

Analyze This: Extracting Structure from Unstructured Documents

Sarah O'Keefe
Scriptorium Publishing
www.scriptorium.com



Doc FRAME

Agenda

- About the presenter
- Why structured authoring?
- Implementation effort
- Structure analysis



Doc FRAME

About the presenter

- Consultant
- Coauthor, *FrameMaker 7: The Complete Reference*, *FrameMaker for Dummies*, and other books
- Founder, Scriptorium Publishing



Doc FRAME

Why structured authoring?

- Better labeling of content
- Consistency
- Automation
- Support for reuse and content management
- Support for open standards (XML)



Doc FRAME

Implementation effort (a twelve-step process)

1. Identify implementation goals.
2. Define roles.
3. Establish timelines.
4. **Perform structure analysis.**
5. Create structure definition files.
6. Set up legacy conversion process.
7. Set up output paths.
8. Develop documentation.
9. Deliver training.
10. Convert legacy documents.
11. Create change management process.
12. Provide transition support and validate implementation.

Structure analysis

- “Regular” documents have structure.
- Formatting usually representative of underlying structure.
- Need to extract implied structure to create explicit structure.

DocFRAME

Structure analysis

- Analyze existing documents.
- Consider future requirements.
- Develop metadata.
- Develop taxonomy for your content universe.
- Balance precision and simplicity.

DocFRAME

Analyze existing documents

- Gather a set of “representative” documents.
- Classify documents into large groups
 - User guides
 - Help
 - Training manuals
 - ...

DocFRAME

Analyze document groups

- Identify high-level components
 - Preface
 - Chapter
 - Module
 - Topic
- Identify differences
 - Variation or deviation



Doc FRAME

Analyze document groups

- Analyze component relationships.
- Develop flowchart for a specific document type.



Doc FRAME

Book example

- TitlePage Required 1
- TOC Required 1
- Chapter Required 1+
- Appendix Optional 0+
- Index Required 1



Doc FRAME

Translate into structured syntax

```
<!ELEMENT Book (TitlePage, TOC, Chapter+,  
Appendix*, Index) >
```



Doc FRAME

Drill down through elements

- Documents
- Sections
 - Topics, concepts, examples
- Paragraph types
 - paragraphs, list items, terms, definitions, titles



Doc FRAME

Structure analysis example



The Appendix element
The Appendix will be valid in the Book element after all the Chapter elements. Appendix elements will require a Title and an AppendixIntro element. The AppendixIntro will contain at least a Para but also can contain other paragraph-level elements. After the AppendixIntro, an Appendix can contain a series of paragraph-level elements if the appendix is simple) or multiple section elements if the appendix is complex.

Simple appendix definition

```
<!ELEMENT Appendix (Title, Intro,
(Section, Section+)|Para+ >
<!ELEMENT TITLE (#PCDATA) >
<!ELEMENT Intro (Para+) >
<!ELEMENT Section (Title,Para+) >
```



Doc FRAME

Real-life appendix definition

```
<!ELEMENT Appendix (Title,
ContentsList?, Para, (Para | List |
Code | CodeListing | Table | Note |
Warning | Caution | Figure |
Comment)*, (Section, Section+)?) >
```



Doc FRAME

Training sample

1 Creating your first XSL file

At a minimum, an XSL file requires the `<xsl:stylesheet>` element. To be useful, it also needs at least one transformation template. In this chapter, you'll learn how to create basic XSL files and use them.

- Setting up an XSL file
- Creating the root template
- Transforming XML with an XSL file

Training sample

Chapter 1: Creating your first XSL file

Setting up an XSL file

At a minimum, an XSL file requires some header information, a stylesheet element, and a template. In this exercise, you will create a minimal stylesheet file.

To set up an XSL file, follow these steps:

1. Open your text or XML editor and create a blank document. Type in the following:

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/xhtml"
  xmlns:exer="http://www.scriptorium.com/exer"
  xmlns:exer2="http://www.scriptorium.com/exer2"
  xmlns:exer3="http://www.scriptorium.com/exer3" />
<xsl:template match="/" />
<xsl:output />
</xsl:stylesheet>
```
2. Save the new file as `hello.xsl`.

Training example

- Module
 - Title
 - Para(s)
 - ExerciseList
 - Exercise(s)



Doc FRAME

Exercise structure

- Title
- Para(s)
- Task



Doc FRAME

Task structure

- TaskIntro
- TaskList
 - ListItem(s)
 - Code(s)
 - ...



Doc FRAME

Future requirements

- What new elements need to be introduced?
- Can you predict additional requirements?
- Can you design so that new structure can be added later?
- Can you accommodate future plans?
 - Content management
 - Reuse



Doc FRAME

Training material variations

Adding the LandscapeLeft master page

 **On your own: Adding the LandscapeLeft master page**

After the LandscapeLeft master page is applied to the previous task you completed, however, you must the page LandscapeLeft in Step 1 on page 26, and you select the Left master page as the starting point for your page to Step 2 on page 26.

 When you've added the LandscapeLeft master page, save the master page file, and display the body pages (View > Body > Pages). If the Page Layout Warning dialog box is displayed, select the Restore When asked to do this, then the Font menu option.

If you need to see what the LandscapeLeft master page looks like, you can select the Show/Hide icon in the PDF file's Template dialog box.

NOTE: For Windows users, the default path to the installation is C:\Program Files\Scriptorium\Print-Advanced\Content\Frame\Body\Templates\Workshop\PDFFile\Template.docx.

You added the LandscapeLeft master page. In the next variation, you'll apply the First master page to the first page of the template.

New in "On your own" exercises

- No steps
- Notes
- Save icon



Doc FRAME

Exercise vs. ExerciseAdvanced

Exercise

- Title
- Para(s), Note(s),
Save(s)
- Task

ExerciseAdvanced

- Title
- Para(s), Note(s),
Save(s)



Doc FRAME

Variations



More variations

- Book references
- Graphics



Doc FRAME

Develop metadata

- Descriptive information about your
elements
- Unique to each environment
- Critical for finding and identifying
chunks



Doc FRAME

Metadata examples

- Platform: Mac, UNIX, Windows
- Output medium: print, help
- User level: beginner, intermediate, advanced
- Release: 1, 2, 3
- Security level: user, admin, superuser



Doc FRAME

Metadata examples (continued)

- Information type: concept, task, reference
- Product
- Document type: book, help, instructor guide, student guide, quick start



Doc FRAME

Develop taxonomy

- Classification system
- Naming conventions
 - list or List
 - TaskList or List[type="Task"]



Doc FRAME

White papers

- XML overview:
www.scriptorium.com/structure.pdf
- Structure implementation:
www.scriptorium.com/str_implementation.pdf



Doc FRAME

Q & A



Doc FRAME

Contact information

Scriptorium Publishing
Research Triangle Park, NC, USA
www.scriptorium.com
sales@scriptorium.com
919-481-2701 x105



Doc FRAME