

# Optimizing Your Content for Reuse and Conditional Publishing

Joe Gelb  
President, Suite Solutions





# Discussion Points

- Who is this guy?
- The case for reuse and conditional publishing
- Topic-oriented content architecture
- Levels of content reuse
- Process for migrating towards a topic-oriented documentation architecture



# Who is this guy?

- Background in Mechanical Engineering
  - CAD/CAM
  - Production process planning
- CTO at Live Linx
  - Built and managed doc conversion team
  - Responsible for design, development and implementation of ConteX CMS
- Formed independent consulting and CMS implementation group: Suite Solutions



# What is Reuse?

- Linking to the same content chunk multiple times
  - Graphics, logos
  - Common text: notes, cautions, warnings, common lists of steps, copyright, etc.
  - Common topics: procedures, descriptions, reference information



# What is Conditional Publishing?

- Process for including and excluding content during publishing to a delivery format
  - Include based on context: per product, customer, configuration, etc.
  - Include based on delivery: per output format, audience
  - Variable Text: set values based on context



# Case for Reuse and Conditional Publishing

- Increase consistency of content wherever it is used
- Quicker content development and maintenance
  - Less need to write new content
  - Updating once affects all instances where content is used
- Quicker reconfiguration
  - Content is more modular
  - For new product/customer/configuration, can easily change modules, add new ones, drop irrelevant ones, change order
- Lower translation and DTP costs
- Share content across the organization: training, eLearning, marketing, service



# Introducing: Topic-oriented Content Architecture

- Methodology for modularizing documentation content
- Topic:
  - unit of information with a title and content
  - short enough to be specific to a single subject or answer a single question
  - long enough to make sense on its own and be authored as a unit
- Map:  
Collection of references to topics to form deliverables.  
May be sequential list or a series of relationships



# Introducing: Topic-oriented Content Architecture

- DITA (Darwin Information Typing Architecture):
  - provides a topic-oriented architecture
  - provides a set of XML DTDs for authoring structured technical documentation
  - provides basic tools for publishing (open-source toolkit)
- Other structured formats which may be modularized
  - S1000D
  - Docbook content units
  - Information Mapping



# Traditional Method for Developing Content

- Deliverable-based:
  - Determine deliverables needed for product
  - Create an outline of topics needed
  - Locate and reuse existing content; write everything else from scratch
- Using topic-oriented architecture:
  - Create outline using map
  - Link to existing topics; create space-holder topics for new content
- Maintain Master Topic List



# Levels of Content Reuse

- Topic Level
  - Topics to include or exclude from the publication map
- Component Level
  - Use topic components in multiple topics
- Conditional Text
  - Text or graphics inside topics to show or hide
- Variables
  - Text that changes conditionally based on context



# Determining when content should be included

- Tag the content, then filter
  - Topics
  - Elements inside topics
- Effectivity fields and values = Conditional attributes and values
  - Product attribute used to set which products the topic is applicable to
  - Platform attribute used to set which platforms the topic is applicable to
  - Audience attribute used to set which audiences may view the topic
- Conditional publishing = filtering by attribute values = determining what to exclude



# Migrating Towards Topic-Oriented Content

- Consider adapting DITA
- Choose a pilot project
- Develop a information architecture
- Develop a reuse model
- Migrate existing content



# Consider Adapting DITA

- Provides open source architecture, DTD, tools
- Combines information typing with authoring DTD
- Very active user community
- Adapted by IBM, Adobe, Nokia and many other companies
- Can continue authoring in FrameMaker
- Can continue authoring in Word



# Develop a High Level Information Architecture

- Create a knowledge model of subjects encompassing:
  - All products, features, system components
  - All life-cycle stages
- From that model, compile lists of content to be created to express the knowledge model
- Compile content to form maps representing documentation deliverables



# Examples of Subjects

Type	Subjects
Products	<ul style="list-style-type: none"><li>➤ Servers: S1, S2, S3</li><li>Endpoints: E1, E2, E3</li><li>Peripherals: P1, P2, P3</li></ul>
Platforms Supported	<ul style="list-style-type: none"><li>➤ Windows, Unix, Linux, Macintosh</li></ul>
Assemblies	<ul style="list-style-type: none"><li>➤ Assy1, Assy2, Assy3</li></ul>
Parts	<ul style="list-style-type: none"><li>➤ P101, P102, P103</li></ul>
Life Cycle	<ul style="list-style-type: none"><li>➤ Design</li><li>Sell: Demos, Positioning, Related Products, Related Services, Strategy, Target Market</li><li>Implement: Initialize, Install, Configuration, Training</li><li>Use: End Use, Administer</li><li>Maintain: Troubleshoot, Optimize, Service, Maintenance</li></ul>

Other Subject Types: Features, Tasks, Interfaces, Screens, Use Cases, User Roles, Legal Info



# Subject Associations

- Define relationships between subject types to help compile the full list of subjects to be documented
  - Solution ↔ System Component
  - Product ↔ Platform Supported
  - Product ↔ Feature
  - Feature ↔ Task
  - Task ↔ Interface
  - Task ↔ Life Cycle
  - Task ↔ User Role
  - Product ↔ Assembly
  - Assembly ↔ Part



# Setting Up Your High Level Subject Taxonomy

- List Solutions your company offers
- For each Solution: list System Components
- For each Product:
  - Associate with System Components
  - Associate with Platforms Supported
  - List Features
- For each Feature:
  - Associate with Platforms Supported
  - Associate with User Roles
  - List Tasks
- For each Task:
  - List Screens
  - Associate with Life Cycle Stage
  - Associate with User Roles



# Generate List of Content to be Created

- Set rules: Topics needed to express each subject

<b>For Each:</b>	<b>Create Content Topic:</b>
Product	➤ Overview (DITA Concept), Legal license / liability / warrantee
Feature	➤ Description, Use Case
Task	➤ Procedure (DITA Task), safety instruction
Screen	➤ List of buttons and fields (DITA Reference)
Assembly	➤ Installation Task, Maintenance Task, List of Parts
Part	➤ Description, Supplier, Replacement Part



# Author Content

- For each content instance, author knows the topic to be described and the characteristics

Example:

Procedure: Configuring Server Ports (DITA Task)

Characteristics:

- Product: Server S1
- Task: Configure Server Ports
- User Roles: Administrator, Implementer
- Interface: Admin
- Screen: Port Configuration
- Life Cycle Stage: Installation, Configuration
- Platform: Windows, Unix, Linux

Related Content:

- Screen element reference, Screen capture



# Generate Deliverables

- Set rules for each type of deliverable to create maps

<b>Deliverable</b>	<b>Contents</b>
<b>Installation Guide</b>	<ul style="list-style-type: none"><li>➤ Product Overview, Legal References, Installation and Configuration Procedures associated with User Role=Implementer</li></ul>
<b>Admin Guide</b>	<ul style="list-style-type: none"><li>➤ Product Overview, Legal References</li><li>➤ For each interface associated with Admin, Implementer, Technician<ul style="list-style-type: none"><li>▪ For each screen, include screen reference</li></ul></li><li>➤ For each Task associated with life cycle stage: Installation, Configuration, Administer, Troubleshoot<ul style="list-style-type: none"><li>▪ Include Procedures (DITA Task)</li></ul></li></ul>



# Generate Deliverables

- Set rules for each type of deliverable to create maps

<b>Deliverable</b>	<b>Contents</b>
--------------------	-----------------

---

- |                   |   |
|-------------------|---|
| <b>User Guide</b> | <ul style="list-style-type: none"><li>➤ Product Overview, Legal References</li><li>➤ Getting Started<ul style="list-style-type: none"><li>▪ For each Task associated with User Role: End User, Life Cycle: Installation, Configuration, include all procedures</li></ul></li><li>➤ Features<ul style="list-style-type: none"><li>For each Feature<ul style="list-style-type: none"><li>▪ Include Feature Description</li><li>▪ For each Task associated with Life Cycle: Use, User Role: End User include Procedure</li></ul></li></ul></li></ul> |
|-------------------|---|
-



# Develop a Reuse Model

- Weigh the additional effort of maintenance versus the benefit of a reduced documentation set
  - To derive benefit from reuse requires planning and organization
  - The lower the level of conditionalization, the more planning required and more effort is required to maintain
- Reuse model should reflect your high level information architecture
- Determine at which level to reuse
  - Topic
  - Component
  - Variable text



# Develop a Reuse Model

- Set rules when permissible to conditionalize and in what cases
  - Topic: included at the map level
  - Component: e.g. alternate set of task steps for different products using the same topic
  - Text: e.g. task steps that should be excluded for certain platforms
  - Variables: e.g. product name changes, company branding
- Determine conditional categories needed
  - Attributes (XML)
  - Groupings of tags (FM, Word)
- Set permissible values
  - For each attribute (XML)
  - Tag names (FM, Word)



# Choose a pilot project

- Analyze current documentation requirements
- Identify areas where reuse is possible and practical
- Questions to answer:
  - Will the documentation serve more than one product, customer, configuration, audience?
  - Is it being published to multiple outputs with minor differences?
  - Can the documentation be shared with others in the organization, e.g. training, service, support, online knowledgebase?
    - Is some or all of the documentation translated?
- Choose an initial project where benefits will be most clearly felt



# Migrating Legacy Content

## Categorize existing content

- List all the topics in the book (automated tools available)
- Categorize based on subject
- Determine where topics can be combined
- Determine which topics should be re-written
- Determine which topics may be converted



## Be in touch

- ❑ Your feedback and ideas are appreciated!
- ❑ Our team can help your organization evaluate and implement a solution suitable to your own documentation requirements

Joe Gelb  
[joe@suite-sol.com](mailto:joe@suite-sol.com)