

# Modular Documentation

*Bringing the Pieces Together*

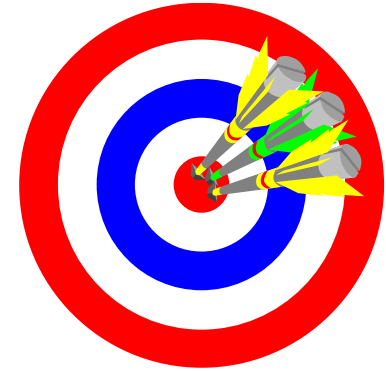
X-Pubs 2008, London

Joe Gelb  
President, Suite Solutions





# Main Discussion Points



- ❑ Who is this guy?
- ❑ What is information architecture?  
Why is it important for documentation?
- ❑ What is modular documentation?
- ❑ How can Topic Maps help?
- ❑ Methodology for developing content using  
Topic Maps

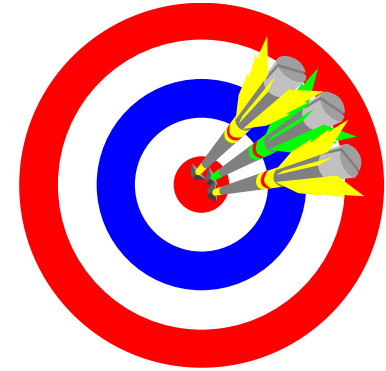


# Who is this guy?

- Background in Engineering and Process Planning
- CTO for a leading techdoc service provider and technology vendor
  - Built and managed professional services group
  - Built CMS with ATA iSpec and DITA support
- Left to form vendor-independent consulting and implementation group
- Goal: implement creative and pragmatic solutions based on accepted standards and best practices



# Main Discussion Points



- ✓ Who is this guy?
- What is information architecture?  
Why is it important for documentation?
- What is modular documentation?
- How can Topic Maps help?
- Methodology for developing content using  
Topic Maps



# What is Information Architecture?

- The science of expressing a model or concept for information
- Used for activities that require expressions of complex systems
- Consists of:
  - Structural design of shared information environments
  - Organizing and labeling information to support usability and findability

From *Wikipedia*



# What is IA for Documentation?

- Method for organizing documentation and training resources – text, media – into an overarching knowledge model
- The knowledge model is created and maintained separate from the actual content – like creating a global index
- “Allows us to provide access to the information based on the model of the knowledge it contains”  
[=findability]  
*Steve Newcombe*
- Simple level: Organization of content by hierarchy, relationships
- Next level: Organization of *subjects*, and relating content to those subjects

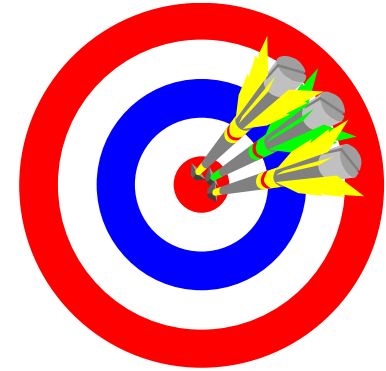


# Why it is it important for us?

- Best practices for technical content development based on *topic-oriented* content architecture
- Topic-oriented content = modular = component-based documentation
- Instead of creating deliverables, we create discrete topics of information – may be thousands
- Standards being adopted:  
DITA, S1000D, SCORM
- Need better methodologies to plan, create, classify, manage, localize, publish, deliver and find content



# Main Discussion Points

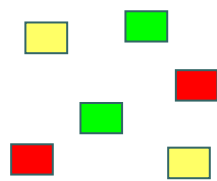


- ✓ Who is this guy?
- ✓ What is information architecture?  
Why is it important for documentation?
- What is modular documentation?
- How can Topic Maps help?
- Methodology for developing content using  
Topic Maps



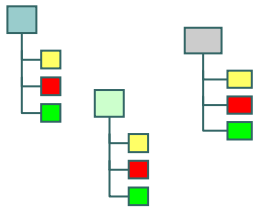
# Component Based Content

## ➤ DITA Topics, S1000D Data Modules



- Each component answers a single question
- Only enough information to understand one **concept**, perform one **task** or provide one set of **reference** information

## ➤ DITA Maps, S1000D Publication Modules



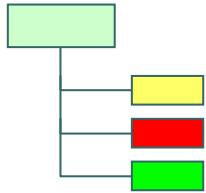
- Assemble components into deliverables
- Define relationships between components

## ➤ Classification

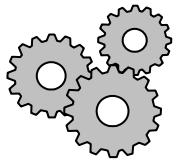
- Assign metadata, applicability / effectivity
- DITA: conditional attributes: Product, Audience, Platform, etc.



# Assembling Components into Deliverables



- Create different types of maps:
  - **Solution-oriented**  
How products and procedures work together
  - **Task-oriented**  
How to accomplish a specific goal
  - **Feature-oriented**  
What does a product or component do
  - **Training Course**

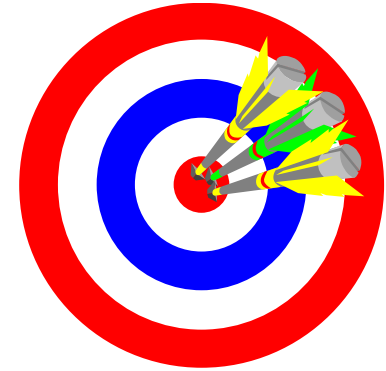


- Automated Publishing
  - Filter by product, audience, other attributes
  - Publish using Open Source Toolkit to multiple formats or to a website





# Main Discussion Points



- ✓ Who is this guy?
- ✓ What is information architecture?  
Why is it important for documentation?
- ✓ What is modular documentation?
  - How can Topic Maps help?
  - Methodology for developing content using Topic Maps



# What's an Ontology?

- Dr. Leo Obrst, MITRE Information Semantics Group

- Terms used to describe and represent an area of knowledge – subject matter
  - Model for the meaning of those terms
  - Vocabulary and the meaning of that vocabulary
- ✱ For us, that means standardized index, glossary, conditional tags/conditional attribute values
- Used by people, databases and applications to navigate, identify and share information in a specific *domain* (specific area of knowledge)
- ✱ For us, that means better planning and content reuse on the authoring side and better delivery on the production side

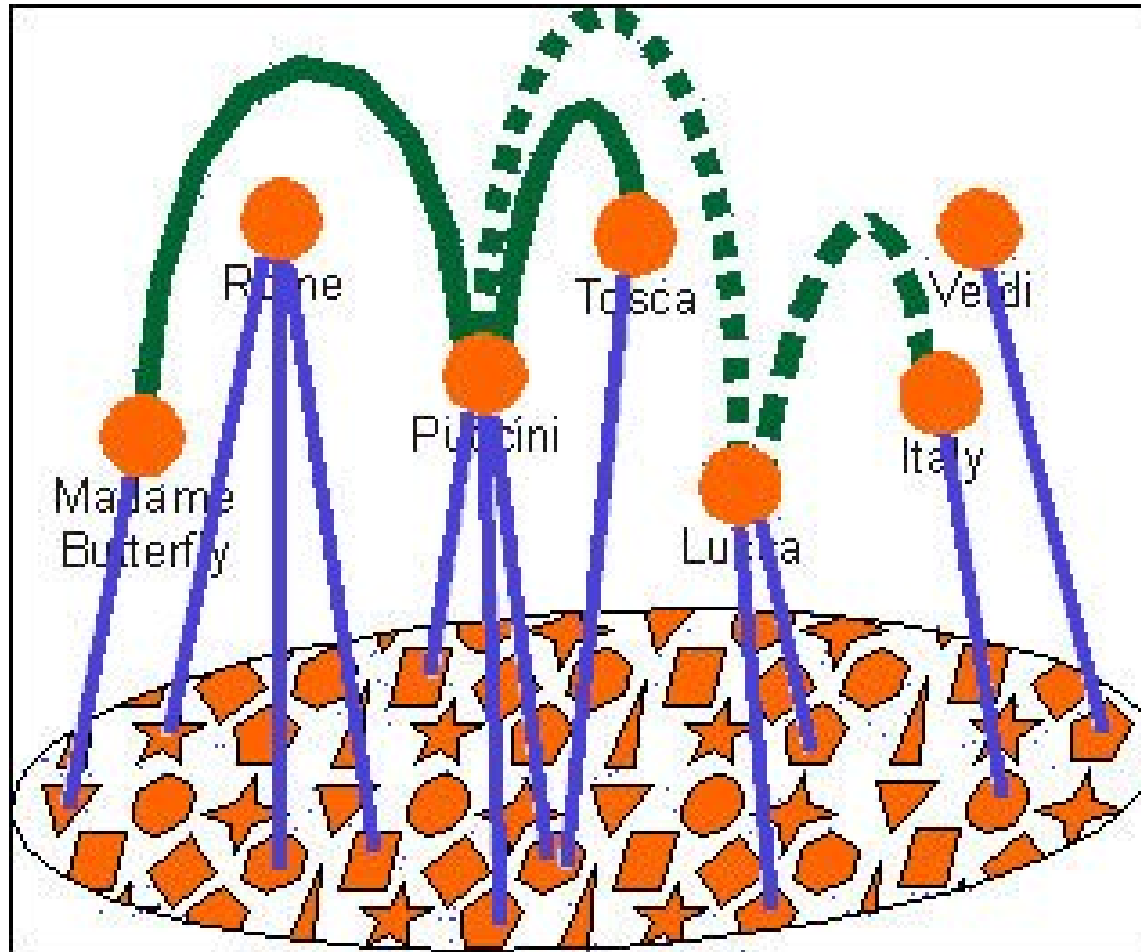


# Topic Maps

- Semantic knowledge model to express an ontology
- ISO standard; can also be expressed as XML
- Topic
  - Represents a subject or concept
  - Categorized by Type
- Map
  - Associations: relationships between topics
  - Occurrences: link between information resources and the subjects that they express something about



# Graphical Representation of a Topic Map



"The TAO of Topic Maps," Steve Pepper

Suite Solutions

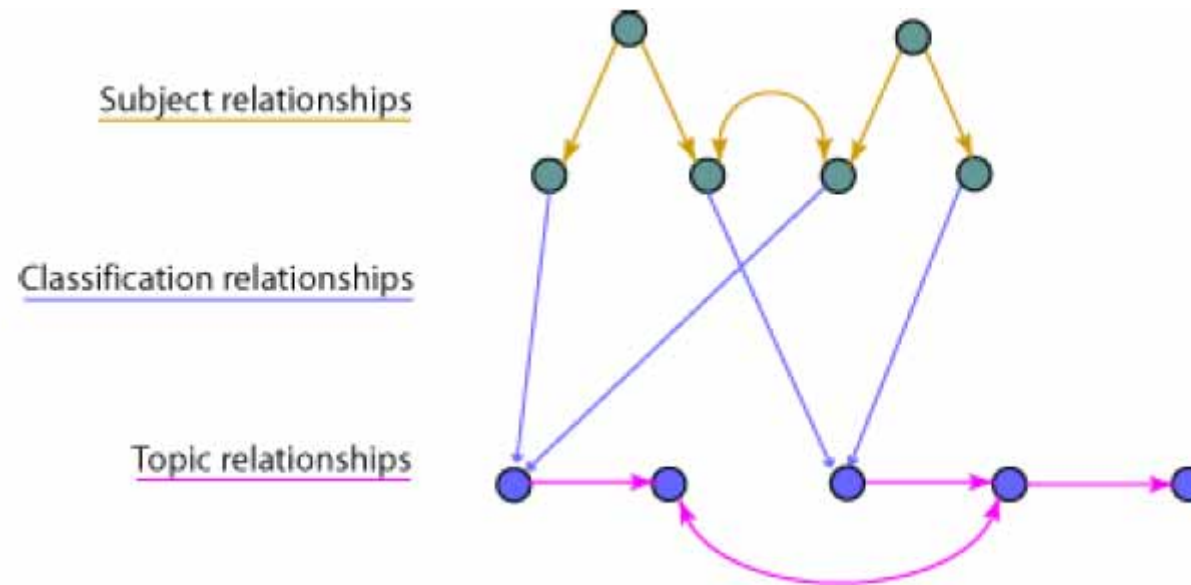


# SKOS: Simple Knowledge Organization System

- Standard for modeling an ontology
- Similar to Topic Maps, however it is expressed using RDF, the accepted syntax of expression for the semantic web
- Thus, SKOS bridges the gap between traditional indexing and formal ontologies for the semantic web
- Define subjects and if desired, organize them into a taxonomy (i.e. hierarchy)
- Classify content to indicate its subject
- DITA Taxonomy specialization allows you to maintain the ontology in DITA and transform it into SKOS RDF for processing and use with standard tools



# SKOS: Simple Knowledge Organization System



“Subject Classification with DITA and SKOS,” Henum, Anderson and Bird, October 2005



# How can Topic Maps Help?

## DITA and Topic Maps: A Natural Match

### DITA:

- Topic-oriented
- Content model
- Develop and reuse content
- Combine topics using Maps
- Separates taxonomy and relationships from content
- Robust tools for authoring and publishing

### Topic Maps:

- Subject-centric
- Semantic model
- Capture and reuse semantics
- Relate subjects using Maps
- Separates subject definition from the content resources
- Robust tools for ontology design, search and navigation




# How can Topic Maps Help?

- Provide a comprehensive ontology for our products
  - Topics: Defines subjects being expressed in the documentation: text, graphics, other media
  - Occurrences: Content topics that relate to each subject
  - Associations: Relationships between subjects can be used for:
    - filtering
    - multi-purpose publishing
    - linking between the content topics



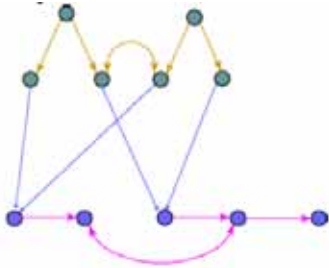
# How can Topic Maps Help?

- ▶ Determine valid usage of metadata and applicability / effectivity values (DITA conditional attributes)
- ▶ Standardize index and glossary
-  ▶ Associative topic navigation, not just hierarchical
- ▶ Results in better planning and content reuse on the authoring side, robust filtering and delivery on the production side



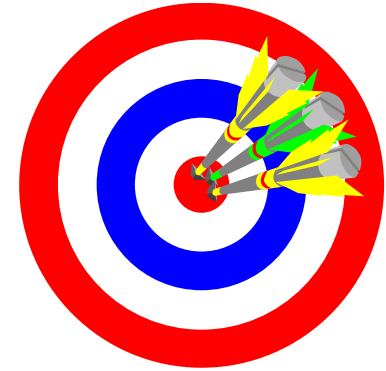
# Interoperability with TM

- Express Topic Maps using DITA Maps via specialization
  - Subjects and their relationships
  - Classification of content resources
  - Prototype built for SKOS can be adapted
- DITA topics can be used to document the ontology
  - Describe each subject: text, images, variants, etc.
  - Publish glossary using the Open Source Toolkit
- Round-trip with XTM, LTM, CTM gives best of both worlds
  - Topic Maps tools for building, checking, navigating, searching, querying, constraining ontologies
  - DITA and S1000D tools for authoring, managing, publishing, translating





# Main Discussion Points



- ✓ Who is this guy?
- ✓ What is information architecture?  
Why is it important for documentation?
- ✓ What is modular documentation?
- ✓ How can Topic Maps help?
- Methodology for developing content using  
Topic Maps



# Methodology for Content Development Using Topic Maps

## Solution Oriented Topic Architecture (SOTA)

1. Create a knowledge model of subjects encompassing:
  - All products, features, system components
  - All product life-cycle stages
2. From that model, compile lists of content to be created to express the knowledge model
3. Compile content to form maps representing documentation deliverables
  - DITA Maps
  - S1000D Publication Modules
  - SCORM Manifests



# Define Subjects and Types

- 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 Interfaces and Screens
  - Associate with Life Cycle Stage
  - Associate with User Roles



# Define Subjects and Types

## Type Subjects

---

**Products**    > Servers: S1, S2, S3  
                  > Endpoints: E1, E2, E3  
                  > Peripherals: P1, P2, P3

---

**Platforms**    > Windows, Unix, Linux, Macintosh

---

**Assemblies**    > Assy1, Assy2, Assy3  
**Parts**            > P101, P102, P103

---

**Life Cycle**    > Design  
                  > Sell: Demos, Positioning, Related Products, Related Services,  
                  > Implement: Initialize, Install, Configuration, Training  
                  > Use: End Use, Administer  
                  > Maintain: Troubleshoot, Optimize, Service, Maintenance

---

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

---



# Define Associations

- Define relationships between subjects
  - Solution ↔ System Component
  - Product ↔ Platform Supported
  - Product ↔ Feature
  - Feature ↔ Task
  - Task ↔ Interface
  - Task ↔ Life Cycle
  - Task ↔ User Role
  - Product ↔ Assembly
  - Assembly ↔ Part



# Content Resources - Occurrences

- Different types of content resources are used to express information about the subjects



- Procedure / Task
- Concept / Description
- Reference



- Illustration
- Drawing



- Flash
- Video
- Presentation



# Generate List of Content to be Created

- Set rules: Occurrence types needed to express each subject

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



# Author Content

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

Example:

Procedure: Configuring Server Ports (DITA Task)

Characteristics / Associations:

- 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> |
|-------------------|---|



# What do you think?

- Which planet is he on?
- Interesting, but is it feasible?
- Interesting; we should consider this approach when evaluating our information architecture
- Cool! I see where this can really help and want to learn more about it



## Be in touch

- ❑ Your feedback and ideas are appreciated!
- ❑ Our team can help you evaluate your needs and implement the right solution.

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