

# Migrating from Unstructured to Structured FrameMaker



Sarah O'Keefe  
Scriptorium Publishing  
www.scriptorium.com

www.scriptorium.com

## Agenda

- Reviewing structure implementation
- The FrameMaker structured application
- Conversion rules tables
- Eternal mysteries
- Resources



2

## About those blank slides...

- No, I didn't forget to fill them in.
- I'm looking for audience participation on these slides.



3

## Why structure?

- Must move to Linux/can't use FM any more
- repurpose content for web/print
- DITA? unified schema
- reduce reliance on proprietary tools (get rid of ePublisher Pro)
- it might increase efficiency?
- future-proof my content
- company-driven...buzz



4

## Why structure?

- automated content generation
- no more reinventing the wheel
- streamline data exchange/localization
- RH>>languages/XML easier
- distributed team
- enforce template
- easier content sharing



5

## Reviewing structure implementation

- Twelve-step process
- Work in FrameMaker begins in step five



6

## 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.



7

## DITA?

- standard
- content reuse?
- topic-based organization
- predefined architecture (no content modeling?)
- What if your content doesn't match?
- role-based content/conditionalization



8

- `<topic customer="a,c" platform="win" output="online">`
- ...
- `</topic>`



9

## What makes FrameMaker a good choice for structure?

- structure view
- print preview/visual editor
- already have it
- lower learning curve?
- really nice PDF/easy to create and format
- authoring/publishing in one package
- going to be around for a while



10

## What makes FrameMaker a good choice for structure?



11

## What are FrameMaker's disadvantages?

- other people don't have it
- print oriented
  - author mindset
  - configuration
- configuration is difficult
- ...conversion of legacy content is horrid
- no specialization support for DITA
- predefined DTD requirement?



12

## Creating an element definition document (EDD)

- Pattern language programming
- EDD is a structured FrameMaker file
- Learning structured authoring while creating an EDD is quite challenging



13

## EDD syntax

- Element specification syntax same as in DTDs
  - ? (question mark) – zero or one
  - \* (asterisk) – zero or many
  - + (plus) – one or many
  - , (comma) – sequence of elements
  - | (pipe) -- Or
  - ( ) (parentheses) – for grouping



14

## EDD creation process

- Elements and attributes first
- Then add formatting



15

## Element types in the EDD

- Containers
- Text range (character tag equivalent)
- Cross-references
- Markers
- Variables
- Graphics
- Table components



16

## Formatting choices in your EDD

- Reference paragraph/character tags
- Format change lists
- Embedded formatting



17

## What is the best formatting approach?

- Downstream, ePublisher Pro requires paragraph/character tags.
- Using an existing formatting template will save development time.
- Format change lists allow inheritance.
- Embedded formatting and format change lists are harder for authors to modify.



18

## What is the best formatting approach for you?



19

## Structured application

- A set of configuration files
- Not a “software application”
- “An application of structure” to a particular problem



20

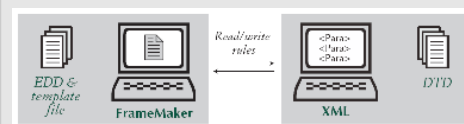
## FrameMaker structured application components

- Element definition document (EDD)
- Formatting template
- Document type definition (DTD) or schema file
- structapps.fm file
- Read/write rules
- Extensible Stylesheet Language (XSL) transformation files
- FrameMaker Developer's Kit (FDK) code



21

## Basic structured application



22

## Real-life structured application



23

## Read/write rules vs. XSL vs. FDK

- Difference is not always clear.
- Read/write rules
  - FrameMaker-specific items, such as table widths or graphic attributes
- XSL
  - Rearranging, reordering, filtering content
- FDK
  - Things that are impossible with read/write rules or XSL



24

## FDK examples

- Automatic assignment of IDs and ID validation across a book.
- Mapping of FrameMaker equation information to MathML
- Extraction of callouts from FrameMaker anchored frame
- Mapping of ID-based cross-references to file-based linking (DITA)



25

## Good news/bad news

- You can do almost anything with FDK development.
- If your project requires FDK work, expect the cost to skyrocket.



26

## Other bits...not technically part of application

- Conversion rules table for unstructured to structured content conversion
- FrameScript scripts for automation of tedious tasks
- Document conversion
  - ePublisher Pro/MIF2GO
  - XSL transformation



27

## FrameScript

- Inexpensive
- Will pay for itself on first project
- No need to learn scripting; pay someone to write the scripts



28

## Should you choose ePublisher Pro or XSL transformation?

- need XSL person/need ePP person
- stability of ePP
- licensing of ePP
- need pretty cross-platform, cross-browser help
- XSL is faster
- ePP is easier to configure/basics



29

## Conversion rules table

- Map unstructured components onto elements
- Finally, a reward for virtuous tagging
- Don't expect 100 percent automation



30

## What about DITA?

- Darwin Information Typing Architecture
- Standard structure for modular documentation
- FrameMaker has support for DITA built into version 8



31

## Should you use DITA?

- 30% maybe, ??
- no content modeling (only not)
- separate format vs. content (all XML)
- OOB authoring tool support
- widely known
- hype
- Easy enough even for technical writers (!)
- What about DocBook?



32

## DITA?

- software support
- topics are nice
- context-sensitive help? (Tony Self)
- multiple outputs
- difficult to implement? (easier than the alternatives??)
- vendor-independent?



33

## The \$64,000 question: How much will migration to structure cost?

- Analyze content
- Research, choosing, configuring, buying software/tools
- Write the business case
- Get buy-in
- (Re-)Training, learning new tools
- Conversion
- Contractors/consultants



34

## Cost

- Shift from not-topic-based to topic-based
- Technical support for configuration
- Additional tools
- Productivity LOSS
- CMS?
- Training  
Personality issues/disruption/change management



35

## Cost

- Turnover in staff
- Change resistance
- Complexity
- Upper management support/executive sponsorship/"eliminating" change resistance
- Software change management
- M-O-N-E-Y



36

## Cost

- Ongoing training
- Transition plan
- Exit strategy ("what if?")
- Software support (vendor commitment)
- 



37

## Cost contributors

- Number of authors
- Document complexity
- Number of documents to be converted
- Degree of customization of structure required
- Number of deliverable types
- Number of supported languages
- Amount of training required



38

## Cost contributors

- Complexity
- Velocity



39

## How much time will migration to structure require?

- Not less than six months.



40

## What are the factors that affect implementation size?



41

## Size factors

- Geography
- In-house expertise and availability
- Resources?
- Who are the stakeholders?



42

## Recommendations

- DITA: Analyze how well DITA fits your content, then make a decision
- FrameMaker for publishing: If you need attractive print/PDF, FrameMaker is the way to go.
- FrameMaker for authoring: If your existing content is in unstructured FrameMaker, structured FrameMaker offers a gentler path into XML-based authoring. Also consider a mixed workflow.



43

## And random thoughts

- Avoid FDK programming if possible to minimize costs.
- Understand FrameMaker's object model and work within that model to simplify development.
- If possible, start from FrameMaker structure rather than from a DTD.



44

## Print challenges

- FrameMaker provides excellent print and PDF support.
- Most other structured workflows do not.
- The trade-off is that FrameMaker has distinct preferences for your content structure.



45

## ePublisher Pro

- Use it if you need a format such as WebWorks Help
- If you have existing ePublisher Pro templates, continue using them for the time being.
- To combine Word/FrameMaker/DITA content, consider it.
- Otherwise, switch to vanilla XSL transformation.



46

## XSL transformation for single sourcing

- For generation of other markup (HTML) from XML content
- Open standard
- Parsers are free and open source
- If you can write scripts, you can write XSL
- Learning curve is significant



47

## XSL-FO

- Extensible Stylesheet Language Formatting Objects
- Would replace FrameMaker for print production
- FrameMaker is *much* easier to use and produces nicer output
- Appropriate for batch processing, on-the-fly generation of PDF, and other specialized uses



48

## Will you switch from FrameMaker to XSL-FO?

- for batch processing...
- for languages not supported in FrameMaker
- for Linux environments...
- PDF is just a check-off



49

## Localization

- Ship XML, not structured FrameMaker, to your localization vendor.
- Build localized templates/structured applications/XSL transformation.
- Unicode now supported in v8.
- No support for right-to-left languages (Hebrew, Arabic).



50

## Learning curve

- It's steep.
- Consider multiple training stages.
- Divide and conquer
  - One person learns EDDs
  - One person learns XSL
- Consultants can help.



51

## Final and most important issue

- Change resistance
- Critical factor in determining success or failure of the entire effort



52

## Resources

- FrameMaker Developer Connection
  - *Structured Application Developer's Guide*
  - Migration white paper
  - [www.adobe.com/devnet/frameMaker/](http://www.adobe.com/devnet/frameMaker/)
- Scriptorium web site ([scriptorium.com](http://scriptorium.com))
  - White papers
  - Structured FrameMaker training
  - *Publishing Fundamentals: FrameMaker 7* (yes, we're working an update)



53

## Resources

- FrameScript, [www.framescript.com](http://www.framescript.com)
- Rick Quatro, [www.frameexpert.com](http://www.frameexpert.com), for custom FrameScript development
- *XSLT Programmer's Reference*, Michael Kay
- FrameMaker Developer's Kit, available from [adobe.com](http://adobe.com)



54

## Contact information

Sarah O'Keefe  
Scriptorium Publishing  
[www.scriptorium.com](http://www.scriptorium.com)  
[okeefe@scriptorium.com](mailto:okeefe@scriptorium.com)  
+1 919 481-2701 x102



63

## Q & A



64