Learn about the new features of DITA 1.3

Presented by:

The OASIS DITA Technical Committee

and

The OASIS DITA Adoption Technical Committee

Welcome

JoAnn Hackos
President, Comtech Services, Inc.
joann.hackos@comtech-serv.com

Chair, OASIS DITA Adoption Technical Committee

- Today’s program
- Our presenters

Yes, we are recording this presentation.

Agenda

- Welcome – JoAnn Hackos
- Introduction – Kris Eberlein
- Context Sensitive Help – Stan Doherty
- Branch Filtering – Robert Anderson
- Key Scopes – Chris Nitchie
- Learning and Training – Mark Myers
- Release Management – Tom Cihak
- DITA Domains – Nancy Harrison
- Troubleshooting – Bob Thomas
- Q&A

Introduction

Kristen James Eberlein
Eberlein Consulting LLC
kris@eberleinconsulting.com

Chair, OASIS DITA Technical Committee

- Who are we?
- Where have we been?
- Where are we going?

So what is the OASIS DITA Technical Committee?

- A group of 16 people who meet on a telephone call each week for 60 minutes.
- We use Robert’s Rules of Order, but work by consensus most of the time.
- Some of us have NEVER met in person …
- We (or our corporate employers) pay so that we can participate in developing DITA.
- For most of us, it’s a labor of love ☺

Background of DITA releases

- DITA 1.0 (June 2005)
  - Information types: Topic, map, concept, task, reference. ditabase
  - Architecture: Specialization; conref; map-based organization & linking
  - Domains: Software; user interface; programming; highlighting; utilities
- DITA 1.1 (May 2007)
  - Information type: Book map; glossentry
  - Architecture: Addition of DITAVAL for conditional processing; attribute specialization
  - Domains: xNAL; indexing
  - New elements: <abstract>, <foreign>, <data>

Background of DITA releases (cont.)

- DITA 1.2 (December 2010)
  - Information types:
    - Learning & Training specialization
    - General task; machinery task
    - Subject scheme; classification map
  - Architecture:
    - Key-based addressing for links and variable text
    - Additions to conref; constraints
    - Topic and domain integration
  - Domains:
    - Classification; delayed conref
    - Task requirements; hazard statement
  - New elements: Many!

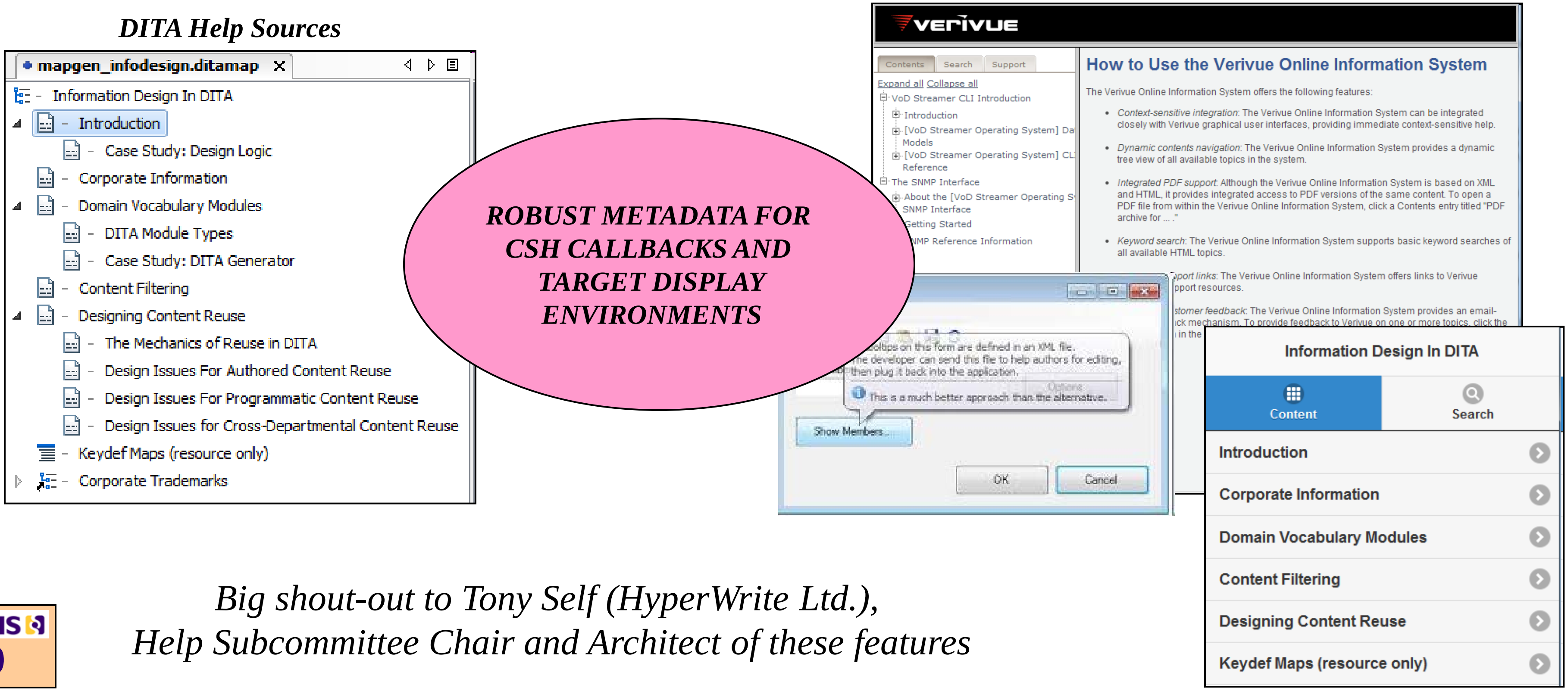
What's coming with DITA 1.3?

- I'm going to let today's panelists address this issue, and then we'll summarize at the end of the Webinar, after the questions.
- Some invisible changes
    - Improvement to quality of specification
    - Ability to single source the XML grammar files
    - Improvement in TC processes and overall reduction of technical debt
    - Release of the specification in three packages
        - Base
        - Technical content
        - Complete

CSH Callback and Display Metadata

Stanley Doherty, Ph.D.
OASIS DITA TC, Help SC, TechComm SC
Stan@ModularWriting.com

Multiple Help Targets



DITA Help Sources

mapgen_infodesign.ditamap

- Information Design In DITA
  - Introduction
    - Case Study: Design Logic
  - Corporate Information
  - Domain Vocabulary Modules
    - DITA Module Types
    - Case Study: DITA Generator
  - Content Filtering
  - Designing Content Reuse
    - The Mechanics of Reuse in DITA
    - Design Issues For Authored Content Reuse
    - Design Issues For Programmatic Content Reuse
    - Design Issues for Cross-Departmental Content Reuse
  - Keydef Maps (resource only)
  - Corporate Trademarks

ROBUST METADATA FOR CSH CALLBACKS AND TARGET DISPLAY ENVIRONMENTS

- VoD Streamer CLI Introduction
  - Introduction
  - [VoD Streamer Operating System] Da Models
  - [VoD Streamer Operating System] CL Reference
- The SNMP Interface
  - About the [VoD Streamer Operating S SNMP Interface
- Getting Started
- NMP Reference Information

How to Use the Verivue Online Information System

The Verivue Online Information System offers the following features:

- *Context-sensitive integration*: The Verivue Online Information System can be integrated closely with Verivue graphical user interfaces, providing immediate context-sensitive help.
- *Dynamic contents navigation*: The Verivue Online Information System provides a dynamic tree view of all available topics in the system.
- *Integrated PDF support*: Although the Verivue Online Information System is based on XML and HTML, it provides integrated access to PDF versions of the same content. To open a PDF file from within the Verivue Online Information System, click a Contents entry titled "PDF archive for ... ."
- *Keyword search*: The Verivue Online Information System supports basic keyword searches of all available HTML topics.
- *pport links*: The Verivue Online Information System offers links to Verivue pport resources.
- *stomer feedback*: The Verivue Online Information System provides an email- ck mechanism. To provide feedback to Verivue on one or more topics, click the in the

oltips on this form are defined in an XML file. he developer can send this file to help authors for editing, then plug it back into the application.

This is a much better approach than the alternative.

Information Design In DITA

- Introduction
- Corporate Information
- Domain Vocabulary Modules
- Content Filtering
- Designing Content Reuse
- Keydef Maps (resource only)

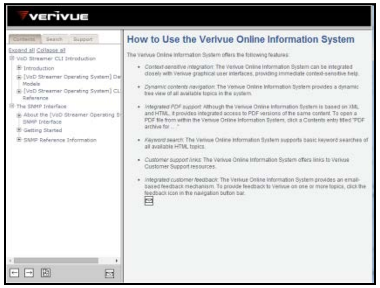
Big shout-out to Tony Self (HyperWrite Ltd.), Help Subcommittee Chair and Architect of these features

CSH Callbacks and Display

Problem statement: We are asked to generate an increasing number of user assistance targets from the same DITA sources.

TARGETS

Tri-pane Help

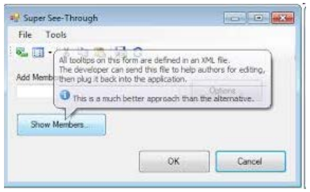

verivue
Contents
Search
Support
Expand all
Collapse all
VoD Streamer CLI Introduction
Introduction
[VoD Streamer Operating System] De Models
[VoD Streamer Operating System] CL Reference
The SNMP Interface
About the [VoD Streamer Operating S SNMP Interface
Getting Started
SNMP Reference Information

How to Use the Verivue Online Information System

The Verivue Online Information System offers the following features:

- *Context-sensitive integration*: The Verivue Online Information System can be integrated closely with Verivue graphical user interfaces, providing immediate context-sensitive help.
- *Dynamic contents navigation*: The Verivue Online Information System provides a dynamic tree view of all available topics in the system.
- *Integrated PDF support*: Although the Verivue Online Information System is based on XML and HTML, it provides integrated access to PDF versions of the same content. To open a PDF file from within the Verivue Online Information System, click a Contents entry titled "PDF archive for ... "
- *Keyword search*: The Verivue Online Information System supports basic keyword searches of all available HTML topics.
- *Customer support links*: The Verivue Online Information System offers links to Verivue Customer Support resources.
- *Integrated customer feedback*: The Verivue Online Information System provides an email-based feedback mechanism. To provide feedback to Verivue on one or more topics, click the feedback icon in the navigation button bar.

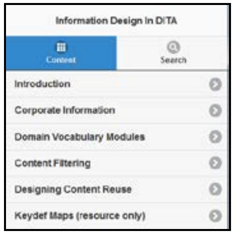
Embedded User Assistance



All tooltips on this form are defined in an XML file. The developer can send this file to help authors for editing, then plug it back into the application.

This is a much better approach than the alternative.

Mobile Help



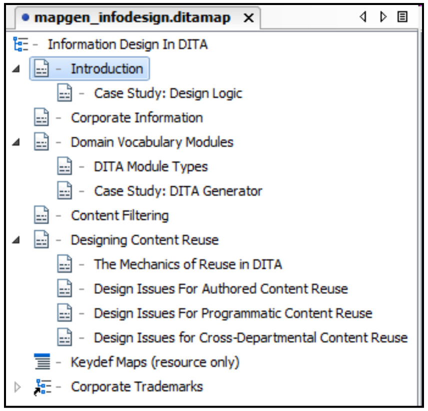
- Introduction
- Corporate Information
- Domain Vocabulary Modules
- Content Filtering
- Designing Content Reuse
- Keydef Maps (resource only)

CSH Callbacks and Display

Problem statement: We are asked to generate an increasing number of user assistance targets from the same DITA sources.

If each target form of user assistance requires callback and display metadata in some proprietary format(s), we get into trouble if we have to maintain this metadata apart from our DITA sources.

DITA Help Sources


- mapgen_infodesign.ditamap ×
- Information Design In DITA
  - Introduction
    - Case Study: Design Logic
  - Corporate Information
  - Domain Vocabulary Modules
    - DITA Module Types
    - Case Study: DITA Generator
  - Content Filtering
  - Designing Content Reuse
    - The Mechanics of Reuse in DITA
    - Design Issues For Authored Content Reuse
    - Design Issues For Programmatic Content Reuse
    - Design Issues for Cross-Departmental Content Reuse
  - Keydef Maps (resource only)
  - Corporate Trademarks

callback IDs

display information

callback IDs

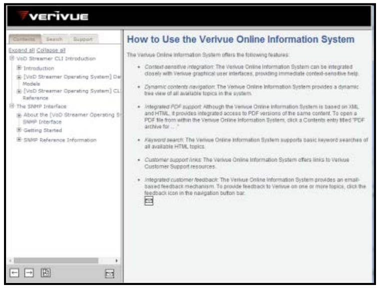
display information

callback IDs

display information

TARGETS

Tri-pane Help



Expand all Collapse all

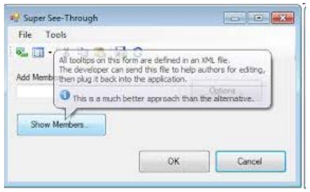
- VoD Streamer CLI Introduction
  - Introduction
  - [VoD Streamer Operating System] De Models
  - [VoD Streamer Operating System] CL Reference
- The SNMP Interface
  - About the [VoD Streamer Operating S SNMP Interface
  - Getting Started
  - SNMP Reference Information

How to Use the Verivue Online Information System

The Verivue Online Information System offers the following features:

- *Context-sensitive integration*: The Verivue Online Information System can be integrated closely with Verivue graphical user interfaces, providing immediate context-sensitive help.
- *Dynamic contents navigation*: The Verivue Online Information System provides a dynamic tree view of all available topics in the system.
- *Integrated PDF support*: Although the Verivue Online Information System is based on XML and HTML, it provides integrated access to PDF versions of the same content. To open a PDF file from within the Verivue Online Information System, click a Contents entry titled "PDF archive for ... "
- *Keyword search*: The Verivue Online Information System supports basic keyword searches of all available HTML topics.
- *Customer support links*: The Verivue Online Information System offers links to Verivue Customer Support resources.
- *Integrated customer feedback*: The Verivue Online Information System provides an email-based feedback mechanism. To provide feedback to Verivue on one or more topics, click the feedback icon in the navigation button bar.

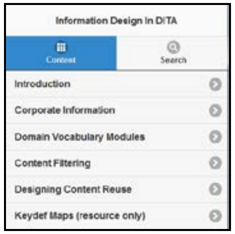
Embedded User Assistance



All tooltips on this form are defined in an XML file. The developer can send this file to help authors for editing, then plug it back into the application.

This is a much better approach than the alternative.

Mobile Help



Information Design In DITA

- Introduction
- Corporate Information
- Domain Vocabulary Modules
- Content Filtering
- Designing Content Reuse
- Keydef Maps (resource only)

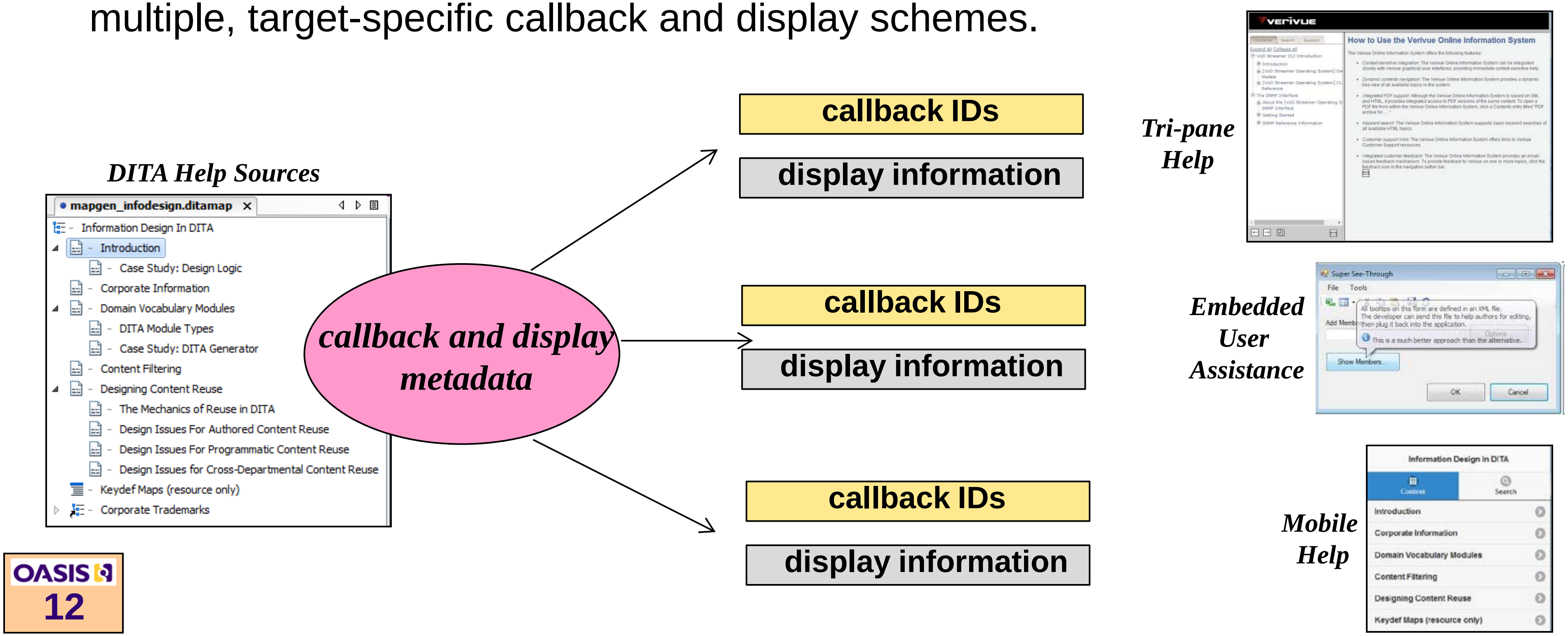
CSH Callbacks and Display

Problem statement: We are asked to generate an increasing number of user assistance targets from the same DITA sources.

If each target form of user assistance requires callback and display metadata in some proprietary format(s), we get into trouble if we have to maintain this metadata apart from our DITA sources.

Solution: Extend DITA metadata to support the generation of multiple, target-specific callback and display schemes.

TARGETS



DITA Help Sources

mapgen_infodesign.ditamap ×

- Information Design In DITA
  - Introduction
    - Case Study: Design Logic
  - Corporate Information
  - Domain Vocabulary Modules
    - DITA Module Types
    - Case Study: DITA Generator
  - Content Filtering
  - Designing Content Reuse
    - The Mechanics of Reuse in DITA
    - Design Issues For Authored Content Reuse
    - Design Issues For Programmatic Content Reuse
    - Design Issues for Cross-Departmental Content Reuse
  - Keydef Maps (resource only)
  - Corporate Trademarks

callback and display metadata

callback IDs

display information

callback IDs

display information

callback IDs

display information

Tri-pane Help

verivue

Contents Search Support

Expand all Collapse all

- VoD Streamer CLI Introduction
  - Introduction
  - [VoD Streamer Operating System] Da Models
  - [VoD Streamer Operating System] CLI Reference
- The SNMP Interface
  - About the [VoD Streamer Operating S SNMP Interface
  - Getting Started
  - SNMP Reference Information

How to Use the Verivue Online Information System

The Verivue Online Information System offers the following features:

- *Context-sensitive integration*: The Verivue Online Information System can be integrated closely with Verivue graphical user interfaces, providing immediate context-sensitive help.
- *Dynamic contents navigation*: The Verivue Online Information System provides a dynamic tree view of all available topics in the system.
- *Integrated PDF support*: Although the Verivue Online Information System is based on XML and HTML, it provides integrated access to PDF versions of the same content. To open a PDF file from within the Verivue Online Information System, click a Contents entry titled "PDF archive for ... "
- *Keyword search*: The Verivue Online Information System supports basic keyword searches of all available HTML topics.
- *Customer support links*: The Verivue Online Information System offers links to Verivue Customer Support resources.
- *Integrated customer feedback*: The Verivue Online Information System provides an email-based feedback mechanism. To provide feedback to Verivue on one or more topics, click the feedback icon in the navigation button bar.

Embedded User Assistance

Super See-Through

File Tools

Add Memb

All tooltips on this form are defined in an XML file.
The developer can send this file to help authors for editing, then plug it back into the application.

This is a much better approach than the alternative.

Options

Show Members...

OK Cancel

Mobile Help

Information Design In DITA

Content Search

- Introduction
- Corporate Information
- Domain Vocabulary Modules
- Content Filtering
- Designing Content Reuse
- Keydef Maps (resource only)

Managing Target-Specific Callbacks

DITA 1.3 ADDED: Three metadata attributes to `<resourceid>`.

- `@appid` – Identifies a referenced topic to an external application.
- `@ux-context-string` – Specifies a context ID for that topic.
- `@ux-source-priority` – Specifies how to resolve conflicts between callbacks defined in a map or in a topic.

Managing Target-Specific Callbacks

DITA 1.3 ADDED: Three metadata attributes to `<resourceid>`.

- `@appid` – Identifies a referenced topic to an external application.
- `@ux-context-string` – Specifies a context ID for that topic.
- `@ux-source-priority` – Specifies how to resolve conflicts between callbacks defined in a map or in a topic.

DITA Map

```
<map>
<topicref href="dialog-1.dita"/>
   <topicmeta>
     <resourceid
       appname="ios7a"
       appid="iphone"
       ux-content-string="callback_4437"
       ux-source-priority="map-takes-priority"/>
     <resourceid
       appname="kitkat"
       appid="droid"
       ux-content-string="id#4500"
       ux-source-priority="map-takes-priority"/>
   </topicmeta>
 </topicref>
</map>
```

Managing Target-Specific Callbacks

DITA 1.3 ADDED: Three metadata attributes to `<resourceid>`.

DITA Topic

```
<topic>
 <prolog>
   <resourceid
    appname="kitkat"
    appid="droid"
    ux-content-string="id#4501"/>
 </prolog>
</topic>
```

DITA Map

```
<map>
<topicref href="dialog-1.dita"/
   <topicmeta>
     <resourceid
       appname="ios7a"
       appid="iphone"
       ux-content-string="callback_4437"
       ux-source-priority="map-takes-priority"/>
     <resourceid
       appname="kitkat"
       appid="droid"
       ux-content-string="id#4500"
       ux-source-priority="map-takes-priority"/>
   </topicmeta>
 </topicref>
</map>
```

Managing Target-Specific Callbacks

DITA 1.3 ADDED: Three metadata attributes to `<resourceid>`.

DITA Topic

```
<topic>
 <prolog>
   <resourceid
    appname="kitkat"
    appid="droid"
    ux-content-string="id#4501"/>
 </prolog>
</topic>
```

DITA Map

```
<map>
<topicref href="dialog-1.dita"/
   <topicmeta>
     <resourceid
       appname="ios7a"
       appid="iphone"
       ux-content-string="callback_4437"
       ux-source-priority="map-takes-priority"/>
     <resourceid
       appname="kitkat"
       appid="droid"
       ux-content-string="id#4500"
       ux-source-priority="map-takes-priority"/>
   </topicmeta>
 </topicref>
</map>
```

Topic-map conflict resolution

Managing Target Display Information

DITA 1.3

ADDED: One map element and one attribute for `<resourceid>`.

`<ux-window>` -- specifies information about a target display.

`@ux-windowref` -- specifies a target display profile for a topic.

Managing Target Display Information

DITA 1.3 ADDED: One map element and one attribute for `<resourceid>`.
`<ux-window>` -- specifies information about a target display.
`@ux-windowref` -- specifies a target display profile for a topic.

DITA Map

```
<map>
 <topicmeta>
  <ux-window
    id="win47" name="mobile1"
    top="10" left="20" height="400" width="500"
    features="status=yes,toolbar=no,menubar=no,location=no"
    relative="true" full-screen="no"/>
 </topicmeta>
. . .
</map>
```

Managing Target Display Information

DITA 1.3

ADDED: One map element and one attribute for `<resourceid>`.
`<ux-window>` -- specifies information about a target display.
`@ux-windowref` -- specifies a target display profile for a topic.

DITA Map

```
<map>
 <topicmeta>
  <ux-window
    id="win47" name="mobile1"
    top="10" left="20" height="400" width="500"
    features="status=yes,toolbar=no,menubar=no,location=no"
    relative="true" full-screen="no"/>
 </topicmeta>
. . .
 <topicref href="dialog-1.dita"/>
  <topicmeta>
    <resourceid
      appname="ios7a"
      appid="iphone"
      ux-content-string="callback_4437"
      ux-source-priority="map-takes-priority"
      ux-windowref="mobile1"/>
  </topicmeta></topicref></map>
```

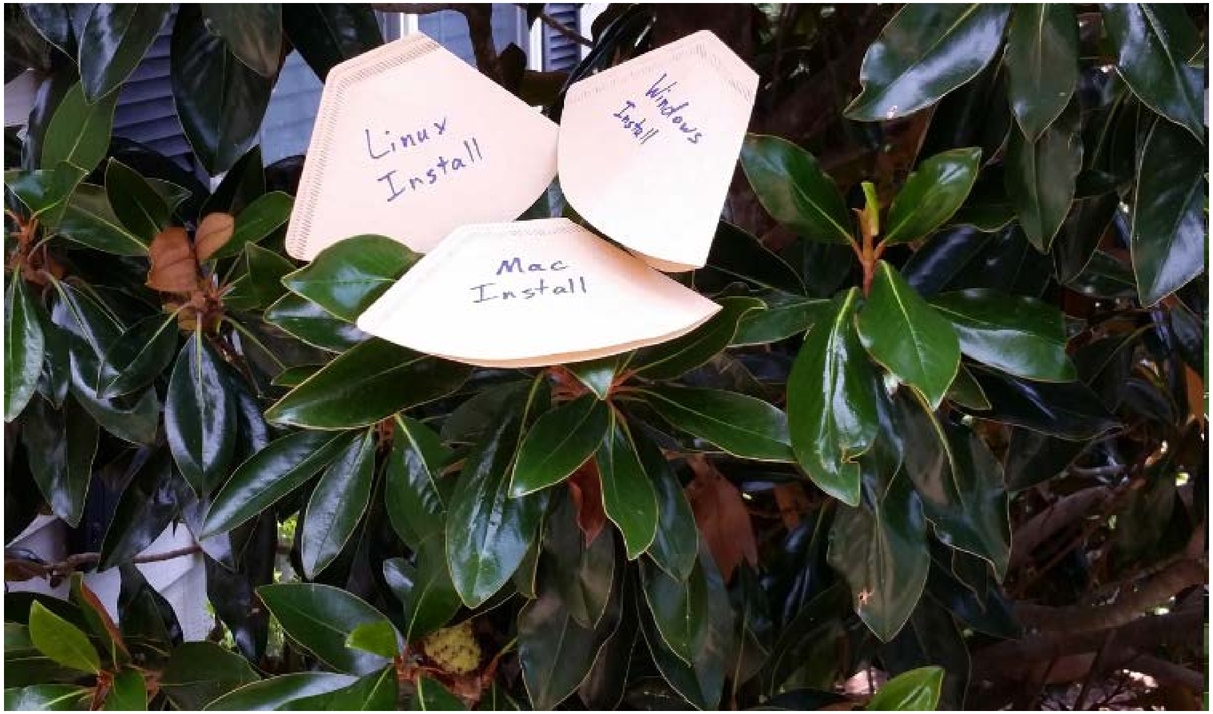
Benefits to DITA Help Practitioners

- Added semantics in DITA maps to define:
  - Multiple target applications consuming callback IDs
  - Multiple target display profiles
  - Multiple contexts or target displays for any topic
- Added logic to allow writers to manage:
  - Conflicts between IDs defined in maps and topics
  - Processing required to generate target-specific API, mapping, or configuration files
- Reduced our dependencies on external, dedicated HATs (Help Authoring Tools).

Branch Filtering

Robert Anderson, IBM

robander@us.ibm.com



Branch filtering: why?

- Problem: publication includes a subset of information with unique conditions
  - Different sections of a map for different audiences
  - Reused external content with its own conditions
  - One section of a map intended for multiple audiences
- DITA defines the “DITAVAL” syntax to set filter conditions – but no formal mechanism to associate unique conditions with portions of a publication (with branches of a map)

Branch filtering: what? how?

- New <ditavalref> element in maps creates a standard mechanism for fine-grained filtering

- Operates on that branch of the map, and on content referenced from that branch

```
<topicref href="parent.dita">
   <ditavalref href="conditions.ditaval"/>
   <topicref href="child.dita"/>
   <topicref href="child2.dita" audience="abc"/>
   <topicref href="child3.dita"/>
</topicref>
```

- Conditions apply to the map branch itself, as well as to content in parent.dita and all three children

Branch filtering: simple case

- Use <ditavalref> as in previous slide
- Conditions apply only to that section; global filters still take priority
- Whatever is filtered out at higher level, stays out
  - For example, global: filter out all but Linux. Branch: filter by Linux variant. Nested branch: filter by version.
- Potential uses:
  - Reuse of external content with unique conditions
  - Include a section that is only relevant to one audience, inside of a more general publication

Branch filtering: what else?

- Also allows publishing a single branch of content multiple times with distinct filter profiles
- Imagine:
  - Product manual with install chapter repeated by OS
  - Marketing guide with sections filtered by region
  - Homeowner’s how-to book with landscaping chapter filtered by climate or by Homeowner’s Association restrictions

Branch filtering: advanced case

- Common content for unique audiences
  - Most of a product manual applies to all audiences
  - Install chapter set up for filtering based on audience:
    - Windows/Mac/Linux; desktop vs laptop; imperial vs metric

  - DITA 1.2: publish one guide for each OS, hardware, or region
    Alternately, publish many versions, stitch together into one

  - DITA 1.3:

```
 <topicref href="install.dita">
   <ditavalref href="imperial-units.ditaval"/>
   <ditavalref href="metric-units.ditaval"/>
   <!-- … … -->
</topicref>
```

Branch filtering: advanced sample

- How did that work exactly?

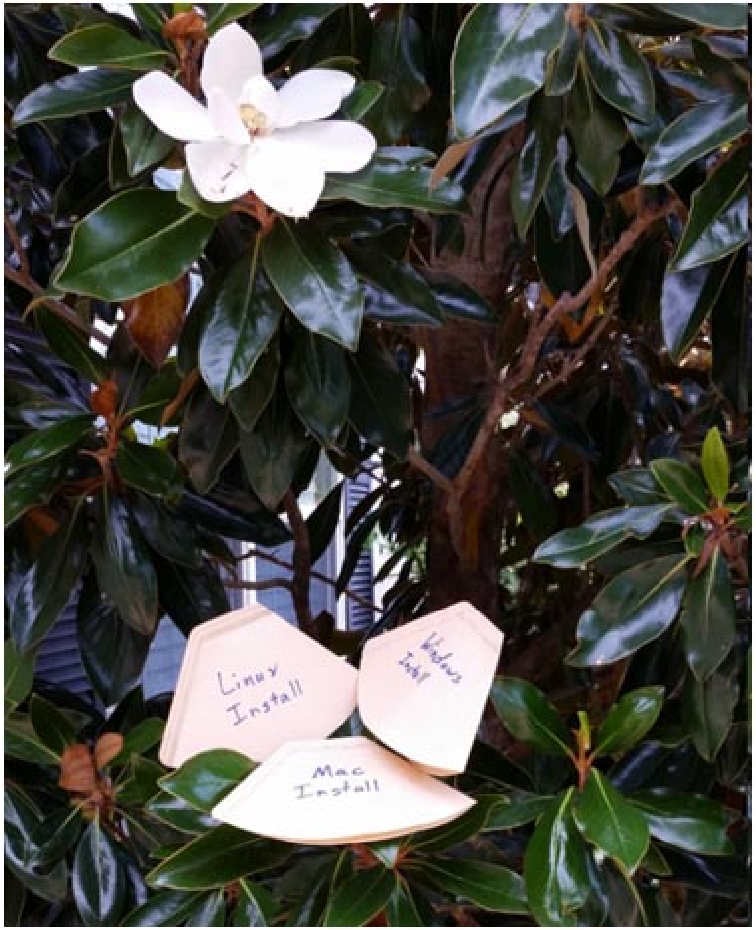
```
<topicref href="install.dita">
   <ditavalref href="imperial-units.ditaval"/>
   <ditavalref href="metric-units.ditaval"/>
   <!-- … … -->
</topicref>
… map continues with other branches …
```

- Install branch published in imperial, then metric. All other branches publish as usual.
- Imagine if this general presentation, common to this point, could now include samples tailored to each of your individual publications

Branch filtering summary

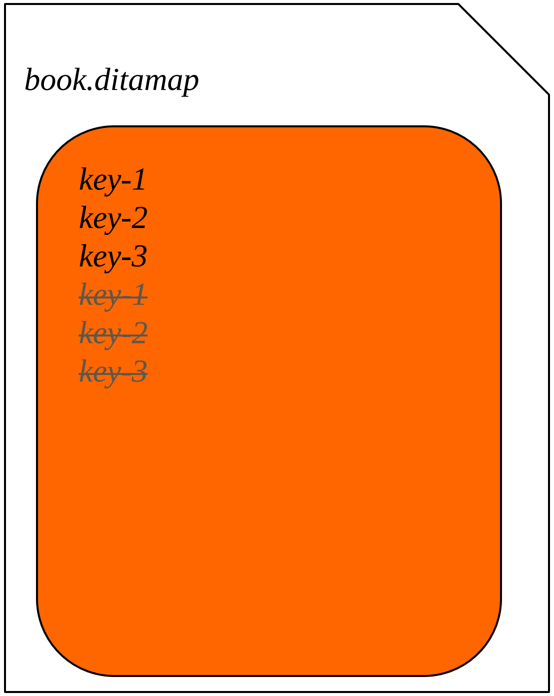
DITA 1.2:

DITA 1.3:



Key Scopes

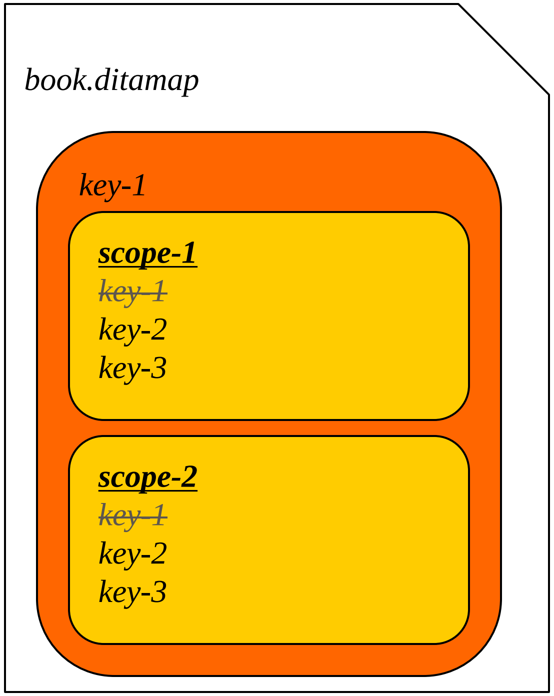
Chris Nitchie, Oberon Technologies
chris.nitchie@oberontech.com



book.ditamap

key-1
key-2
key-3
~~key-1~~
~~key-2~~
~~key-3~~

DITA 1.2 - One definition per key per publication.



book.ditamap

key-1

scope-1
~~key-1~~
key-2
key-3

scope-2
~~key-1~~
key-2
key-3

DITA 1.3 - Each scope within a publication has its own definitions.

Key Scopes – Solution

keyscope="scopeName"

- Any map or topicref can declare itself the container for a key scope by specifying the @keyscope attribute.
- Keys defined within a scope can only be directly referenced by key references in the same scope, allowing different definitions for different references.
- Scoped key definitions can be referenced from outside their scopes by including the scope name, like this: keyref="scopeName.keyName"

Key Scopes – Example

```
<map>

  <topichead keyscope="tractorX" navtitle="Tractor X">
    <topicref keys="oilChart" href="tractorX/oilChart.dita" />
    <topicref href="common/oilChangeProcedure.dita" />
  </topichead>

  <topichead keyscope="tractorY" navtitle="Tractor Y">
    <topicref keys="oilChart" href="tractorY/oilChart.dita" />
    <topicref href="common/oilChangeProcedure.dita" />
  </topichead>

</map>

... And in common/oilChangeProcedure.dita ...

<p>
For information about the type and amount of oil to use, refer to
<xref keyref="oilChart" />.
</p>
```

Key Scopes – Benefits

- Fully backwards-compatible with DITA 1.2; existing maps will continue to work exactly as they did before.
- Keyref semantics entirely unchanged.
- Topic authors do not generally need to be aware of the scope structures of the maps using the topic.
- Simple markup for declaring key scopes.

Key Scopes – New Possibilities

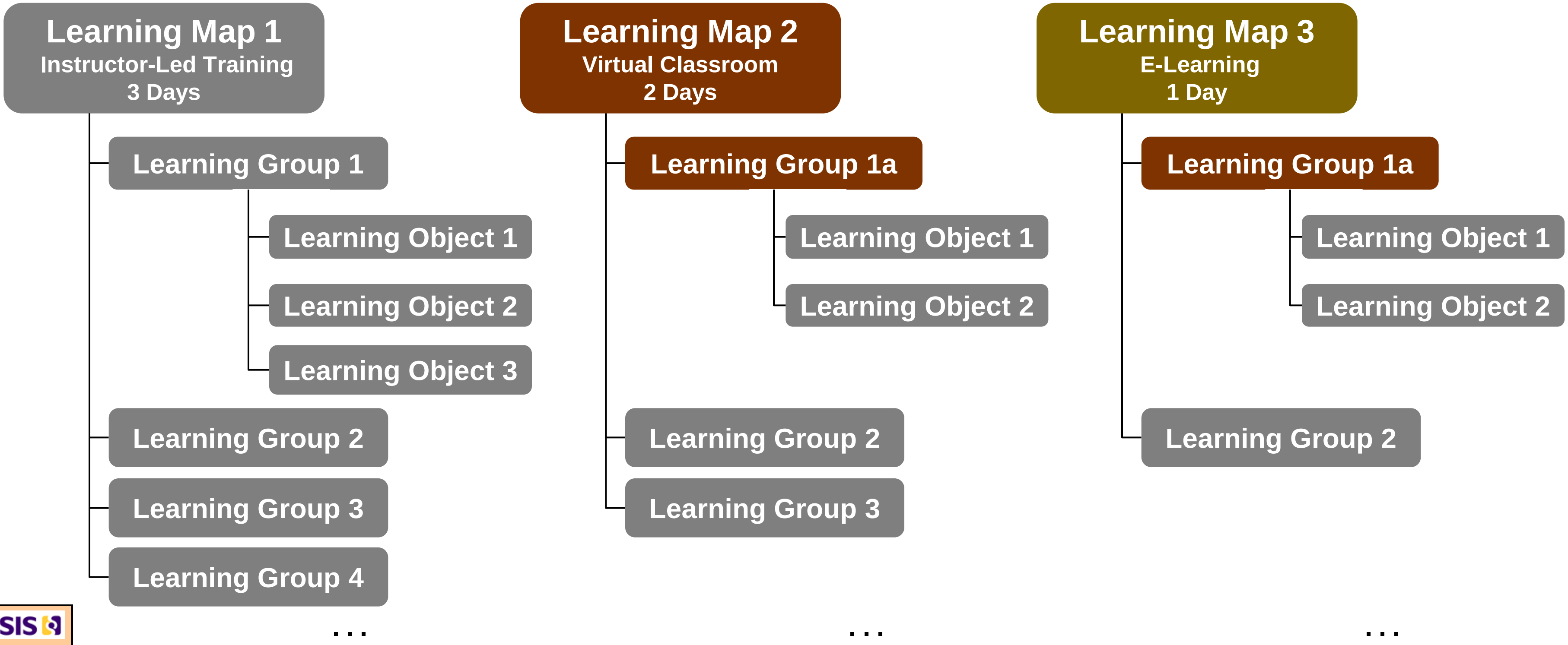
- Key scopes enable use cases that are difficult or impossible to support in DITA 1.2, including:
  - Omnibus publications combining standalone maps with conflicting keys.
  - Multi-product documentation with common key names with different meanings in different parts or chapters (e.g. keyref="introduction").
  - Multi-lingual publications with conkeyrefs to language-specific notes and warnings.

Learning and Training Updates in DITA 1.3

Mark Myers
SAP
mark.myers@sap.com

New Feature: Standardize Learning Object and Group Referencing

- Standardize markup for referencing and storing learning objects and learning groups as separate files in a file system or repository to promote reuse



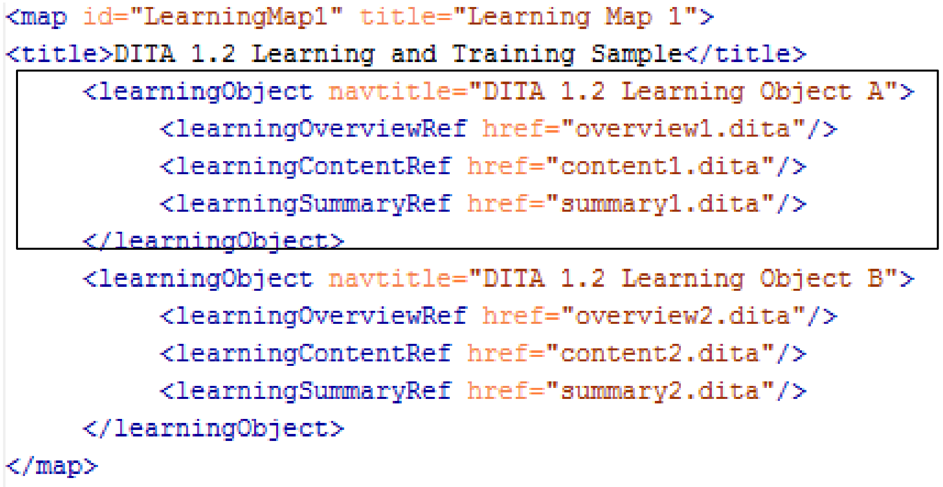
Standardize Learning Object and Group Referencing

- Learning objects and groups are intended to be reused in multiple maps
- DITA 1.2 is not consistent in the way it allows you to reference learning objects, and learning groups
- DITA 1.3 addresses this by adding:
  - Two new map refs (LearningObjectMapRef, LearningGroupMapRef)
  - Two new wrapper elements (LearningObjectMap, LearningGroupMap)

Learning Object DITA 1.2 Example

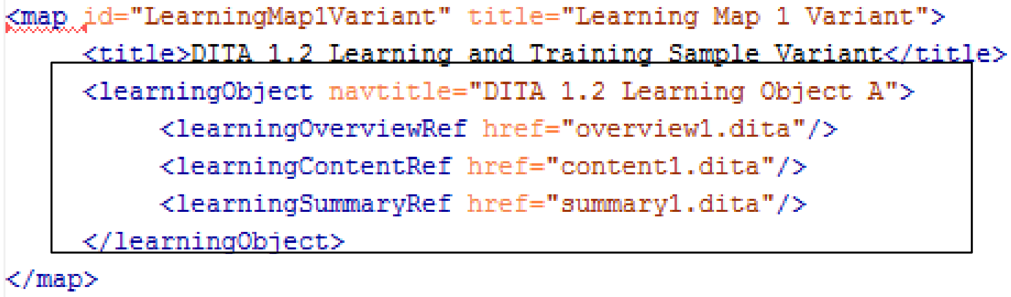
- Contents of learning object are defined, and redefined, in each map
- This allows reuse of the contents of a learning object, but not the learning object container itself
- Not ideal for true “object-based” learning object
- Inefficient and potentially error prone

Main Learning Map



```
<map id="LearningMap1" title="Learning Map 1">
<title>DITA 1.2 Learning and Training Sample</title>
    <learningObject navtitle="DITA 1.2 Learning Object A">
        <learningOverviewRef href="overview1.dita"/>
        <learningContentRef href="content1.dita"/>
        <learningSummaryRef href="summary1.dita"/>
    </learningObject>
    <learningObject navtitle="DITA 1.2 Learning Object B">
        <learningOverviewRef href="overview2.dita"/>
        <learningContentRef href="content2.dita"/>
        <learningSummaryRef href="summary2.dita"/>
    </learningObject>
</map>
```

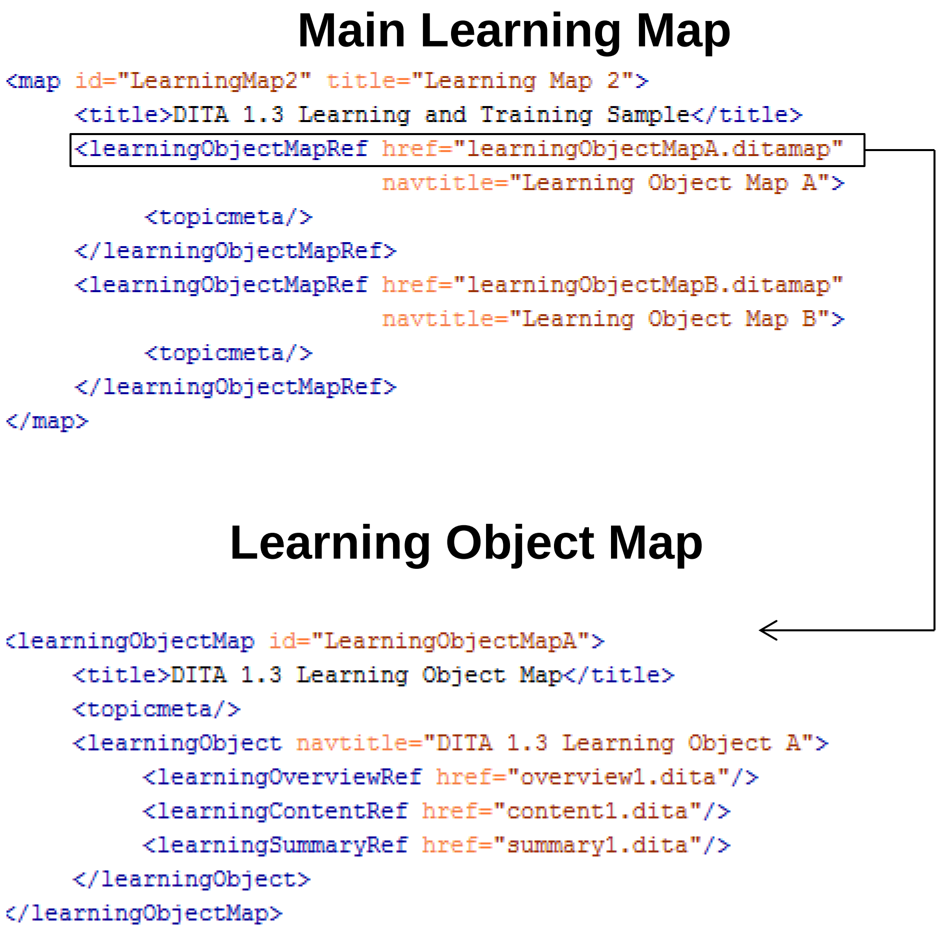
Variant Learning Map



```
<map id="LearningMap1Variant" title="Learning Map 1 Variant">
    <title>DITA 1.2 Learning and Training Sample Variant</title>
    <learningObject navtitle="DITA 1.2 Learning Object A">
        <learningOverviewRef href="overview1.dita"/>
        <learningContentRef href="content1.dita"/>
        <learningSummaryRef href="summary1.dita"/>
    </learningObject>
</map>
```

Learning Object DITA 1.3 Example

- In DITA 1.3, you can wrap a learning object with learningObjectMap
- Store it as a separate file
- Then reference it from many maps using learning ObjectMapRef
- Similar approach can also be used for learningObjectGroupMap



Main Learning Map

```
<map id="LearningMap2" title="Learning Map 2">
     <title>DITA 1.3 Learning and Training Sample</title>
     <learningObjectMapRef href="learningObjectMapA.ditamap"
                           navtitle="Learning Object Map A">
          <topicmeta/>
     </learningObjectMapRef>
     <learningObjectMapRef href="learningObjectMapB.ditamap"
                           navtitle="Learning Object Map B">
          <topicmeta/>
     </learningObjectMapRef>
</map>
```

Learning Object Map

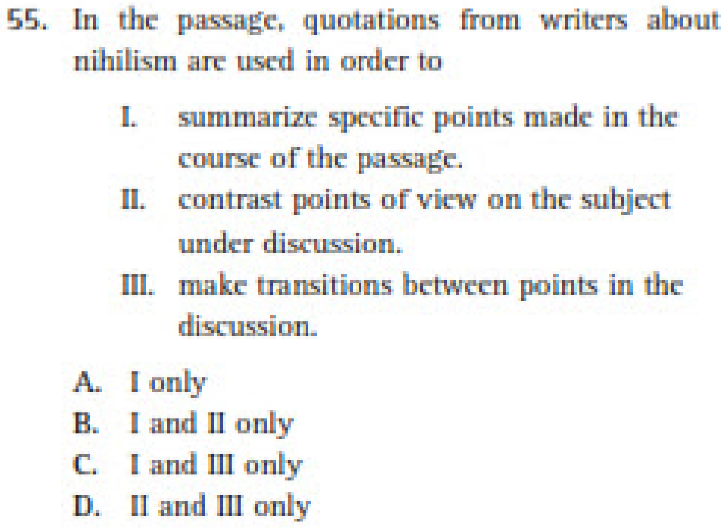
```
<learningObjectMap id="LearningObjectMapA">
     <title>DITA 1.3 Learning Object Map</title>
     <topicmeta/>
     <learningObject navtitle="DITA 1.3 Learning Object A">
          <learningOverviewRef href="overview1.dita"/>
          <learningContentRef href="content1.dita"/>
          <learningSummaryRef href="summary1.dita"/>
     </learningObject>
</learningObjectMap>
```

Standardize Learning Object and Group Referencing

- Standardize method for referencing learning objects and groups
- Encourage reuse of learning objects and groups within and across organizations and companies

New Feature: Improve Learning and Training Interactions

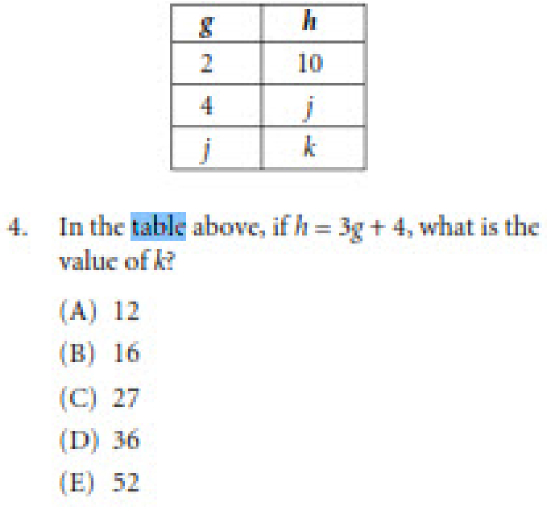
- Provide more robust content models for interaction (assessment) question statements, answer values, and feedback (allows multiple block elements).
- Examples (not easily possible in DITA 1.2):



55. In the passage, quotations from writers about nihilism are used in order to

    I. summarize specific points made in the course of the passage.
    II. contrast points of view on the subject under discussion.
    III. make transitions between points in the discussion.

    A. I only
    B. I and II only
    C. I and III only
    D. II and III only



g	h
2	10
4	j
j	k

4. In the table above, if $h = 3g + 4$, what is the value of k?

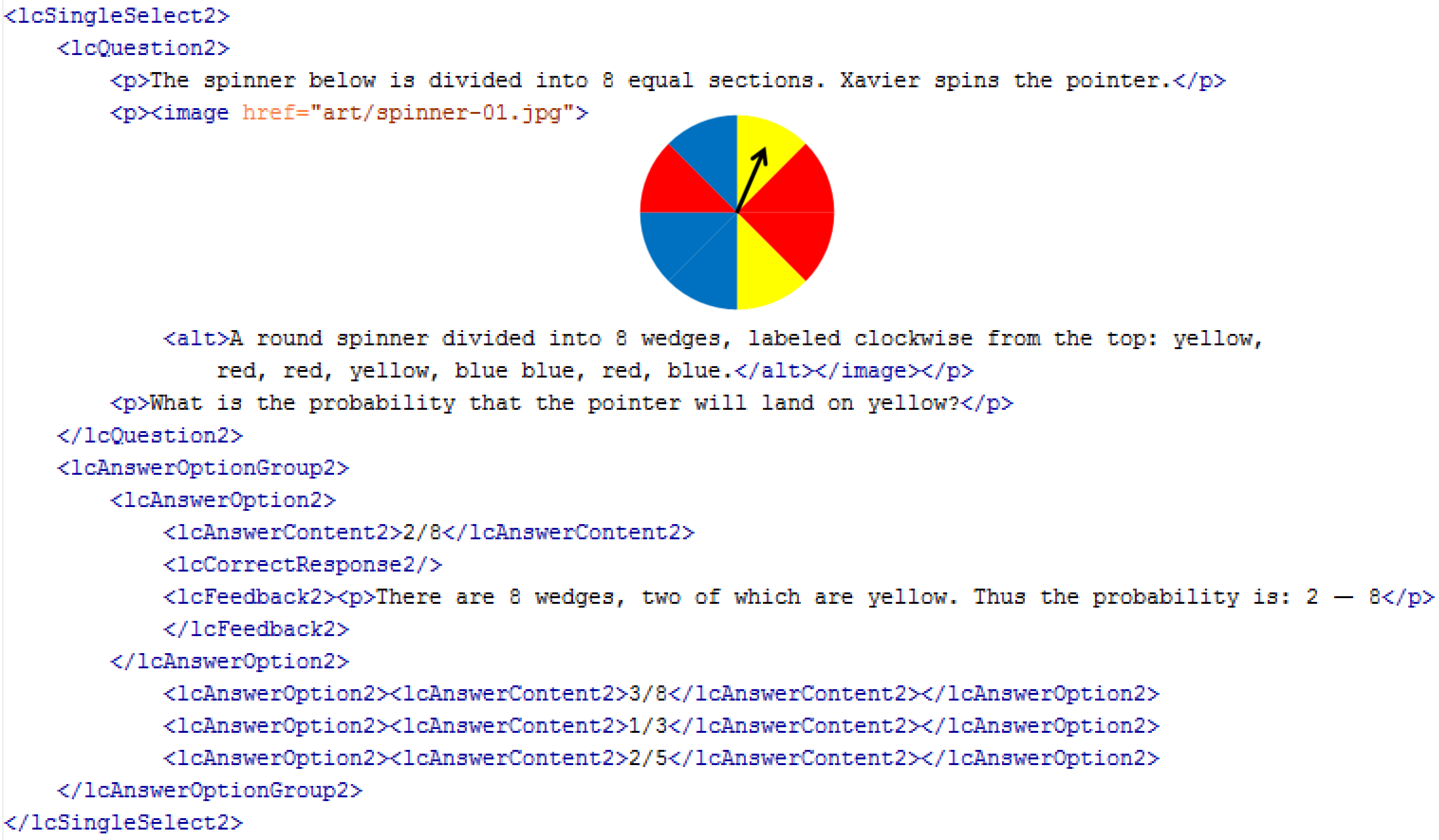
   (A) 12
   (B) 16
   (C) 27
   (D) 36
   (E) 52

Improve Learning and Training Interactions

- The available semantic elements in DITA 1.2 in assessment questions, answer values, and feedback blocks were unnecessarily restrictive.
- Define two new domains for the Learning and Training Specialization:
  - interactionBase2Domain
  - learning2Domain
- New domains provide a base model that allows multi-block element questions, answers, and feedback
- New domains are otherwise semantically identical to the original domains
- Original domains are still available

Improve Learning and Training Interactions

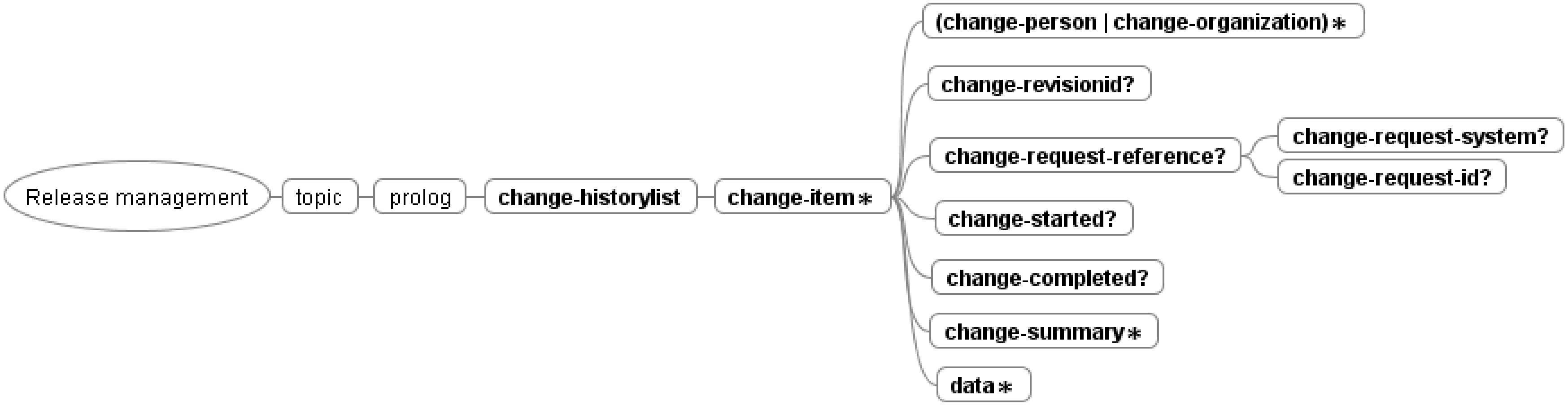
- Enables more robust interaction question statements, answers, and feedback
- Enables more engaging assessments, ideal for electronic learning deliveries like e-learning, mobile learning, and so on



```
<lcSingleSelect2>
    <lcQuestion2>
        <p>The spinner below is divided into 8 equal sections. Xavier spins the pointer.</p>
        <p><image href="art/spinner-01.jpg">
            <alt>A round spinner divided into 8 wedges, labeled clockwise from the top: yellow,
                red, red, yellow, blue blue, red, blue.</alt></image></p>
        <p>What is the probability that the pointer will land on yellow?</p>
    </lcQuestion2>
    <lcAnswerOptionGroup2>
        <lcAnswerOption2>
            <lcAnswerContent2>2/8</lcAnswerContent2>
            <lcCorrectResponse2/>
            <lcFeedback2><p>There are 8 wedges, two of which are yellow. Thus the probability is: 2 — 8</p>
            </lcFeedback2>
        </lcAnswerOption2>
            <lcAnswerOption2><lcAnswerContent2>3/8</lcAnswerContent2></lcAnswerOption2>
            <lcAnswerOption2><lcAnswerContent2>1/3</lcAnswerContent2></lcAnswerOption2>
            <lcAnswerOption2><lcAnswerContent2>2/5</lcAnswerContent2></lcAnswerOption2>
    </lcAnswerOptionGroup2>
</lcSingleSelect2>
```

Release management

Tom Cihak, Freescale
DITA Technical Communications Subcommittee, r65612@freescale.com



Release management: Problem

- **Use case 1:**
  - Our 2000-page manual is being updated. How can I alert my existing readers to the significant changes?
- **Use case 2:**
  - A regulatory agency requires that we record the date and time changes are made *down to the second*.

Release management: Solution

- An optional domain to enable authors to log significant changes between document revisions
  - Based on bookmap/bookchangehistory.
  - Comments and metadata (release notes) logged in the topic or map.
  - Elements only—no processing specified. XQuery used to gather changes.
  - Filtering can be by date *and* using select attributes, giving users maximum flexibility.

Release management: Elements

Element	Description
<change-item>	A single release note
<change-person>	Person making change
<change-organization>	Agency requiring change
<change-revisionid>	ID associated with change
<change-request-reference>	Wrapper element
<change-request-system>	Tracking system
<change-request-id>	ID associated with tracking system
<change-started	Date change was completed
<change-completed>	Date change was completed
<change-summary>	Description of change

Release management: Example

```
<prolog>
<!-- ... -->
  <changehistory-list>
    <change-item product="productA productB">
      <change-person>Tom Cihak</change-person>
      <change-completed>2014-03-23</change-completed>
      <change-summary>Made change 1</change-summary>
    </change-item>
    <change-item product="productA">
      <change-person>Tom Cihak</change-person>
      <change-organization>JEDEC</change-organization>
      <change-completed>2014-06-07</change-completed>
      <change-summary>Made change 2</change-summary>
    </change-item>
  </changehistory-list>
<!-- ... -->
</prolog>
```

Release management: Summary

- **Benefit for content workers**
  - Easier logging of changes—no external documents needed
  - Potentially increased compliance with standards
    - Changes are more likely to be made if it's easier
- **Benefit for readers**
  - Improved accuracy of change reporting
- **Miscellaneous benefits**
  - Facilitates Wikipedia-like tabbed display
  - Release notes can be edited

DITA Domains

Nancy Harrison
Infobridge Solutions
nharrison@infobridge-solutions.com

Secretary, DITA Technical Committee

New domains for DITA 1.3

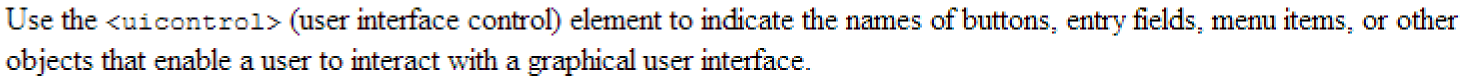
- Domains define markup for a particular subject area. DITA 1.3 provides a number of new domains:
  - Domains to extend available semantic markup:
    - XML mention
    - Release management (Tom Cihak)
    - Branch filtering (Robert Anderson)
  - Domains that formalize how DITA references non-DITA objects:
    - Mathematics: MathML & Equation domains
    - Graphics: SVG domain

XML mention

- **Purpose and benefit:** A way to mark up mentions of XML syntactic components such as element and attribute.
  - Can be used in the DITA standard
  - Documentation for XML applications
  - Training materials for XML technologies

- **Example:**

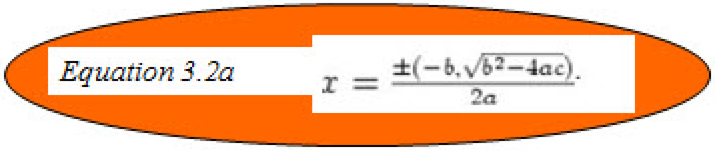
```
<p>Use the <xmlelement>uicontrol</xmlelement> (user
interface control) element to indicate the names of buttons,
entry fields, menu items, or other objects that enable a
user to interact with a graphical user interface.</p>
```



Use the `<uicontrol>` (user interface control) element to indicate the names of buttons, entry fields, menu items, or other objects that enable a user to interact with a graphical user interface.

MathML & Equation domains

- **Purpose and benefit:**
  - Provides containers for equations and an integration with the MathML standard
  - Enables direct use of MathML markup within DITA documents
  - Enables use-by-reference of MathML markup held in separate non-DITA documents
- **Example:**



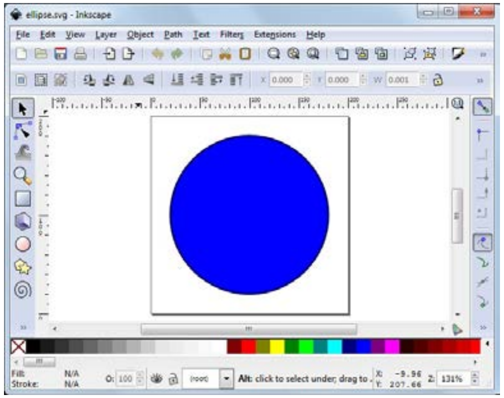
Equation 3.2a $x = \frac{\pm(-b,\sqrt{b^2-4ac})}{2a}.$

```
<p>A block equation using an image:
   <equation-block id="eq-3.2a">
      <equation-number>3.2a</equation-number>
      <image keyref="equation-image-01">
          <alt>a squared plus b squared.</alt>
      </image>
   </equation-block> </p>
```

Graphics: Scalable Vector Graphic domain

- **Purpose and benefit:**
  - Provides the ability to include as content the <svg:svg> element from the SVG 1.1 vocabulary
  - Enables direct use of SVG markup within DITA documents
  - Enables use-by-reference of SVG markup held in separate non-DITA documents.
- **Example:**

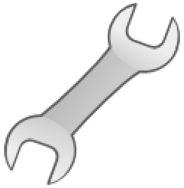
```
<p>SVG Directly in body:</p>
<svg_container>
  <svg:svg width="200" height="200">
      <svg:ellipse cx="100" cy="100"
                   rx="80" ry="80"
         style="fill:blue; stroke:rgb(0,0,100);stroke-width:2"/>
  </svg:svg>
</svg_container>
```



DITA 1.3 Troubleshooting features

Bob Thomas
Tagsmiths, LLC

OASIS DITA TC, TechComm SC, DITA Adoption TC



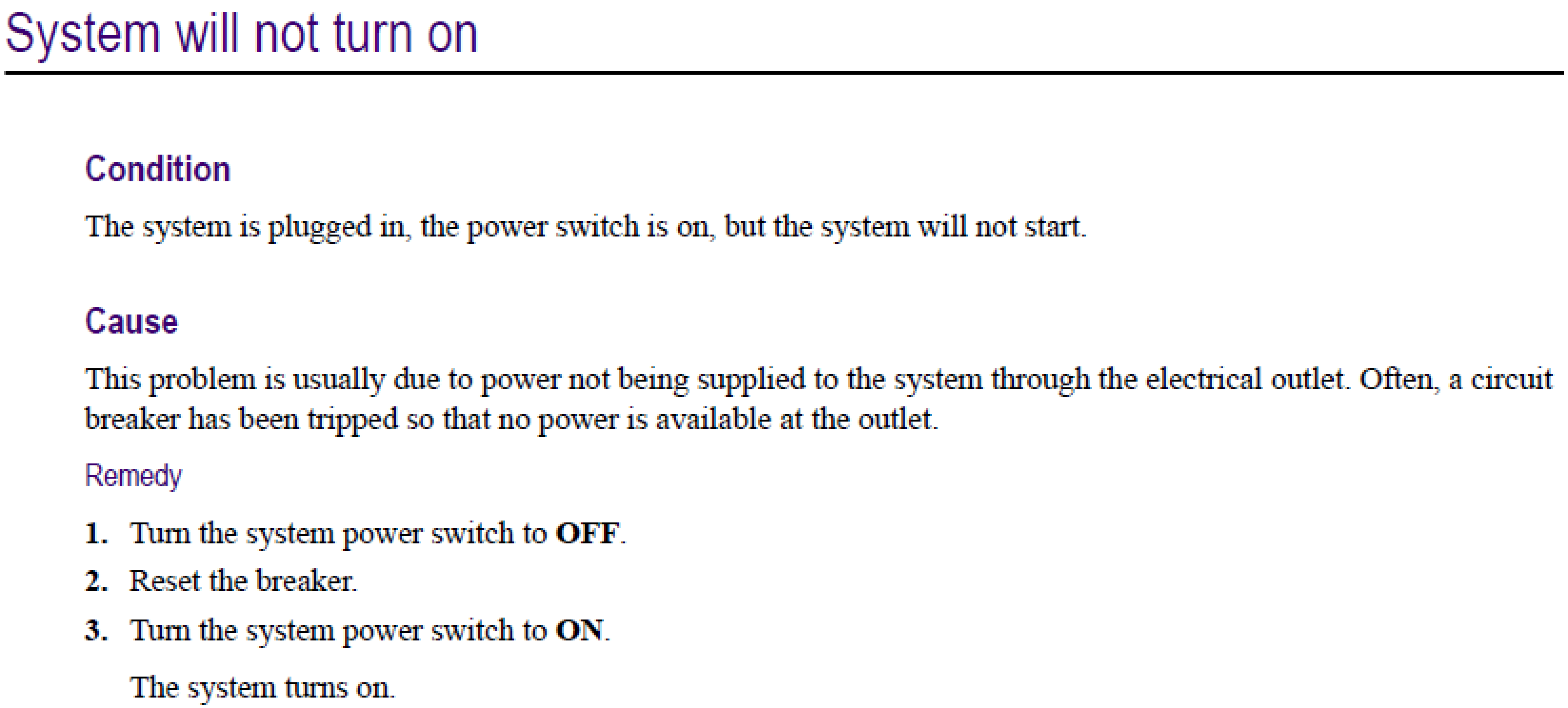
Trouble?

The new troubleshooting topic type and the new troubleshooting elements can help

bob.thomas@tagsmiths.com

Sample troubleshooting topic

Output:



System will not turn on

Condition

The system is plugged in, the power switch is on, but the system will not start.

Cause

This problem is usually due to power not being supplied to the system through the electrical outlet. Often, a circuit breaker has been tripped so that no power is available at the outlet.

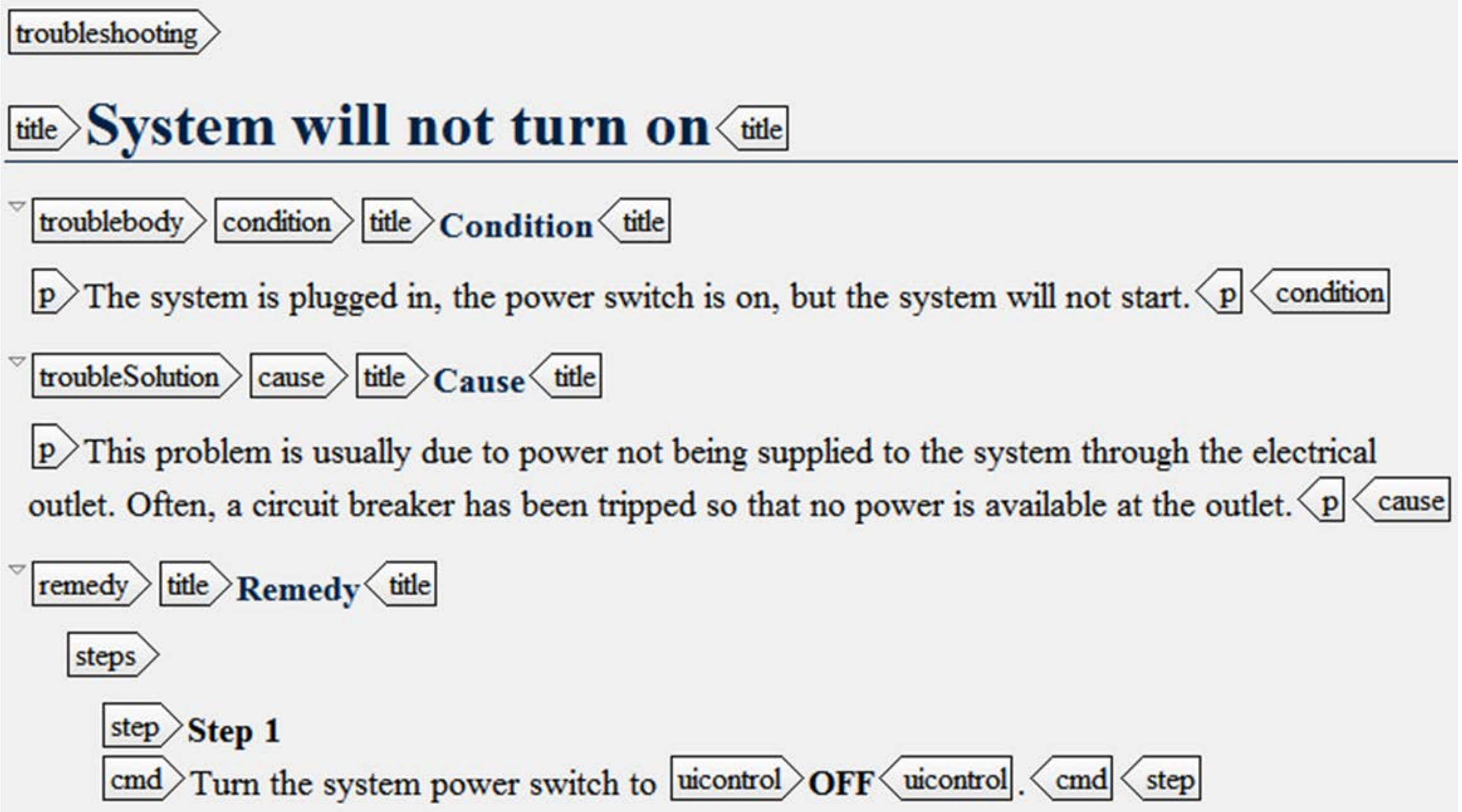
Remedy

1. Turn the system power switch to **OFF**.
2. Reset the breaker.
3. Turn the system power switch to **ON**.

   The system turns on.

Troubleshooting topic type

Markup:



troubleshooting

title System will not turn on title

troublebody condition title Condition title

p The system is plugged in, the power switch is on, but the system will not start. p condition

troubleSolution cause title Cause title

p This problem is usually due to power not being supplied to the system through the electrical outlet. Often, a circuit breaker has been tripped so that no power is available at the outlet. p cause

remedy title Remedy title

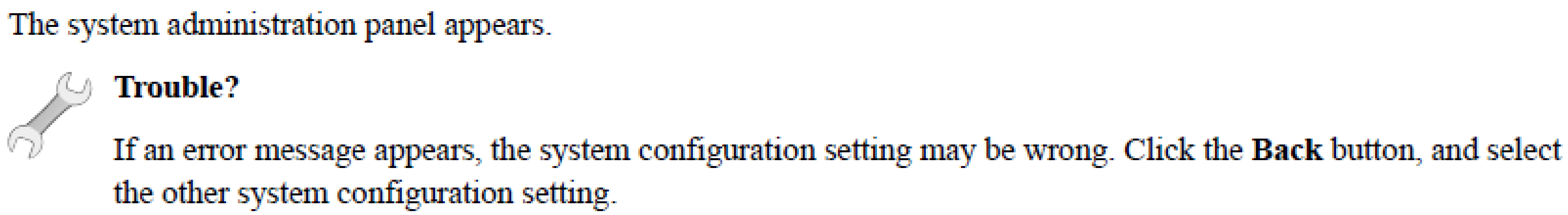
steps

step Step 1

cmd Turn the system power switch to uicontrol OFF uicontrol . cmd step

Step troubleshooting element

Output:

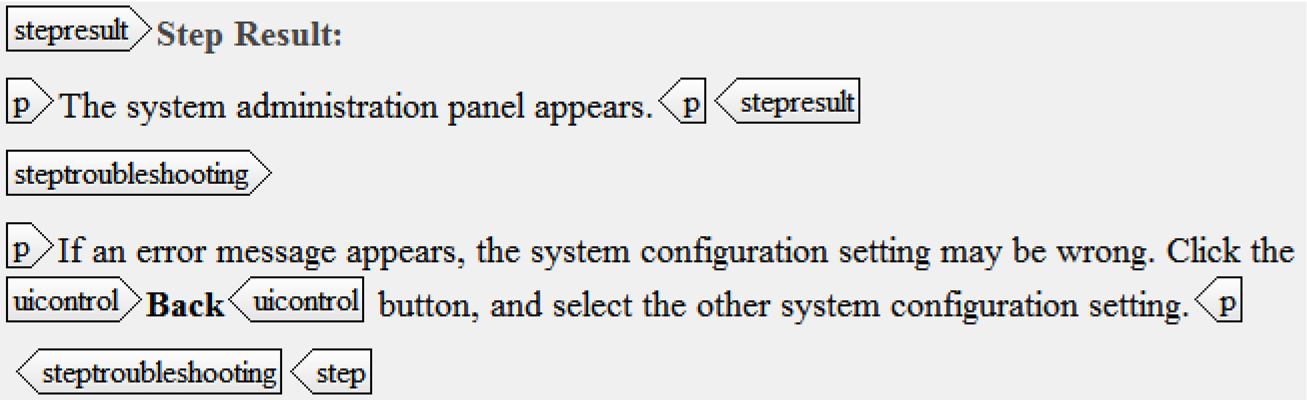


The system administration panel appears.

Trouble?

If an error message appears, the system configuration setting may be wrong. Click the **Back** button, and select the other system configuration setting.

Markup:



stepresult **Step Result:**

p The system administration panel appears. p stepresult

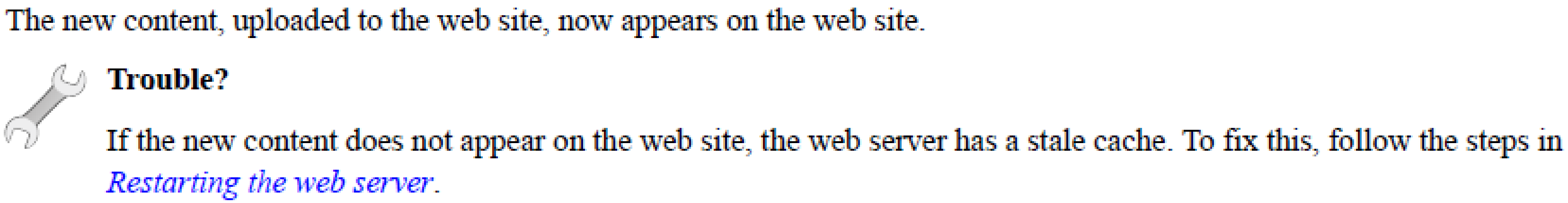
steptroubleshooting

p If an error message appears, the system configuration setting may be wrong. Click the uicontrol **Back** uicontrol button, and select the other system configuration setting. p

steptroubleshooting step

Task troubleshooting element

Output:

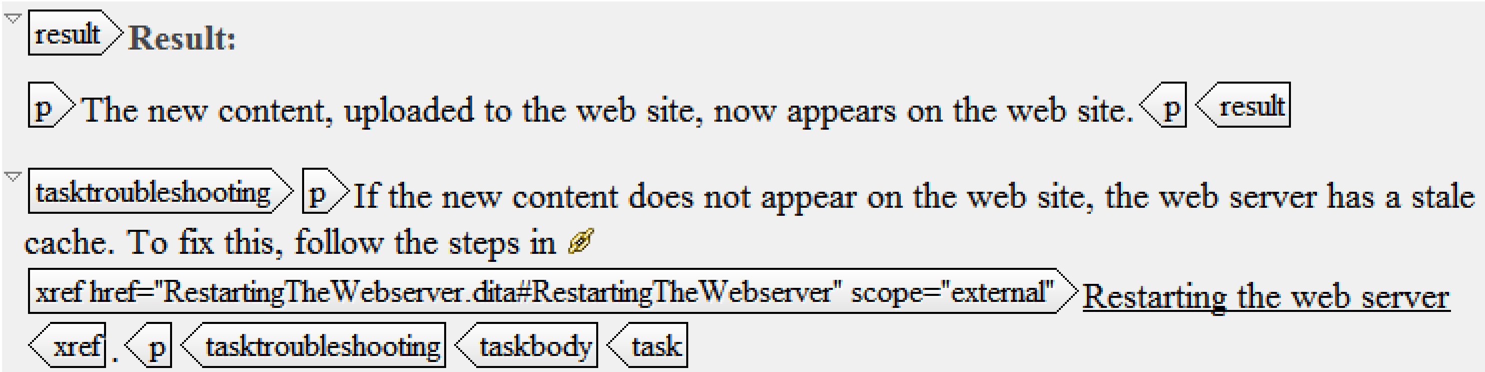


The new content, uploaded to the web site, now appears on the web site.

Trouble?

If the new content does not appear on the web site, the web server has a stale cache. To fix this, follow the steps in *Restarting the web server*.

Markup:



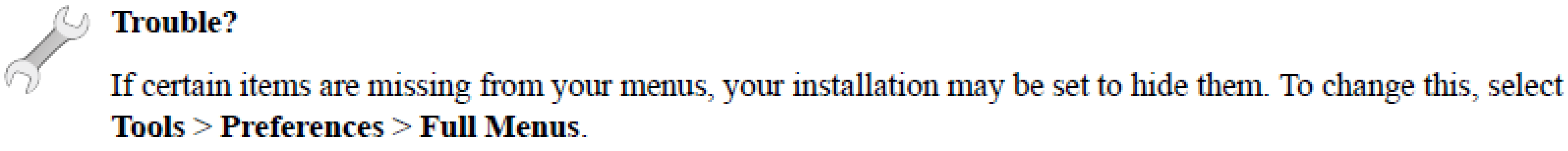
```
result Result:

p The new content, uploaded to the web site, now appears on the web site. p result

tasktroubleshooting p If the new content does not appear on the web site, the web server has a stale
cache. To fix this, follow the steps in
xref href="RestartingTheWebserver.dita#RestartingTheWebserver" scope="external" Restarting the web server
xref . p tasktroubleshooting taskbody task
```

Trouble note type

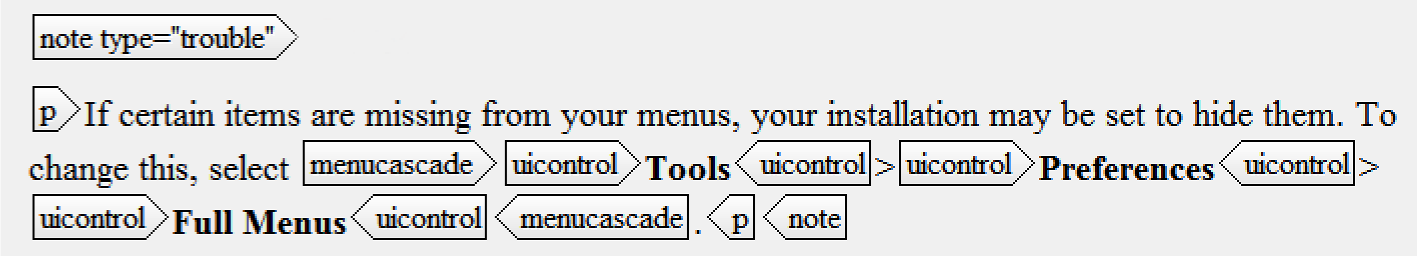
Output:



Trouble?

If certain items are missing from your menus, your installation may be set to hide them. To change this, select **Tools** > **Preferences** > **Full Menus**.

Markup:



```
<note type="trouble">
<p>If certain items are missing from your menus, your installation may be set to hide them. To change this, select <menucascade><uicontrol>Tools</uicontrol>><uicontrol>Preferences</uicontrol>><uicontrol>Full Menus</uicontrol></menucascade>.</p></note>
```

Why troubleshooting was added

- DITA 1.2 has no formal semantic support for troubleshooting information
- Many DITA adopters addressed this need through specialization
- Adding troubleshooting to DITA 1.3 provides a consistent set of semantics and common support for troubleshooting
- Troubleshooting markup supports other corrective-action scenarios such as alarm-clearing

Troubleshooting feature benefits

- Easy to create information design-patterns for most types of problem solving
- <steptroubleshooting> and <tasktroubleshooting> provide a place for immediate corrective action in task
- The “trouble” note type provides a place for corrective action wherever <note> is allowed.
- <step> content from <task> can be re-used in <troubleshooting> topics
- Authors see troubleshooting semantics while writing, making it easier to create consistent and tightly focused troubleshooting information

Troubleshooting — more information please

- Read my OASIS feature article on troubleshooting:

  https://www.oasis-open.org/committees/download.php/53516/DITA13TroubleshootingArticle_FINAL_03jul14.pdf

Questions?

Summary: DITA 1.3 (1st Q 2015)

- Information types
  - Troubleshooting; L & T
- Architecture
  - Key scopes and cross-deliverable referencing
  - Branch filtering; expanded syntax for filtering
  - Support for RELAX NG; reuse of structural specifications
- Domains:
  - DITAvalref; markup and XML mention; scalable vector graphics (SVG)
  - Equation and MathML; release management; learning2
- New elements and attributes:
  - Troubleshooting elements; user assistance elements
  - Additions to highlighting domain; <div>