

## Integrating Structured FrameMaker and DITA



Sarah O'Keefe  
STC 2006

[www.frame-maker.com](http://www.frame-maker.com)

## Agenda

- About the presenter
- Integration challenges
- Constraining integration requirements
- Gratuitous opinions
- Your questions



[www.frame-maker.com](http://www.frame-maker.com)

## FrameMaker & me

- User since 1994
- Author/coauthor, *FrameMaker for Dummies*, *FrameMaker 7: The Complete Reference*, *FrameMaker 7 Workbook Series*, and "Unstructured to Structured FrameMaker Migration Guide"
- FrameMaker trainer and consultant



[www.frame-maker.com](http://www.frame-maker.com)

## And a few XML items...

- Structure implementation consultant
- Coauthor, Scriptorium white papers:
  - Structured Authoring and XML
  - Managing Implementation of Structured Authoring
  - Integrating FrameMaker and XML



[www.frame-maker.com](http://www.frame-maker.com)

## Handouts?

- Leave me an email address and I'll send you a PDF.



[www.frame-maker.com](http://www.frame-maker.com)

## What is DITA?

- Darwin Information Typing Architecture
- Topic-based XML architecture
- XSL transformation for various outputs
- Originally developed by IBM and used by information developers there
- Designed for extensibility



[www.frame-maker.com](http://www.frame-maker.com)

## State of the union



- FrameMaker v7.2 includes a **starter** application for DITA
- Earlier FrameMaker versions do not have any out-of-the-box support
- The starter application requires additional configuration for a production environment



## DITA/FrameMaker integration is hard.

- FrameMaker document model and DITA document models must be matched up:
  - Links
  - Tables
  - Index entries



## Links

- Structured FrameMaker
  - Links are ID based
  - Link text is derived from link target
  - `<CrossRef Idref="xxx" format="heading"/>`
- DITA
  - Links are based on file names
  - Link text is arbitrary
  - `<xref href="file.xml#xxx">link text</xref>`



## Linking choices

- DITA starter app:
  - Store arbitrary DITA link text in an attribute
  - Use FrameMaker cross-reference formats to create cross-reference text
- Other options:
  - Modify FrameMaker to support DITA xref structure
  - Use FrameMaker hyperlinks instead of cross-references for DITA xrefs?



## import.xsl controls translation

```
<xsl:attribute name="linktext">
<xsl:choose>
  <xsl:when test="linktext">
    <xsl:value-of select="linktext/text()"/>
  </xsl:when>
  ...
</xsl:choose>
</xsl:attribute>
```



## Tables

- One table group (tgroup) per table
- No nested tables
- Similar to DocBook table issues; not a huge concern



## Index entries

### ➤ DITA:

```
<indexterm>chocolate</indexterm>
<indexterm>chocolate
  <indexterm>rabbit</indexterm>
</indexterm>
```

### ➤ FrameMaker:

```
<indexterm>chocolate</indexterm>
<indexterm>chocolate:rabbit</indexterm>
```



## XSL is your friend.

- Framemaker v7.2 provides support for XSL transformation on import and export of XML files.
- Use XSL to map DITA structures to FrameMaker-friendly structure and back.



## And it gets worse

- Some features are implemented completely differently:
  - conref links (closest equivalent: text insets)
  - Topic maps (closest equivalent: books)
  - Specialization (!)
  - Conditional processing



## Conref links

- Points to another element
- Can be almost anything
- Much more flexible than text insets
- <section conref="file.xml#some\_id"></section>
- Could process during import into FrameMaker
- Could create attribute reference in FrameMaker, but automatic expansion would require major effort



## Topic maps

- List of content files, similar to book
- **BUT** can nest topic references:

```
<topicref navtitle="Tasks">
  <topicref href="a.xml" navtitle="title1">
    <topicref href="b.xml" navtitle="title2"/>
  </topicref>
  <topicref href="c.xml" navtitle="title3">
</topicref>
```



## What is specialization?

- Elements have a type (*task* is a topic)
- When you create a new element, you use an existing type (*example* is a topic)
- When creating output, DITA bases default formatting for the new element on the processing for that element type (examples processed like all other topics)
- Makes element addition theoretically easier



## Specialization

- FrameMaker can read a specialized DTD
- But no support for formatting based on specialization
- If DTD is specialized, you must modify FrameMaker configuration to address specialized elements



## Specialization workflow

- Specialize the DITA DTD
- Bring into FrameMaker to create matching FrameMaker EDD
- FrameMaker DITA application uses the default DTD
- Supporting specialized container elements relatively easy; specializations of links and other “object” elements trickier



## Conditional processing

- FrameMaker: conditional text
- DITA: Attribute values and ditaval file to specify what to include and exclude
- Options:
  - Sourcerer provides FrameMaker-based solution for attribute-based show/hide
  - Filter content before import into FrameMaker
  - Custom FrameMaker configuration



## Can you minimize DITA integration requirements?

- Publishing
  - Edit in XML
  - Import to FrameMaker for print/PDF rendering
- Authoring
  - Edit in FrameMaker
  - Export to XML
- Round-tripping
  - Editing on either FrameMaker or XML side



## Publishing

- Relatively easier
- On content import, transform DITA XML into FrameMaker-friendly XML
- Discard content not needed for print rendering
- Resolve/expand conrefs, topic maps, and the like on import
- No need to preserve editable links
- XMetaL recommends this approach



## Authoring

- Author topics in FrameMaker.
- Use a different tool for topic maps or resolving conrefs – or customize FrameMaker (FrameScript, FDK?).
- Export topics from FrameMaker to DITA. Use XSL transformation (or read/write rules) to convert over to correct DITA structure.
- FrameDITA-Lite



## Round-tripping

- Much more difficult than one-way translation
- Must account for all unique constructs on both sides and translate and/or preserve them
- Example: FrameMaker uses a cross-reference format attribute; DITA does not:  

```
<CrossRef Idref="xxx" format="Heading"/>
```
- Where do you store format information in DITA XML?



## My questions

- How many of you are currently using FrameMaker? Structured or unstructured?
- How many of you are currently using DITA?
- Considering DITA?
- Considering XML in general?



## A few opinions

- Configuring FrameMaker for DITA publishing is a reasonable approach.
- Authoring within FrameMaker will require a huge investment to support *all* DITA constructs (conrefs, topic maps). Consider subsetting?
- How badly do you want DITA?
- How badly do you want FrameMaker?



## It all comes down to print (and PDF).

- Do you need complete control over your print/PDF output?
- Is the output generated by DITA's XSL-FO stylesheets not good enough?



## In the land of the endless buffet, let's use food analogies.

- Content analysis is broccoli.
- Creating a content model is like cooking from scratch.
- DITA is the heat-and-serve pizza at the grocery store.
  - What if you have food allergies?
  - What if you want your own toppings?
  - Is whole-wheat crust available?



## Your questions



## Resources

- DITA Open Toolkit:  
<http://sourceforge.net/projects/dita-ot/>
- FrameDITA-Lite:  
[http://www.publishingsmarter.com/pages/develop/framemaker\\_DITA.html](http://www.publishingsmarter.com/pages/develop/framemaker_DITA.html)
- Intech's DITA edition:  
[http://www.intech.com/dita/Frame\\_DITA\\_Edition.php](http://www.intech.com/dita/Frame_DITA_Edition.php)



## Contact information

Sarah O'Keefe  
Scriptorium Publishing  
Research Triangle Park, NC, USA  
[www.scriptorium.com](http://www.scriptorium.com)  
[info@scriptorium.com](mailto:info@scriptorium.com)  
+919-481-2701 x105

