Towards a Common Body of Knowledge on Engineering Secure Software and Services

Experiences from an EU project

Widura Schwittek

paluno – The Institute for Software Technology

University of Duisburg-Essen

SMWCon, Berlin, 22/09/2011
Agenda

- Background: NESSoS
- **Requirements** for a Common Body of Knowledge
- **Concepts** for a Common Body of Knowledge
- **Realization** of the Common Body of Knowledge
- Experience
- Outlook
Background

NESSoS
(NoE on Engineering Secure Software and Services)

- 12 partners from academia and industry
- Total budget: 4.7 mio. € (3.8 mio. € funded by EU)
- Duration: 42 month
- Started in 01/10/2010
Background

- Build a long-lasting research community
  - On Engineering Secure Software and Services
  - Joining existing research communities
    - Security Engineering
    - Software Engineering
    - Service Engineering
    - Formal Methods
Why a Common Body of Knowledge (CBK)?

→ Some challenges of the NoE are:
   - Different communities
   - Different “bodies of knowledge“ (books, papers, mind-sets etc.)
   - Different terminologies

CBK supports building and integrating a joint community
Background

- Software Engineering Body of Knowledge (SWEBOK) → www.swebok.org

- It is an overview of the
  - state-of-the-art
  - state-of-practice

- Create a self-understanding of the discipline

- Basis for curricula creation
Background

CBK goals should go one step further

- Should be created in a collaborative fashion

- Should provide additional knowledge access options
  - More than a TOC, Index, Outline along knowledge areas
  - Show all “security requirements engineering“ „methods“ that use „UML“
Background: NESSoS

Requirements for a Common Body of Knowledge

Concepts for a Common Body of Knowledge

Realization with SMW+

Experience

Outlook
## CBK requirements

<table>
<thead>
<tr>
<th>Feature</th>
<th>DSpace</th>
<th>Liferay</th>
<th>ELGG</th>
<th>SharePoint</th>
<th>MyCoRe</th>
<th>Semantic Wiki</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creation of knowledge object collections for a specific topic and target group</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>o wiki overview page</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>o learning module</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>o other concepts (e.g. groups in social networks etc.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>feedback possibilities for surrogates (meta data of uploaded documents)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>feedback possibilities for wiki pages and forum entries</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>feedback possibilities for any kind of knowledge objects</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rating</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>wiki pages</td>
</tr>
<tr>
<td>comments</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>report abuse</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>editing workflows: free configureable or at least three steps (e.g. draft -&gt; accepted -&gt; published)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>roles/user groups: user, editor, administrator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>role permissions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Repository layer**

Configureable or elaborated meta model                                  |        |         |      |            |        |               |
Management of surrogates for documents like PDF, Doc, ODT, XML, Multimedia (e.g. JPG, AVI) |        |         |      |            |        |               |

...
CBK requirements (excerpt)

1. easy and intuitive user interface
2. collaborative content creation
3. access rights management
4. elaborate search functionality
5. mechanisms to realize SWEBOK concepts (e.g. knowledge areas)
6. mechanisms to built up a common terminology
7. configuration rather than programming
8. adequate licensing options
9. generate a book from the CBK

→ Semantic MediaWiki (+ Halo core ext.)
Background: NESSoS

Requirements for a Common Body of Knowledge

Concepts for a Common Body of Knowledge

Realization with SMW+

Experience

Outlook
CBK concepts (1/2)

- Aligned with the SWEBOK

Knowledge Areas
(e.g. Security Requirements Engineering, Security Design)

Instances/Knowledge Objects
(e.g. secureUML, MagicUWE)

Knowledge Object Types
(e.g. Tools, Methods, Notations)
Published on International Conference on Knowledge Management and Information Sharing (KMIS2011)
- Background: NESSoS
- **Requirements** for a Common Body of Knowledge
- **Concepts** for a Common Body of Knowledge
- **Realization** with SMW+
- Experience
- Outlook
Realization with SMW+

Representing individuals

- Knowledge Object = Ontology
- Individuals = Wiki pages
- Knowledge Object Type = Ontology Classes = Wiki categories
- Template for each Knowledge Object Type (e.g. tool, method)
The CASE tool MagicUWE has been created to support the development of web applications. It focuses on the modeling phase and uses the UML-based Web Engineering (UWE) methodology. UWE provides among others a UML extension (a so called UML profile) based on stereotypes, tagged values and OCL constraints. The tool is built as a plugin for MagicDraw v.16.8. The aim is to augment usability providing additional support in the use of the web specific elements in the design, automatizing certain steps and providing shortcuts.

**Context**

MagicUWE is implemented as a MagicDraw plugin. It was created for Web engineers who want to model secure web applications using the UML-based Web Engineering (UWE) profile and the MagicDraw CASE tool.

**Problem (and motivation)**

Whenever UWE models are created, some tasks have to be repeated over and over, such for example how the navigation menu structure is built. Furthermore, some consistency checks and transformations are very time consuming if executed manually.

**Solution**

The plugin MagicUWE provides features like inserting UWE’s stereotyped elements and copying stereotypes and their tags. Furthermore, MagicUWE supports RIA patterns, transformations between UWE models and a consistency check for secure connection redefinitions in substrates or substrate machines.

**Consequences**

MagicUWE facilitates the modeling of web applications. In particular, it provides specific elements for the modeling of security aspects, such as role based access. The advantage of UWE is to stick to the standard UML, which allows using UML CASE tools such as MagicDraw.
Realization with SMW+

- Creation of semantically enriched Wiki pages (part 1/3)
  - Semantic Forms
  - Tab extension

---

**Executive Summary:**

The CASE tool MagicUWE has been created to support the development of web applications. It focuses on the modelling phase and uses the UML-based Web Engineering (UWE) methodology. UWE provides among others a UML extension (a so called UML profile) based on stereotypes, tagged values and OCL constraints. The tool is built as a plugin for MagicDraw v.16.8. The aim is to augment usability providing additional support in the use of the web specific elements in the design, automatizing certain steps and providing shortcuts.

**Context:**

MagicUWE is implemented as a MagicDraw plugin. It was created for Web engineers who want to model secure web applications using the UML-based Web Engineering (UWE) profile and the MagicDraw CASE tool.
Realization with SMW+

- Creation of semantically enriched Wiki pages (part 2/3)
  - Multiple Instance Template for n-ary relations (e.g. tool x belongs to one or more knowledge areas)
Realization with SMW+

- Creation of semantically enriched Wiki pages (part 3/3)
  - Qualified relations to relate individuals to each other
  - Autocompletion along the ontology
Realization with SMW+

- Overview of all methods
- Semantic Queries
- Up-to-date and sortable

<table>
<thead>
<tr>
<th>Knowledge Object (KO)</th>
<th>KO type</th>
<th>Responsible partner</th>
<th>Review Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>MASTER Design Workbench</td>
<td>Tool</td>
<td>ATOS</td>
<td>1 - Incomplete</td>
</tr>
<tr>
<td>X-CREATE</td>
<td>Tool</td>
<td>CNR</td>
<td>1 - Incomplete</td>
</tr>
<tr>
<td>Jalepa</td>
<td>Tool</td>
<td>CNR</td>
<td>1 - Incomplete</td>
</tr>
<tr>
<td>WS-TAXI</td>
<td>Tool</td>
<td>CNR</td>
<td>1 - Incomplete</td>
</tr>
<tr>
<td>SSG: Smart &amp; Secure GUI Builder</td>
<td>Tool</td>
<td>IMDEA</td>
<td>1 - Incomplete</td>
</tr>
<tr>
<td>UMLsec</td>
<td>Notation</td>
<td>IMDEA</td>
<td>1 - Incomplete</td>
</tr>
<tr>
<td>SecureUML</td>
<td>Notation</td>
<td>IMDEA</td>
<td>2 - Complete</td>
</tr>
<tr>
<td>SECTET</td>
<td>Notation</td>
<td>IMDEA</td>
<td>2 - Complete</td>
</tr>
<tr>
<td>EOS</td>
<td>Tool</td>
<td>IMDEA</td>
<td>1 - Incomplete</td>
</tr>
<tr>
<td>Aspect Oriented Modeling of Component Architectures Using AADL</td>
<td>Technique</td>
<td>INRIA</td>
<td>2 - Complete</td>
</tr>
<tr>
<td>Avantssar Orchestrator</td>
<td>Tool</td>
<td>INRIA</td>
<td>1 - Incomplete</td>
</tr>
<tr>
<td>Acr@r</td>
<td>Method</td>
<td>INRIA</td>
<td>1 - Incomplete</td>
</tr>
<tr>
<td>An Aspect-Oriented and Model-Driven Approach for Managing Dynamic Variability</td>
<td>Technique</td>
<td>INRIA</td>
<td>2 - Complete</td>
</tr>
<tr>
<td>An Expressive Aspect Composition Language for UML State Diagrams</td>
<td>Notation</td>
<td>INRIA</td>
<td>2 - Complete</td>
</tr>
<tr>
<td>Security-driven Model-based Dynamic Adaptation</td>
<td>Technique</td>
<td>INRIA</td>
<td>2 - Complete</td>
</tr>
<tr>
<td>Model-Based Software Design and Adaptation</td>
<td>Technique</td>
<td>INRIA</td>
<td>2 - Complete</td>
</tr>
<tr>
<td>Introducing variability into aspect-oriented modeling approaches</td>
<td>Technique</td>
<td>INRIA</td>
<td>2 - Complete</td>
</tr>
<tr>
<td>Verification of Access Control Requirements in Web Services Choreography</td>
<td>Technique</td>
<td>INRIA</td>
<td>2 - Complete</td>
</tr>
<tr>
<td>Trust Evolution Policies for Security in Collaborative Ad Hoc Applications</td>
<td>Technique</td>
<td>INRIA</td>
<td>2 - Complete</td>
</tr>
<tr>
<td>An approach for model composition and verification</td>
<td>Technique</td>
<td>INRIA</td>
<td>2 - Complete</td>
</tr>
<tr>
<td>An aspect-oriented methodology for designing secure applications</td>
<td>Technique</td>
<td>INRIA</td>
<td>2 - Complete</td>
</tr>
<tr>
<td>Averager stats</td>
<td>Tool</td>
<td>INRIA</td>
<td>1 - Incomplete</td>
</tr>
</tbody>
</table>
Realization with SMW+

- Handbook (LaTeX)
  - Multiple semantic queries
  - Semantic Result Format: Template

---

\section{Knowledge Area: Access Control}

\subsection{Methods}
no contributions yet

\subsection{Notations}
no contributions yet

\subsection{Techniques}

\subsubsection{Security-driven Model-based Dynamic Adaptation}

This paper presents a novel combination of Model-Driven Engineering (MDE), software product lines (SPL) and Aspect-Oriented Modeling (AOM) to support dynamic variability. SPL allows modeling variability and AOM allows the synthesis of an architecture from a choice of variants in the SPL. By composing aspects, it is possible to produce a wide range of configuration models, while managing the combinatorial explosion of variants. Using a MDE approach, they use the architecture to generate the adaptation logic needed to reconfigure the running system, instead of writing it by hand.
Realization with SMW+

Current state: month 12

- Planning (until month 6)
  - Evaluation of platforms, Initial version of structure etc.

- Inception (month 6 – month 24)
  - Closed user group
  - Creation of a sound basis of contents within the NoE
  - Evolution of CBK structure

- Run (from month 24)
  - Opened for public
  - Quality assurance by moderators and user feedback
Background: NESSoS

Requirements for a Common Body of Knowledge

Concepts for a Common Body of Knowledge

Realization with SMW+

Experience

Outlook
Experience

- Evaluation was tedious
  - Situation changed by provided Installer and VM
- Takes time to unlock full potential
- Vibrant community

- Learn from good real-world SMW projects! (e.g. AIFB)
- Besides the technical issues mind the organizational ones!

After one year: SMW+ is still the right decision!
Outlook

Goals

■ Collaborative creation of a Common Body of Knowledge
■ Queryable Common Body of Knowledge
■ Map SWEBOK structure (Knowledge Areas, Knowledge Objects)
■ Handbook generator
■ Community-driven ontology evolution
■ Common Terminology
■ Comparison of user-selected individuals