SMW in Technical Documentation

Mark Schubert & Jonas Wäckerle, parson AG
parson AG

- Software documentation
- Knowledge management
- Wikis and XML-based authoring
- Consulting and training

We create knowledge
Agenda

- About technical documentation
- Structured authoring
- Content management systems
- Writing documentation in wikis
- Documentation with SMW
- Summary, discussion, ideas
## Agenda

- **About technical documentation**
- Structured authoring
- Content management systems
- Writing documentation in wikis
- Documentation with SMW
- Summary, discussion, ideas
Managers and readers

Technical writers create information products

- Documentation for end users:
  Information on how to use the product
- Documentation for service technicians:
  Information on how to install, configure and maintain a product
- Documentation for developers:
  Information on how to develop extensions to a product or modify the software modules
Content quality

Technical writing aims at producing good content

• Relevant to the target group
  – Solve actual problems and tasks
  – Up-to-date

• Accessible
  – Easy to see what content is about
  – Well suited for different types of user
The life of technical writers

This is how we live:

• Driven by frequent release cycles
• Addicted to the input from subject matter experts
• Struggling with vast amount of products and product variants
• Fiddling with different publishing formats
• Reading the minds of the readers
Agenda

About technical documentation

**Structured authoring**

Content management systems

Writing documentation in wikis

Documentation with SMW

Summary, discussion, ideas
Grail quest of technical writing

• Easy and fast to update content
• Integrated workflow with review process
• Reuse content in different variations
• Use the same source for different output formats

Break down content into smaller parts.
Structured authoring

Classification

• Classify content into information types
• Standardize contents according to information types

Topics

• Information types = topic classes
• One topic deals with one subject or question
• Topics are self-contained
• Groups of topics handle broader subjects or processes
DITA information architecture

- Darwin Information Typing Architecture
- Defines topic classes:
  - Concept topic
  - Task topic
  - Reference topic
  - and more
Structured authoring in XML

• Topics are written in XML
• One XML file represents one topic
• Every piece of information is an XML element:
  – Title
  – Paragraph
  – Table
  – Figure
• Topic classes define:
  – Required elements and their sequence
  – Allowed elements
  – Attributes
Clash of the tools

Technical writer‘s tool box

• Everything you can use to generate print, PDF, HTML output
  – Word, PowerPoint, InDesign, FrameMaker, Oxygen, vim…

• XML-based authoring widely used
  – One source to rule them all
  – Output: XML >>> HTML, PDF, WIKI …
  – Separates content from layout
  – Facilitates reuse of content
  – Allows management of content variants for different products
<table>
<thead>
<tr>
<th>Agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td>About technical documentation</td>
</tr>
<tr>
<td>Structured authoring</td>
</tr>
<tr>
<td><strong>Content management systems</strong></td>
</tr>
<tr>
<td>Writing documentation in wikis</td>
</tr>
<tr>
<td>Documentation with SMW</td>
</tr>
<tr>
<td>Summary, discussion, ideas</td>
</tr>
</tbody>
</table>
Content Management Systems

A tool or a bundle of tools supporting technical writers in their basic tasks:

- Writing topics
- Creating and publishing information products
- Managing topics and information products

CMS are systems for authors, not for readers.
Content Management Systems

Writing topics

- XML or WYSIWYG editor
- Comfortable interface for adding tables, figures, and cross references
- Validation of structure and language
- Check-in/ check-out (locking topics)
- Text modules
Creating information products

- Information product: collection of topics
- Add topics or groups of topics (drag & drop)
- Manual or automatic publication
- Individual rules for publishing
### Teamcenter Content Management (KGU-Consulting GmbH)

<table>
<thead>
<tr>
<th>Item</th>
<th>Internal</th>
<th>Training</th>
<th>Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bedienungsanleitung, A.2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Einleitung, A.2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sicherheit, A.2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Allgemeine Hinweise, A.2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sicherheitshinweise, A.2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Leistungsbeschreibung, A.2</td>
<td>❌</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Gerätebeschreibung, A.2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Tätigkeitsbeschreibung, A.2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Installation, A.2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Bedienung, A.2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Entsorgung, A.2</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>Wartung, A.2</td>
<td>✓</td>
<td>❌</td>
<td>✓</td>
</tr>
<tr>
<td>Funktionsweise, A.2</td>
<td>✓</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>Funktion der Maschine, A.2</td>
<td>✓</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>Funktion der Komponenten, A.2</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>Technische Unterlagen, A.2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Content Management Systems

Managing topics and information products

- Metadata
  - Workflow
  - Usage
  - Variant information
  - Languages
- Version management
- Find topics easily
<table>
<thead>
<tr>
<th>Agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td>About technical documentation</td>
</tr>
<tr>
<td>Structured authoring</td>
</tr>
<tr>
<td>Content management systems</td>
</tr>
<tr>
<td><strong>Writing documentation in wikis</strong></td>
</tr>
<tr>
<td>Documentation with SMW</td>
</tr>
<tr>
<td>Summary, discussion, ideas</td>
</tr>
</tbody>
</table>
A wiki is a tool for both, writing and accessing content. Usually, end users and authors do not use the same tool. Some use cases support use of a wiki for documentation:

- Standardized PDF or HTML documentation not required
  - Company-internal documentation
  - Open source software
- Collaborative writing
- Authors accept wiki syntax
- Readers are invited to ask questions or leave comments
- Readers should always read the current version
Writing documentation in wikis

Our SMW project:

• Company internal end-user documentation
• Software manages business and logistics processes
• User access software and SMW instances via intranet
• Context-sensitive linking between software and wiki documentation

Agenda

About technical documentation

Structured authoring

Content management systems

Writing documentation in wikis

Documentation with SMW

Summary, discussion, ideas
Requirements

Author’s view

• Use a content model (information types and structure)
• Workflow
  – Create blank pages to fill them later
  – Assign draft pages to reviewers
  – Follow-up your open tasks
• Flexible modelling of information products
• Reuse content
  – For example, store relations only once
• Use text modules (templates)
Requirements

Reader’s view

• Accessibility: find relevant content
  – Two target groups
• Recognize what the content is about
• Create and print PDF books
Documentation in an SMW instance

Transferred concepts of structured authoring and DITA

- Topic = wiki page
- Topic class = form, templates and category
- General templates and form-specific templates
- Annotations are used to simplify accessibility
Semantic Forms

- One form for each information type (task, concept, reference)
- Metadata fields
  - Generic fields for all information types
  - Form-specific fields
- One text field (concept) or multiple text fields (task, reference)
- Mandatory and optional (text) fields
**Semantic Forms**

---

**Procedure**

- Open `[[Form:Reference]]`.
- Enter the title in the text field.
- Click the "Create" or "Edit" button.
- The reference form opens, see `{{RelatesTo|Reference Form}}`.
- And so on...

---

**Results**

- cannot be blank

---

**Next Steps**

- Check whether the related task topic is linked by means of the Related Topics macro.
- Forward the ID of the reference topic to the responsible software developer for context-sensitive linking.
Annotions (Metadata)

- Input in a (text) field creates annotation
- Generic metadata:
  - Target group
  - Workflow
  - Responsible author
  - Parent topic
  - Sequence number
  - Tags
  - Language
Accessibility (Reader’s view)

- Table of Contents
Accessibility (Reader‘s view)

• Navigation
  – Breadcrumbs
  – Previous page
  – Next page

The Semantic Web

You are here: Manual > General Concepts of Wiki Authoring > The Semantic Web


The Semantic Web is a collaborative movement led by the World Wide Web Consortium (W3C) that promotes common formats for data on the World Wide Web. By encouraging the inclusion of semantic content in web pages, the Semantic Web aims at converting the current web of unstructured documents into a “web of data”. It builds on the W3C’s Resource Description Framework (RDF).

According to the W3C, “The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries.”

The term was coined by Tim Berners-Lee, the inventor of the World Wide Web and director of the World Wide Web Consortium ("W3C"), which oversees the development of proposed Semantic Web standards. He defines the Semantic Web as "a web of data that can be processed directly and indirectly by machines."

While its critics have questioned its feasibility, proponents argue that applications in industry, biology and human sciences research have already proven the validity of the original concept. Scholars have explored the social potential of the semantic web in the business and health sectors, and for social networking.[4] The original 2001 Scientific American article by Berners-Lee described an expected evolution of the existing Web to a Semantic Web, but this has yet to happen. In 2006, Berners-Lee and colleagues stated that: "This simple idea... remains largely unrealized."

TAGS technical writing
Accessibility (Reader’s view)

- Status information
  - Conditional watermark
  - Headline

Prerequisites
- You must be a member of the editors group.
- There is no page in this wiki with the same title.

Procedure
1. Open Form:Reference.
Accessibility (Reader’s view)

- Tags
- Semantic Drilldown
Further scenarios

What we would like to do…but nobody wanted to pay us for:

• Add more semantic power to DITA in SMW
  – Link „Next steps“ and „Prerequisites“
• Reuse more content and evaluate possibilities of conditional text
  – Show templates (text modules) based on metadata
  – For example: Task relates to product > Content is displayed based on product property
• Adding Terminology, checks by using Semantic Glossary (abuse)
• Use properties for context-sensitive help within the software
Agenda

About technical documentation

Structured authoring

Content management systems

Writing documentation in wikis

Documentation with SMW

Summary, discussion, ideas
Limitations, issues, wishes

- MediaWiki
  - Page management (for example, copy a set of pages)
  - Compare (groups of) pages
  - Versioning
  - Template picker
- Semantic Forms
  - Define form input field for a group of forms
  - Embed multiple instance fields in multiple instance fields
  - Allow more than one free text field (sections 😊)
  - „showonselect“: functionality for „hideonselect“?
  - Editor
References

- Images:
  - fotolia.com
  - Screenshots: <oXygen/> xml editor (oxygenxml.com)
  - Screenshots: Teamcenter CMM provided by KGU-consulting GmbH

- DITA Examples: http://docs.oasis-open.org/dita/v1.2/spec/DITA1.2-spec.html