

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 an 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 the common body of knowledge (CBK¹) 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 account².

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), SEKT³ and ACTIVE⁴, and by the project Halo⁵. The Halo core extension in turn is an extension to the Semantic MediaWiki featuring more options for configuring the UI. The whole framework 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 improve 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.nessos-project.eu/cbk>

²<http://www.nessos-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 diagram⁶, 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 DBLP⁷ (`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.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-Essen¹⁰ in Germany. An initial challenge for participants is that these 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*.

⁸The 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.

⁹In general, enumeration types allow us to pre-define values a property might have.

¹⁰<http://www.uni-due.de>

⁶<http://www.uml.org/>

⁷<http://www.informatik.uni-trier.de/~ley/db/>

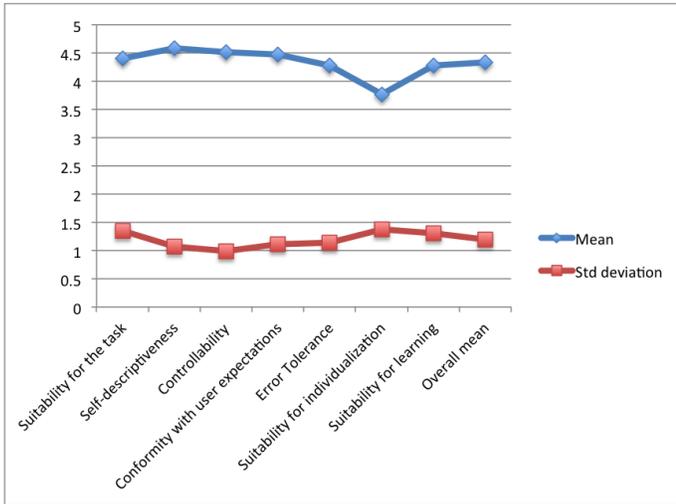


Fig. 3. Results for the usability test for knowledge gathering use case of the CBK (2/2)

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.

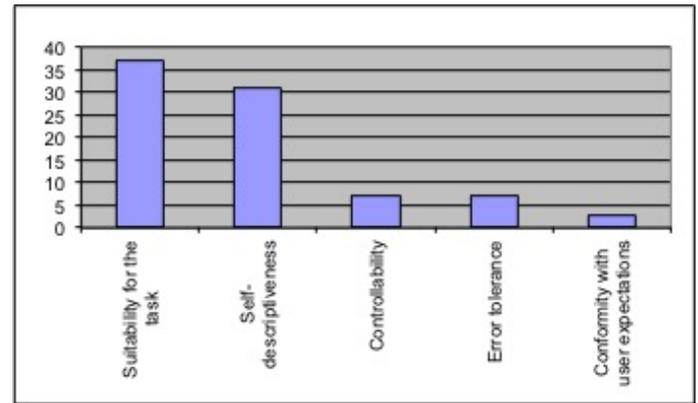


Fig. 6. Diagram of usability principles for which improvements are suggested

TABLE I
LIST OF USABILITY PRINCIPLES FOR WHICH IMPROVEMENTS ARE SUGGESTED

Usability principle	Number of usability issues found
Suitability for the task	37
Self-descriptiveness	31
Controllability	7
Error tolerance	7
Conformity with user expectations	3

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

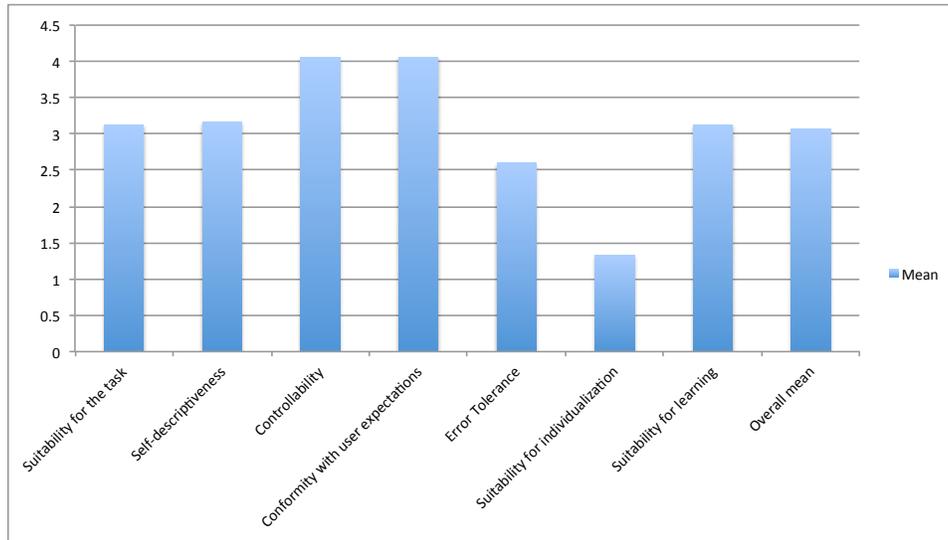


Fig. 4. Results for the usability test for knowledge object entering use case of the CBK (1/2)

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.

TABLE II
FUNCTIONAL AREAS TO WHICH IMPROVEMENTS WERE SUGGESTED.

Functional area	Number of usability issues found
start page	14
add method	37
add method: startpage	2
add method: create method	4
add method: general	9
add method: knowledge area	1
add method: save	7
add method: help	7
add method: Image Gallery	4

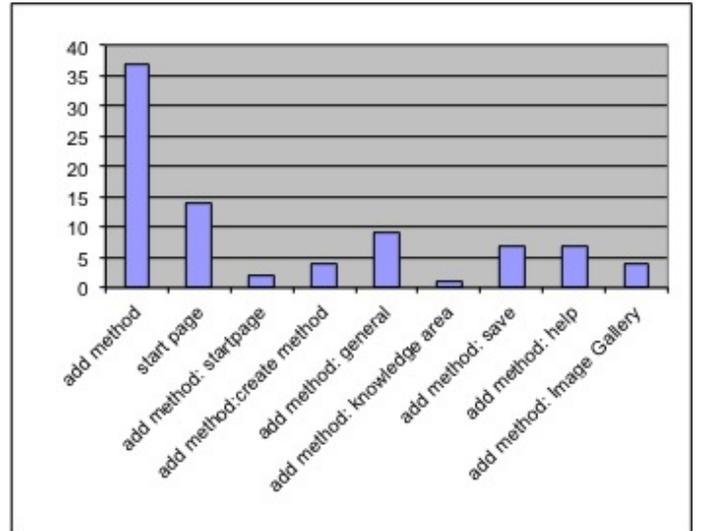


Fig. 7. Diagram of functional areas for which improvements are suggested

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

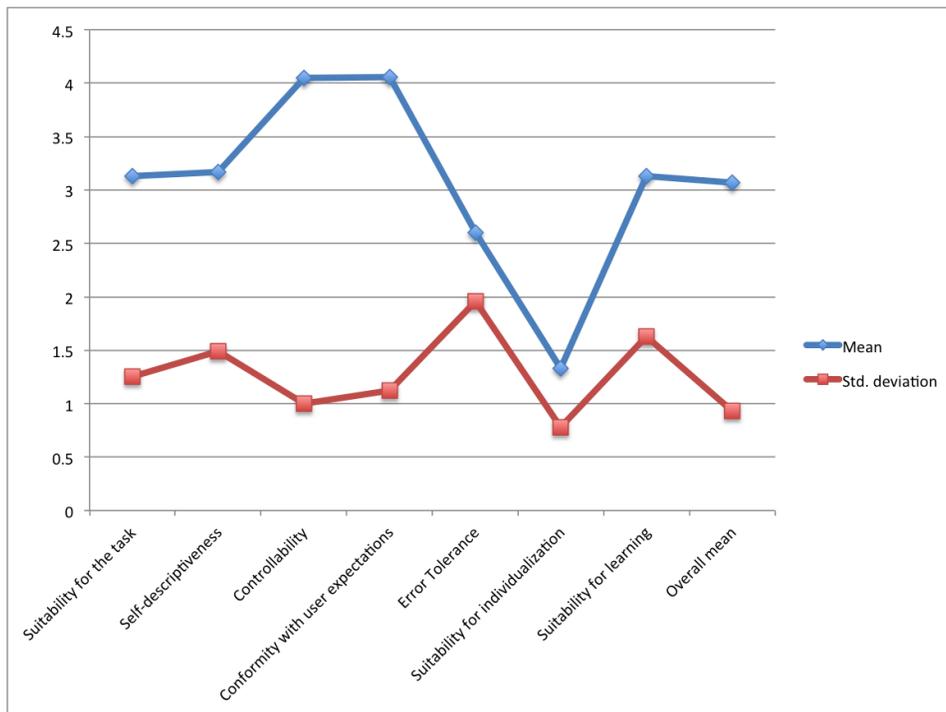


Fig. 5. Results for the usability test for knowledge object entering use case of the CBK (2/2)

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 III
MAPPING USABILITY SUGGESTIONS TO FUNCTIONAL AREAS

Usability ID	Functional Area	Description	Usability Principle
1	start page	Is myDashboard needed?	Suitability for the task
2	start page	Is new page needed?	Suitability for the task
4	start page	Explicit support for cbk use cases missing	Suitability for the task
6	start page	Login difficult to find	Self-descriptiveness
7	start page	Login missing feedback	Self-descriptiveness
8	start page	login failure message did not explain the cause of the error	Error tolerance
14	start page	Lots of Text to read after login	suitability for the task
15	add method: save	The save function won't work until an email is added	Suitability for the task
17	add method: save	"save" functionality did not explain the cause of the error	Error tolerance
18	add method: save	"save" should always be possible	Error tolerance
23	add method: help	help text examples partially incomprehensible	Self-descriptiveness
25	add method: help	The help is not everywhere present	Self-descriptiveness
26	add method: help	More help texts are needed. More structured textfields and predefined values would be helpful.	suitability for the task
30	add method	Too much "garbage" on the screen	Suitability for the task
35	add method	Earlier versions of Knowledge Objects shall be kept	Error tolerance
38	add method	Notation should offer choices	suitability for the task
41	add method	Additional information should provide examples or more detailed questions.	Self-descriptiveness
47	add method	Some input information was not displayed (Method notations).	Error tolerance
48	add method	It should be explained that the text fields can be increased in size.	Self-descriptiveness
51	add method	What does fckLR mean?	Self-descriptiveness
52	add method	Next step does not work.	Controllability
54	add method	The about the CBK entries should be linked to the help during the method entry.	Self-descriptiveness
55	add method	What does the common terminology tab mean.	Self-descriptiveness
57	add method	Steps and activities in a method are not well explained in the tool or the manual.	Self-descriptiveness
58	add method	A field should be added that explicitly states that the texts here are copied from a paper.	suitability for the task
59	add method	Papers shall be referenced in the text from bibtex entries.	Suitability for the task
60	add method	I could not add a method notation.	Controllability
62	add method	Relation to other objects shall offer the other objects in the CBK.	Suitability for the task
67	add method: startpage	Remove AckLR from page button	Suitability for the task
71	add method: Image Gallery	How can I reference graphics I already added? A help text is missing.	Self-descriptiveness
80	add method: general	Dialogue for adding a method difficult to find	Self-descriptiveness
82	add method: create method	Do we need "Facts about method" information on the screen	Suitability for the task
83	add method: create method	Can we remove "Category form" from the page	Suitability for the task
84	add method: create method	Do we need "page-state: unprotected"	Suitability for the task

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

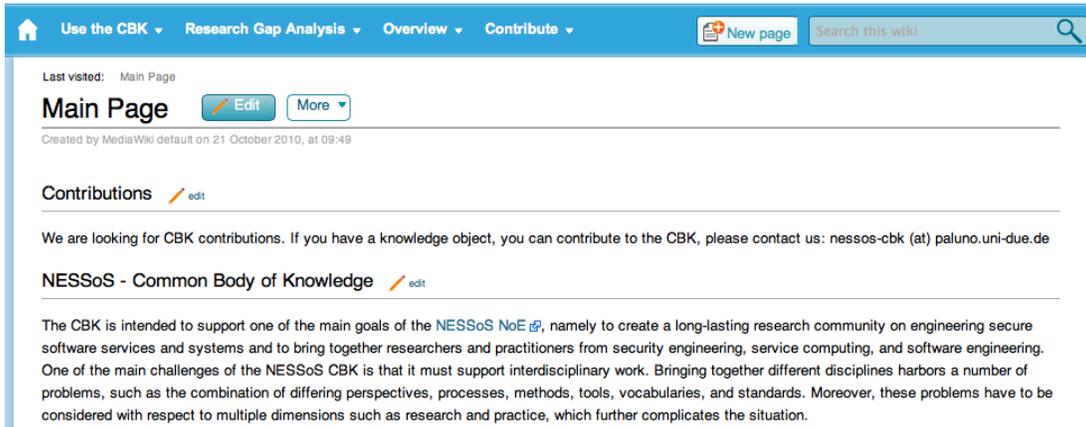


Fig. 8. Disappearing Login Warning Message

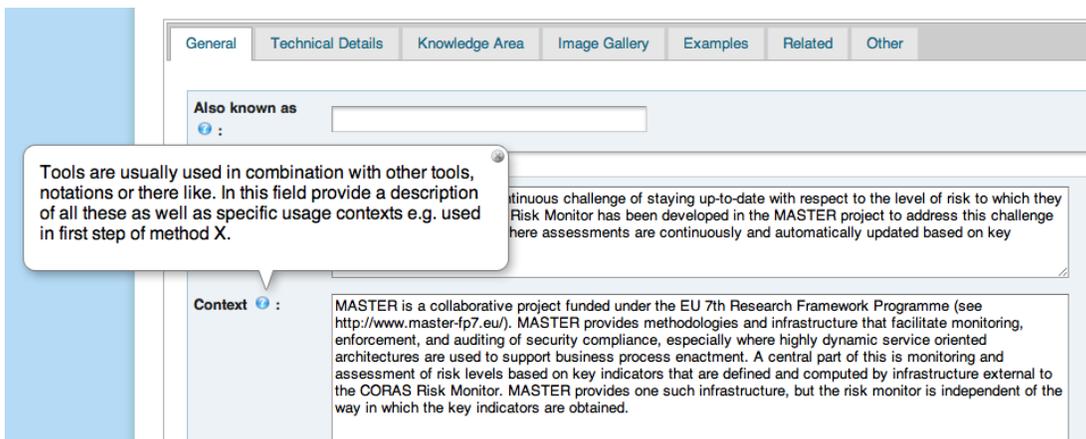


Fig. 11. Improved help message for adding knowledge objects



Fig. 9. Improved Start Menu of the CBK

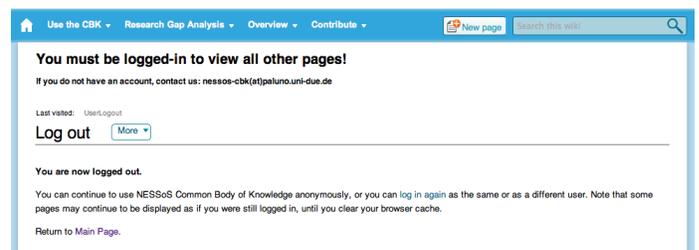


Fig. 10. Login Warning Message

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

TABLE IV
USABILITY SOLUTION 1

Unnecessary functionality		
Usability Issue	Solution	Usability ID
The myDashboard button is not useful for the CBK.	We removed the functionality.	1
The complains message about not entirely supported browsers is not useful.	We removed the functionality.	3
Next step does not work.	We removed the functionality since it has no benefit for the user.	52
Is the red sentence needed?	No, that's why we removed the functionality.	66
Is the message "this is a minor edit" needed?	No, that's why we removed the functionality.	68
Remove moralising text/specific attributes	We removed the text as well as the attributes postal address and email since it is not always clear to assign this information to a knowledge object	73-75
The summary field should be the last that is queried, because I do not know what to summarise	We removed the functionality since it has no benefit for the user.	81

TABLE V
USABILITY SOLUTION 2

Improved Start Menu		
Usability Issue	Solution	Usability ID
Explicit support for the CBK use cases is missing.	We improved the Start Menu to clearly distinguish browse and contribute actions. We added entries for help and using the research gap analysis, see Fig. 9.	4

TABLE VI
USABILITY SOLUTION 3

Improved Login Functionality		
Usability Issue	Solution	Usability ID
The CBK expects a login before using or contributing. The system did not provide any feedback about this issue.	We posted a warning message on every page that states the need for a login, see Fig. 10. This message disappears after a successful login.	6,7,8,9,11
In order to login with a CBK account the user needs to change the domain type field on the login page, because the default value is set to accounts for the JVLR.	We changed the default value.	13
There is long text displayed after the login.	We reduced the amount of text on that page.	14

a limited number of participants, we were able to discover numerous issues. We are planning to conduct further usability tests and want to stress how important usability tests are in the early phases of ontology design. We began to discuss how an ontology should look like for the field of security

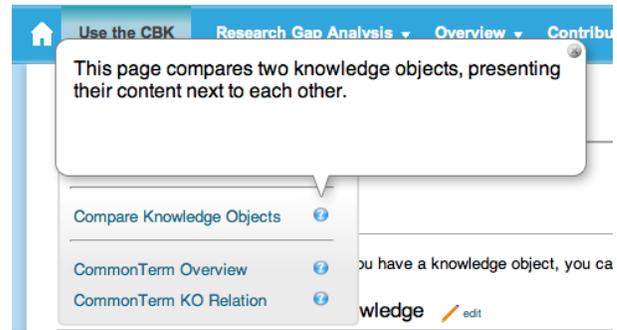


Fig. 12. Improved help message on the start menu

TABLE VII
USABILITY SOLUTION 4

The save functionality always works.		
Usability Issue	Solution	Usability ID
When entering knowledge objects the save functionality did not work until the email and several other fields were entered. The saving functionality also did not respond with a proper warning message, when the problem appeared.	We changed the configuration of the system so that the save functionality is always possible.	15 - 21

TABLE VIII
USABILITY SOLUTION 5

Added and improved help texts and manuals.		
Usability Issue	Solution	Usability ID
Help texts were not present for every field for entering knowledge objects or using the main menu.	We included help texts in all entry fields, and start menu entries, see Fig. ??.	23 - 27, 39, 40, 44, 49/56, 53, 57, 61, 64
The details of the help fields and manuals for entering knowledge objects were not sufficient in detail.	We improved the help messages, see Fig. 12, and our manuals with more details and examples.	29 - 33, 41, 57, 69, 70, 71
Self-descriptiveness: Add a help text that the field size can be changed.	Descriptions regarding the user interface has been added to relevant places.	37, 48, 85

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 purpose¹².

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. 256980).

REFERENCES

- [1] K. Beckers, S. Eicker, S. Faßbender, M. Heisel, H. Schmidt, and W. Schwitek, "Ontology-based identification of research gaps and immature research areas," in *CD-ARES 2012*, ser. LNCS. Springer, 2012, pp. 1–16.
- [2] International Organization for Standardization (ISO), "ISO 9241 - Ergonomic requirements for office work with visual display terminals (VDTs)," 1997.
- [3] G. Gediga and K.-C. Hamborg, "Isometrics: An usability inventory supporting summative and formative evaluation of software systems," in *HCI International*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1999, pp. 1018–1022.
- [4] C. H. Lewis, "Using the "Thinking Aloud" Method In Cognitive Interface Design," IBM, Tech. Rep. RC-9265, 1982.
- [5] I. Harms and W. Schweibenz, "Testing Web Usability," *Information Management & Consulting*, vol. 03, no. 03, pp. 61–66, 2000, <http://usability.is.uni-sb.de/beitrag/testwebu.pdf>.
- [6] E. García-Barriocanal, M. Sicilia, and S. Sánchez-Alonso, "Usability evaluation of ontology editors," *Knowledge Organization*, vol. 32, pp. 1–9, 2005.
- [7] J. C. Mace, S. Parkin, and A. van Moorsel, "A collaborative ontology development tool for information security managers," in *CHI'10*. ACM, 2010, pp. 5:1–5:10.
- [8] S. E. Parkin, A. van Moorsel, and R. Coles, "An information security ontology incorporating human-behavioural implications," in *SIN*. ACM, 2009, pp. 46–55.
- [9] S. Parkin, A. van Moorsel, P. Inglesant, and M. A. Sasse, "A stealth approach to usable security: helping it security managers to identify workable security solutions," in *NSPW*. ACM, 2010, pp. 33–50.
- [10] J. S. Pettersson, S. Fischer-Hübner, N. Danielsson, J. Nilsson, M. Bergmann, S. Clauss, T. Krieglstein, and H. Krasemann, "Making prime usable," in *SOUPS*. ACM, 2005, pp. 53–64.

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