfunctions of the system. For example, in the current version method notations were not displayed on the screen. We also found several improvements that we cannot address. For example, a complaint was that the CBK is a browser-based software and when the browser is closed abruptly all the information is lost. We would have to invest a considerable amount of resources into programming a specific browser extension that can deal with this problem. Hence, we cannot solve this issue.
<table>
<thead>
<tr>
<th>Usability ID</th>
<th>Functional Area</th>
<th>Description</th>
<th>Usability Principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>start page</td>
<td>Is myDashboard needed?</td>
<td>Suitability for the task</td>
</tr>
<tr>
<td>2</td>
<td>start page</td>
<td>Is new page needed?</td>
<td>Suitability for the task</td>
</tr>
<tr>
<td>4</td>
<td>start page</td>
<td>Explicit support for cbk use cases missing</td>
<td>Suitability for the task</td>
</tr>
<tr>
<td>6</td>
<td>start page</td>
<td>Login difficult to find</td>
<td>Self-descriptiveness</td>
</tr>
<tr>
<td>7</td>
<td>start page</td>
<td>Login missing feedback</td>
<td>Self-descriptiveness</td>
</tr>
<tr>
<td>8</td>
<td>start page</td>
<td>login failure message did not explain the cause of the error</td>
<td>Error tolerance</td>
</tr>
<tr>
<td>14</td>
<td>start page</td>
<td>Lots of Text to read after login</td>
<td>suitability for the task</td>
</tr>
<tr>
<td>15</td>
<td>add method: save</td>
<td>The save function won’t work until an email is added</td>
<td>Suitability for the task</td>
</tr>
<tr>
<td>17</td>
<td>add method: save</td>
<td>“save” functionality did not explain the cause of the error</td>
<td>Error tolerance</td>
</tr>
<tr>
<td>18</td>
<td>add method: save</td>
<td>“save” should always be possible</td>
<td>Error tolerance</td>
</tr>
<tr>
<td>23</td>
<td>add method: help</td>
<td>help text examples partially incomprehensible</td>
<td>Self-descriptiveness</td>
</tr>
<tr>
<td>25</td>
<td>add method: help</td>
<td>The help is not everywhere present</td>
<td>Self-descriptiveness</td>
</tr>
<tr>
<td>26</td>
<td>add method: help</td>
<td>More help texts are needed. More structured textfields and predefined values would be helpful.</td>
<td>suitability for the task</td>
</tr>
<tr>
<td>30</td>
<td>add method</td>
<td>Too much “garbage” on the screen</td>
<td>Suitability for the task</td>
</tr>
<tr>
<td>35</td>
<td>add method</td>
<td>Earlier versions of Knowledge Objects shall be kept</td>
<td>Error tolerance</td>
</tr>
<tr>
<td>38</td>
<td>add method</td>
<td>Notation should offer choices</td>
<td>suitability for the task</td>
</tr>
<tr>
<td>41</td>
<td>add method</td>
<td>Additional information should provide examples or more detailed questions.</td>
<td>Self-descriptiveness</td>
</tr>
<tr>
<td>47</td>
<td>add method</td>
<td>Some input information was not displayed (Method notations).</td>
<td>Error tolerance</td>
</tr>
<tr>
<td>48</td>
<td>add method</td>
<td>It should be explained that the text fields can be increased in size.</td>
<td>Self-descriptiveness</td>
</tr>
<tr>
<td>51</td>
<td>add method</td>
<td>What does fckLR mean?</td>
<td>Self-descriptiveness</td>
</tr>
<tr>
<td>52</td>
<td>add method</td>
<td>Next step does not work.</td>
<td>Controllability</td>
</tr>
<tr>
<td>54</td>
<td>add method</td>
<td>The about the CBK entries should be linked to the help during the method entry.</td>
<td>Self-descriptiveness</td>
</tr>
<tr>
<td>55</td>
<td>add method</td>
<td>What does the common terminology tab mean.</td>
<td>Self-descriptiveness</td>
</tr>
<tr>
<td>57</td>
<td>add method</td>
<td>Steps and activities in a method are not well explained in the tool or the manual.</td>
<td>Self-descriptiveness</td>
</tr>
<tr>
<td>58</td>
<td>add method</td>
<td>A field should be added that explicitly states that the texts here are copied from a paper.</td>
<td>Suitability for the task</td>
</tr>
<tr>
<td>59</td>
<td>add method</td>
<td>Papers shall be referenced in the text from bibtext entries.</td>
<td>Suitability for the task</td>
</tr>
<tr>
<td>60</td>
<td>add method</td>
<td>I could not add a method notation.</td>
<td>Controllability</td>
</tr>
<tr>
<td>62</td>
<td>add method</td>
<td>Relation to other objects shall offer the other objects in the CBK.</td>
<td>Suitability for the task</td>
</tr>
<tr>
<td>67</td>
<td>add method: startpage</td>
<td>Remove AckLR from page button</td>
<td>Suitability for the task</td>
</tr>
<tr>
<td>71</td>
<td>add method: Image Gallery</td>
<td>How can I reference graphics I already added? A help text is missing.</td>
<td>Self-descriptiveness</td>
</tr>
<tr>
<td>80</td>
<td>add method: general</td>
<td>Dialogue for adding a method difficult to find</td>
<td>Self-descriptiveness</td>
</tr>
<tr>
<td>82</td>
<td>add method: create method</td>
<td>Do we need “Facts about method” information on the screen</td>
<td>Suitability for the task</td>
</tr>
<tr>
<td>83</td>
<td>add method: create method</td>
<td>Can we remove “Category form” from the page</td>
<td>Suitability for the task</td>
</tr>
<tr>
<td>84</td>
<td>add method: create method</td>
<td>Do we need “page-state: unprotected”</td>
<td>Suitability for the task</td>
</tr>
</tbody>
</table>

**add method: start page** The start page of the entry contains several unnecessary texts, which we will remove.

**add method: create method** The create method has numerous texts and buttons on the screen that are not relevant for the task. We will remove these in order to simplify the dialog. For example, we had a large “Facts about the method” entry on the page that is not required for this task and will be removed.

**add method: general** The general field queries much information that is not directly relevant. We will discuss of how to change the dialog so that the user can enter the most relevant entries first, and we will also reduce the number of required fields for the entry.

In the following, we show several usability improvements that have been already implemented. We show these in Tables IV, V, VI, VII, and VIII, which state the usability issue in the first column, the usability solution in the second, and a usability ID in the last column. The usability ID refers to the usability ID in Tab. III. We included pictures for some of the already implemented usability solutions.

We grouped the usability issues into solutions for specific issues. Table IV shows solutions regarding unnecessary functionality and Tab. V concerns our improved start menu. Table VI discusses our improved login functionality, and Tab. VII addresses the the save functionality. Table VIII concerns help texts and manuals of the CBK.

Our usability analysis resulted in further solution tables, which we omit in this paper for space reasons.

### VII. Discussion

The usability test presented in this paper was developed and conducted, because we wanted to open our ontology to the general public. The public should receive an easy to use common body of knowledge (CBK) for security engineering. In addition, we plan to ask for contributions. These contributions are vital for sustainability of the CBK. The knowledge...
has to grow inside the CBK and consider new research, as well as previously not considered work. The growing needs to go on until the CBK achieved a critical mass of content. This critical mass of content will lead to people using the CBK and forming an active community. We require this community in order to continue the CBK effort in the future. The usability testing is so important, because when the users of the general public have to invest too much time to understand the CBK, they will most likely not do it. Hence, we will not achieve a critical user or content mass.

Even though we conducted the usability test with only
engineering and came up with a rather complex set of classes and relations. In an initial usability test of the implementation of this ontology, we realised that the resulting input form for KOs in our CBK has grown increasingly complex. This was an issue that never occurred during the meetings about creating the ontology. We can not stress enough how important these early usability tests are to discover and solve these kinds of problems.

We discussed another issue in detail during the usability testing. Creating an body of knowledge and enforcing a certain set of input fields via an ontology limits the freedom of users.
to input content into the CBK. It is very important early on to present benefits for users that are direct result of the ontology. Otherwise it is very challenging to motivate researchers and practitioners to work with the CBK. For example, we provide a functionality for comparing KOs in the CBK, based on the entry fields. These fields were specified using our ontology, e.g., a context field for the KO. The comparator can now display the contexts of several KOs next to each other. This results in a structured comparison of KOs along these fields. User could realise the effects using an ontology has, once we provided these kinds of functionalities. It is important to implement these as early as possible to reduce resistance against using an ontology.

VIII. RELATED WORK

García-Barriocanal et al. [6] evaluate graphical interfaces to edit ontologies. The work resulted in the statement that most editors are usable for simple ontologies. The authors of this work also stated that browsing mechanisms, help texts and visualisation metaphors contribute to a positive usability experience. Our work arrived at a similar conclusion, because a lot of our found usability issues target these topics.

Mace et al. [7] present a tool to support the population of a security ontology. The tool shall fulfil several requirements like knowledge capturing, user guidance, etc. The authors conducted a usability evaluation of the tool in relation to the defined requirement. The evaluation was based upon a structured questioners. This work can complement our own in such a way that we can map these requirements to our ontology and use also an adapted set of questions to evaluate the CBK against these requirements.

Parkin et al. [8] developed an ontology for human behavioural factors in relation to security management. This ontology shall support security managers to deploy security policies effectively. The ontology can consider, e.g., how a password policy may be perceived by the individuals of an organisation. In a subsequent work, Parkin et al [9] improved their tool and conducted further usability testing. In contrast to our work, Parkin et al. aim to include usability factors into a security ontology and not to evaluate the usability of an security ontology.

Pettersson et al. [10] conducted a series of usability tests for privacy-enhanced identity management system as part of the PRIME project. The authors evaluate three alternative UI paradigms. The authors conclude that the design should not only be based on detailed requirements, but also consider usability testing with mock ups. The reason is that the set of requirements can not be considered final in the early stages of development, because further requirements are elicited during testing. We take the lessons learned by Pettersson et al. into account and conducted usability testing with our implemented ontology.

IX. CONCLUSION

In this paper, we presented a usability analysis, and its results, for the Common Body of Knowledge (CBK). We used existing methods for usability evaluation like the thinking aloud method and conducted an evaluation on two different use cases. We evaluated the browsing of knowledge objects as well as the contribution of knowledge objects using the ontology of the NESSoS CBK and its implementation in SMW+. Our work comprises the following main contributions to improve the quality of the Common Body of Knowledge:

- The usability of entering KOs and browsing KOs has been evaluated. Numerous improvements have been suggested.
- We already implemented a great number of the suggested improvements.
- We reported on lessons learned during our usability analysis.

All of the endeavors presented in this work and also our plans for future work have one major goal: The research and practice communities to be addressed and integrated by the NESSoS project should eventually use the CBK. We have to further improve the usability of the CBK and extend its functionality in order to provide incentives for contributors. We also welcome feedback on the CBK and the work presented here and offer a feedback form for this purpose.

ACKNOWLEDGMENT

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. 256998).

REFERENCES


12http://www.nessos-cbk.org/index.php/Feedback