



Introduction to the DITA architecture

ACM SIGDOC Structured Authoring Committee

Stanley Doherty, Ph.D.
SJDoherty.ACM@gmail.com

<https://acm-sigdoc-structured.org/1-curriculum-resources.html>



**Association for
Computing Machinery**

Advancing Computing as a Science & Profession

Agenda

1. Challenges to industry/academic collaboration.
2. What is structured content?
3. What are DITA topics?
4. What are DITA maps?
5. How is working in DITA different from working in a word processor?
6. Demo – what does content look like in DITA?



Who is this guy?



Education

- B.A. English (Merrimack)
- M.A. Linguistics (Northeastern)
- Ph.D. English (Harvard)

Software documentation

- Lotus Development (1987-97)
- Parametric Technologies (1997-99)
- Akamai Technologies (2012-23)
- Mathworks (2013-14)
- HPE (2017)

Hardware documentation

- Sun Microsystems (1999-2008)
- Verivue (2008-2012)
- SimpliVity (2015-17)

Cloud documentation

- Oracle (2017-2022)
- Google LLC (2022-2025)

Industry roles

- Individual contributor (IC):
 - > TW, Information Architect, Consulting Engineer
- Manager:
 - > TW manager, Director, Sr. Director

Teaching

- Northeastern University (TA)
- Harvard University (Teaching Fellow)
- Bentley University (Asst. Professor)

Avocations

- [OASIS DITA](#)
- [ACM SIGDOC](#)
- [IEEE Transactions](#)
- [Boston DITA UG](#)
- [Crimson Courage](#)
- [ARC of MA](#)
- [SensusFidelium.org](#)
- [ModularWriting.com](#)



Challenges to Industry/Academic Collaboration

What are the things that we can influence?

- Industry/academic engagement:
 - Conferences
 - Professional organizations
- Curriculum-oriented resources:
 - Content about structured content
 - Free or low-cost tools for authoring
 - Relevant learning paths and teaching plans
 - Relevant sample content
- Consulting and support:
 - Implementation advice (free consulting)
 - Industry speaker/visitor bureau (ZOOM visits)
- Shared industry/academic vocabulary
- Shared industry/academic review processes



Challenges to Industry/Academic Collaboration

What is mostly beyond our influence?

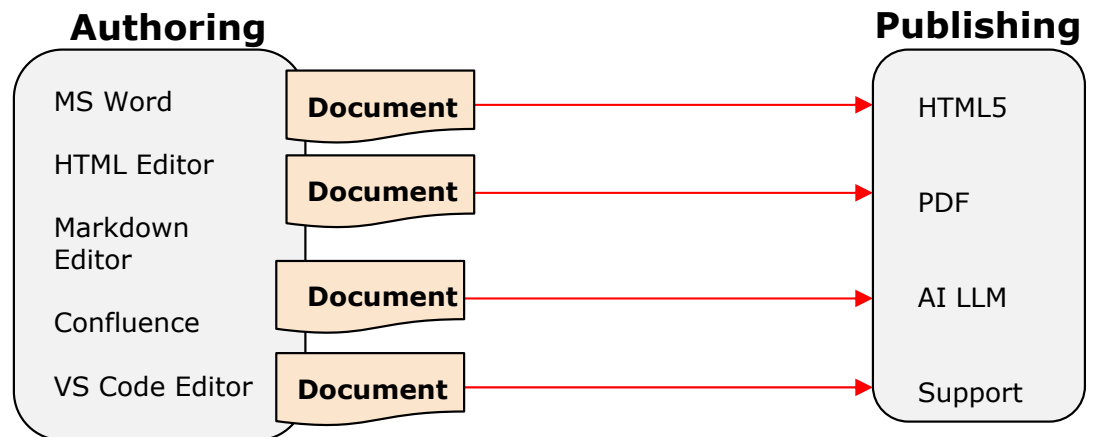
- Institutional infrastructure restrictions:
 - No software downloads
 - No IT support
 - No discretionary budgets for tech comm pilots
- The number of college programs on technical communication:
 - 70% reduction over the past 15 years
 - Lack of tenure attenuation
- Research collaboration and cross-over:
 - Industry research focuses on internal-only work (not public)
 - Academic research not accessible to industry people
- Professional opportunities for entry-level writers:
 - Few (if any) internships in technical writing
 - Few (if any) college hiring budgets
- AI/Markdown FUD



What is structured content?

Big picture

Word processors and unstructured editors provide a unified environment for authoring and publishing individual documents.



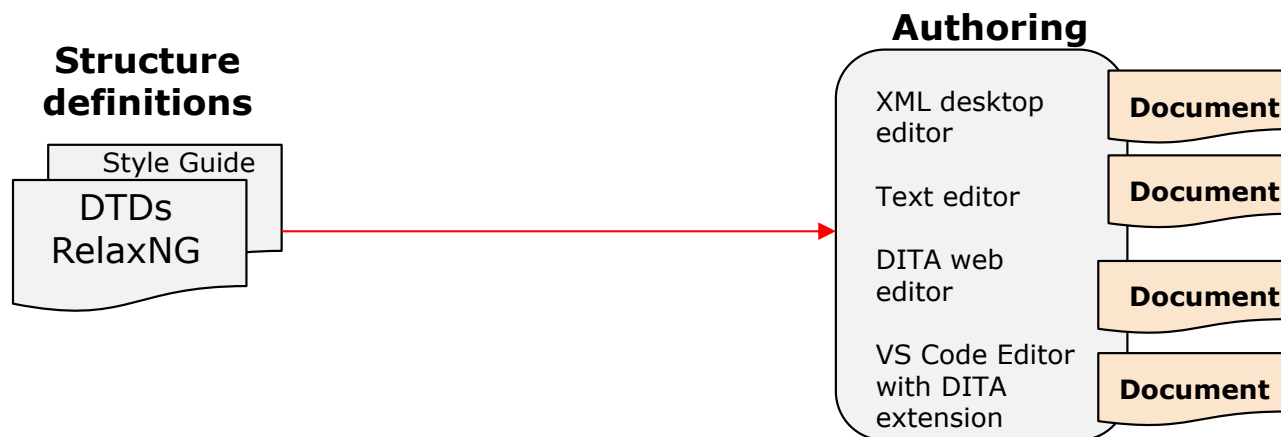
What is structured content?

Big picture

Structured content depends on formal definitions for content types, content hierarchy, and semantic elements required for a valid XML, database, JSON, or DITA document. DITA documents declare that they use XML syntax and instances of a particular DTD.

```
<?xml version="1.0" encoding="UTF-8"?>
```

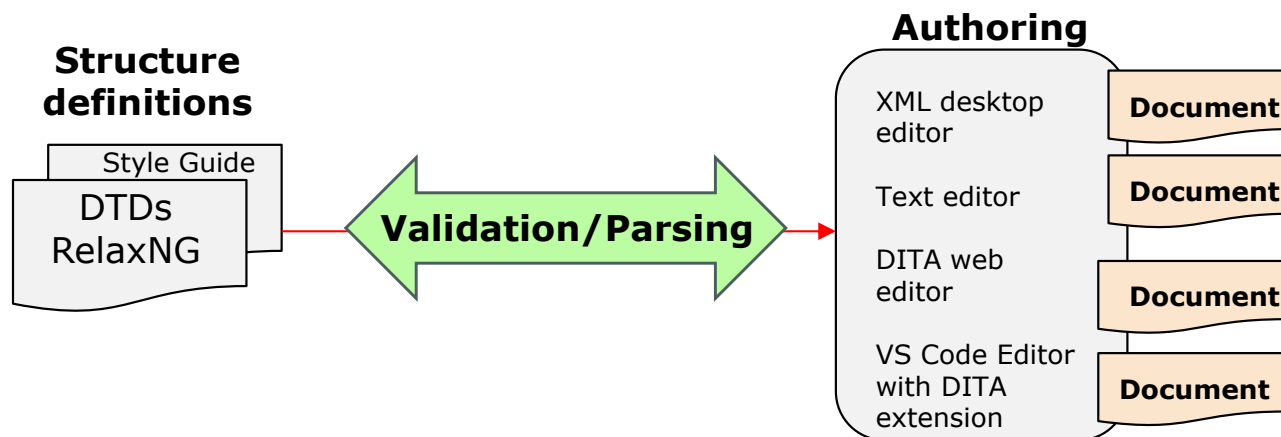
```
<!DOCTYPE concept PUBLIC "-//OASIS//DTD DITA 2.x Concept//EN" "concept.dtd">
```



What is structured content?

Big picture

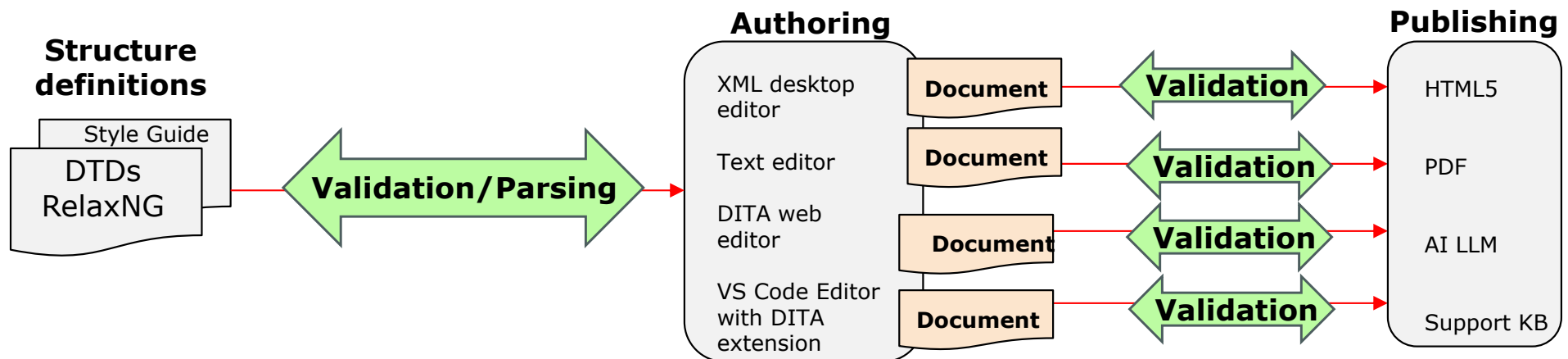
DITA editors typically have integrated XML parsers. As we enter content in the editor, that content gets reviewed by the integrated parser. If the content complies with the DTD declared in the document, the content is **valid**. If the content deviates from the DTD, the parser evaluates the content as **invalid** and (typically) suggests corrections.



What is structured content?

Big picture

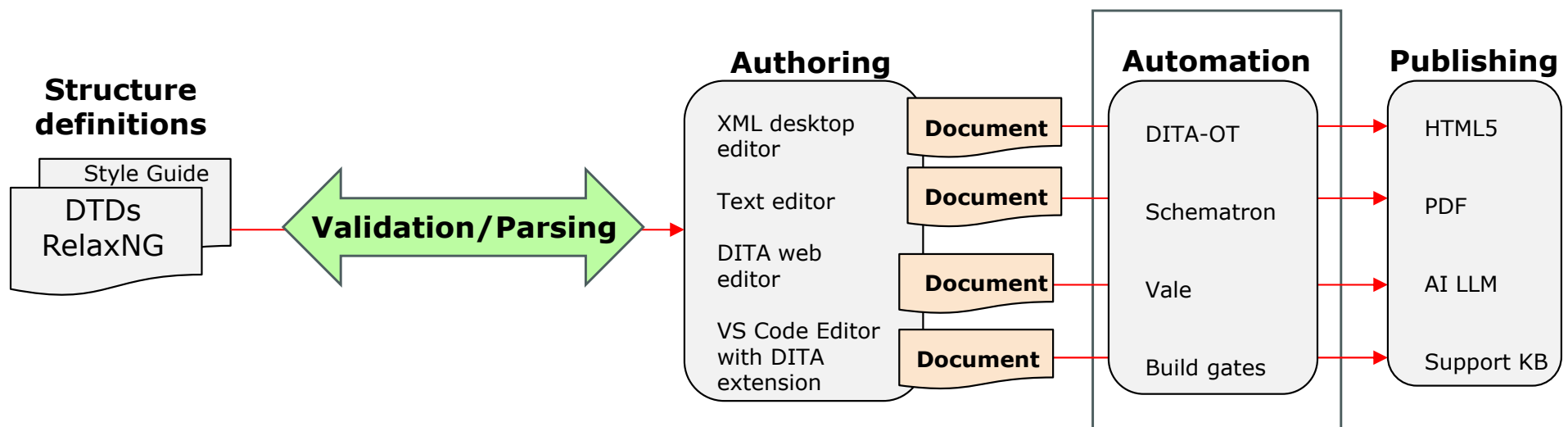
When you want to publish content, you submit it to a separate publishing engine (processor). Because the DITA editor and the DITA processor validate against the same DTDs, the content is validated for a second time before it is published. A thousand DITA concept topics will have the same structure.



What is structured content?

Big picture

Valid structured content is consistent, making it ideal for automated style checking, content management, build processing, AI ingestion, and dynamic delivery. The same validation and automation that works for 20 topics, scales for 1,000,000.



What are DITA topics?

A DITA topic is the basic, modular unit of authoring, consisting of a self-contained XML file focusing on one subject. It typically includes a title and content, categorized by type — usually concept, task, or reference — to ensure consistency and maximize reuse across different publications.

Out-of-the-box DITA 2.0 supports six popular topic types.

Type	Answers	DTD
Concept	What is this?	concept.dtd
Task	How do I use this?	task.dtd
Reference	What are the details?	reference.dtd
General	TBD	topic.dtd
Troubleshooting	How do I fix this?	troubleshooting.dtd
Glossary	What terms are relevant?	glossaryentry.dtd

NOTE: Any of these OOTB topic types can be specialized.

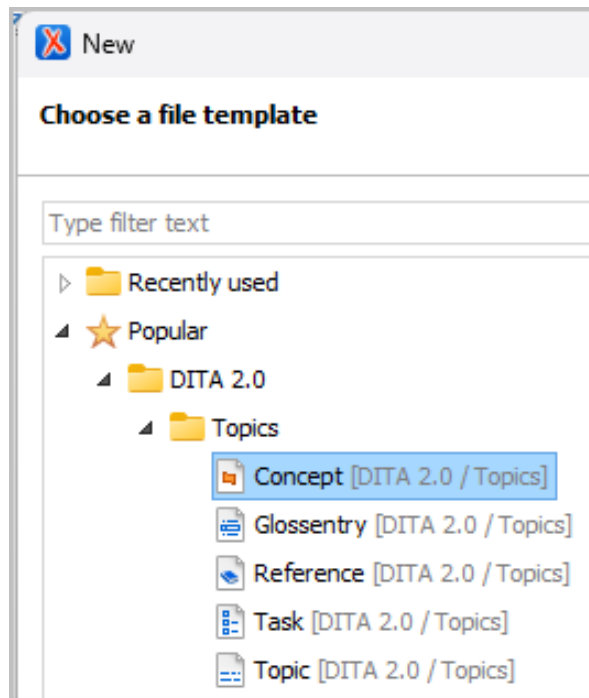


Association for
Computing Machinery

Advancing Computing as a Science & Profession

What are DITA topics?

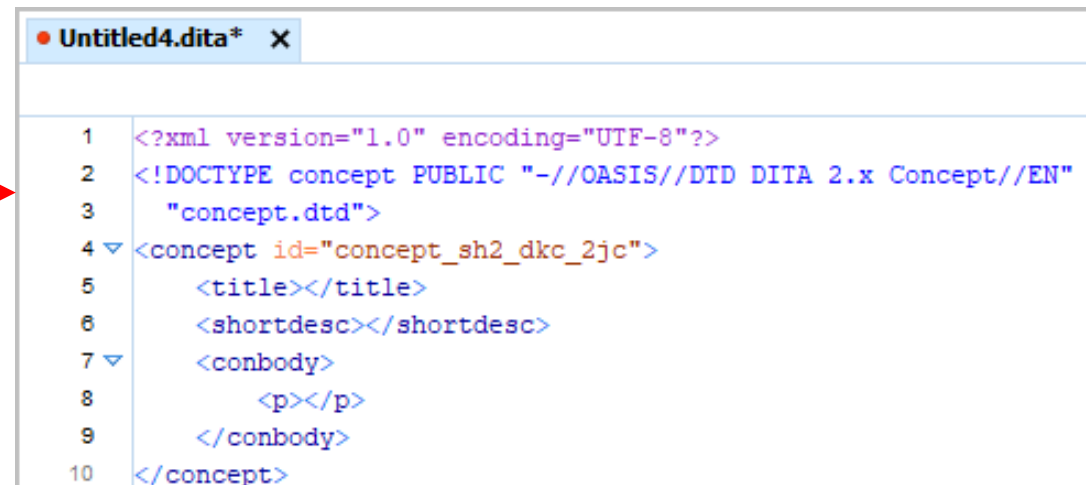
Creating a particular type of DITA topic is easy in a DITA editor.



View in
"tags" mode



View in
"text" mode



What are DITA topics?

All DITA markup is XML and consists of:

XML elements: `<xref>`, `<note>`

XML attributes: `<xref href="topic1.dita"/>`
`<note type="caution">`

XML attribute values: `<xref href="topic1.dita"/>`
`<note type="caution">`

HTML-style comments: `<!-- This is a comment -->`



What are DITA topics?

All DITA topics offer optional elements in the same order.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE concept PUBLIC "-//OASIS//DTD DITA 2.x
  Concept//EN" "concept.dtd">
```

```
<concept id="concept_sh2_dkc_2jc">
  <title></title>
```

Short description →

```
<shortdesc></shortdesc>
```

Prolog with metadata →

```
<prolog>
  <metadata>
    <prodinfo>
      <prodname></prodname>
    </prodinfo>
  </metadata>
</prolog>
```

Topic body container →

```
<conbody>
  <p></p>
</conbody>
```

Related links →

```
<related-links>
  <link href="topic1.dita"/>
</related-links>
```

```
</concept>
```



What are DITA topics?

Writers use the same 16 body elements most frequently (95%)

<code><p></code>	(Paragraphs)
<code><section></code>	(Organizing body content)
<code></code>	(Unordered lists)
<code></code>	(List items)
<code><xref></code>	(Cross-references)
<code><image></code>	(Graphics and screenshots)
<code><table></code>	(Structured tabular data)
<code><entry></code>	(Table cell)
<code><row></code>	(Table row)
<code><fig></code>	(Figure wrapper)
<code><note></code>	(Warnings, tips, notices)
<code><codeph></code>	(Inline code phrase)
<code><ph></code>	(Generic inline phrase)
<code><keyword></code>	(Semantic keyword markup)
<code><cmd></code>	(Task step command)
<code><step></code>	(Task procedure step)
<code><steps></code>	(Container for task steps)



What are DITA topics?

Semantic markup is everywhere! The name of the element describes its content. Great for AI ingestion engines.

Block elements

- <shortdesc>
- <prereq>
- <context>
- <steps>
- <cmd>
- <codeblock>
- <stepresult>
- <result>
- <postreq>
- <msgblock>
- <syntaxdiagram>
- <systemoutput>
- <audio>
- <video>

Phrase elements

- <cite>
- <codeph>
- <cmdname>
- <menucascade>
- <uicontrol>
- <filepath>
- <msgph>
- <userinput>
- <apiname>
- <equation-inline>
- <partno>



What are DITA topics?

DITA is optimized for content reuse-by-reference, not copying. Any element with a unique @id attribute value can be referenced by other elements.

Source markup with @ids

```
<ph id="ph_product1">Product name #1</ph>
```

```
<section id="section_new-restrictions">. . .</section>
```

```
<fig id="fig_network-topography">. . . </fig>
```

```
<codeblock id="codeblock_python-create-object">. . . </codeblock>
```

Content references

```
<p>Buy <ph conref="lib.dita#lib-id/ph_product1"/></p>
```

```
<section conref="lib.dita#lib-id/section_new-restrictions"/>
```

```
<fig conref="lib.dita#lib-id/fig_network-topography"/>
```

```
<codeblock conref="lib.dita#lib-id/codeblock_python-create-object"/>
```



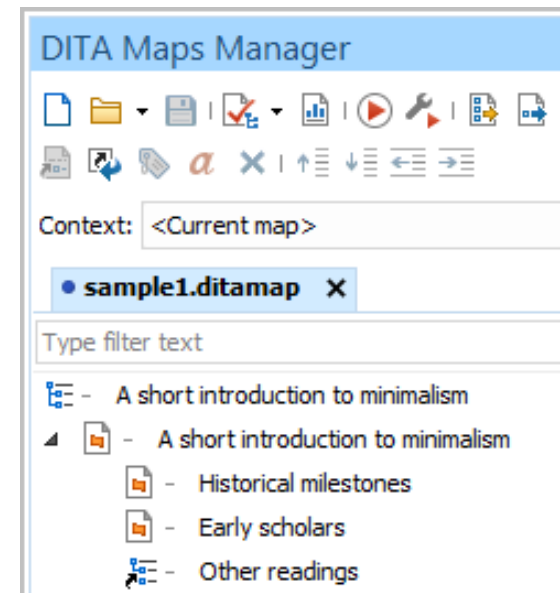
What are DITA maps?

DITA maps are XML documents that reference subordinate topics, maps, and resources. Topics provide content for a publication; maps provide the "logic" – topic sequence, hierarchy, metadata, and publication-related information.

When viewed as text . . .

```
sample1.ditamap X
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE map PUBLIC "-//OASIS//DTD DITA 2.x Map//EN"
3   "map.dtd">
4 <map>
5   <title>A short introduction to minimalism</title>
6   <topicref href="introduction.dita">
7     <topicref href="history.dita"/>
8     <topicref href="early-writings.dita"/>
9     <mapref href="related-topic-list.ditamap"/>
10  </topicref>
```

When viewed in a DITA editor . . .



When processed into a PDF . . .

Contents	
Chapter 1. A short introduction to minimalism.....	3
Historical milestones.....	3
Early scholars.....	3

What is the DITA architecture?

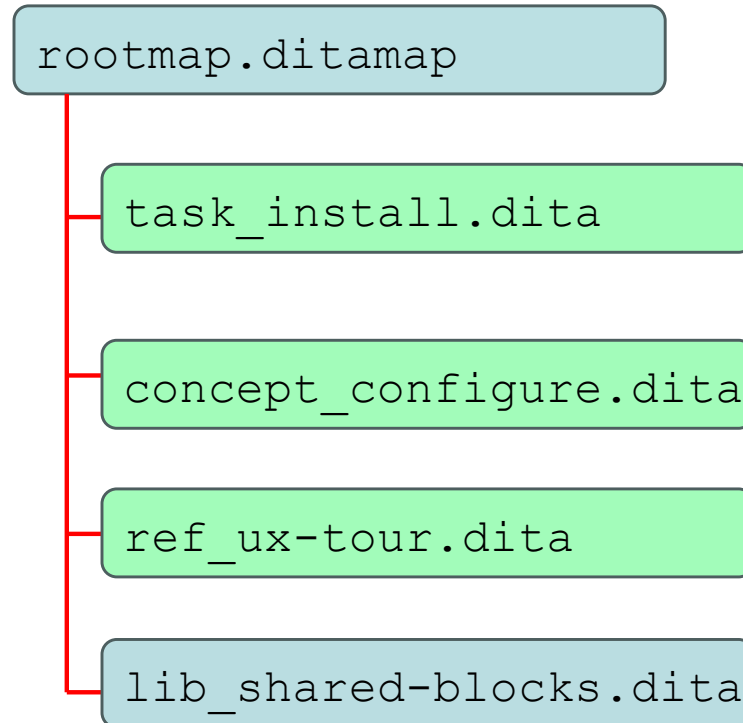
Maps provide logic and reference child topics

Topics focus on specific info types

Topics have similar structures

Topics support semantic markup

Topics support content reuse



What is the DITA architecture?

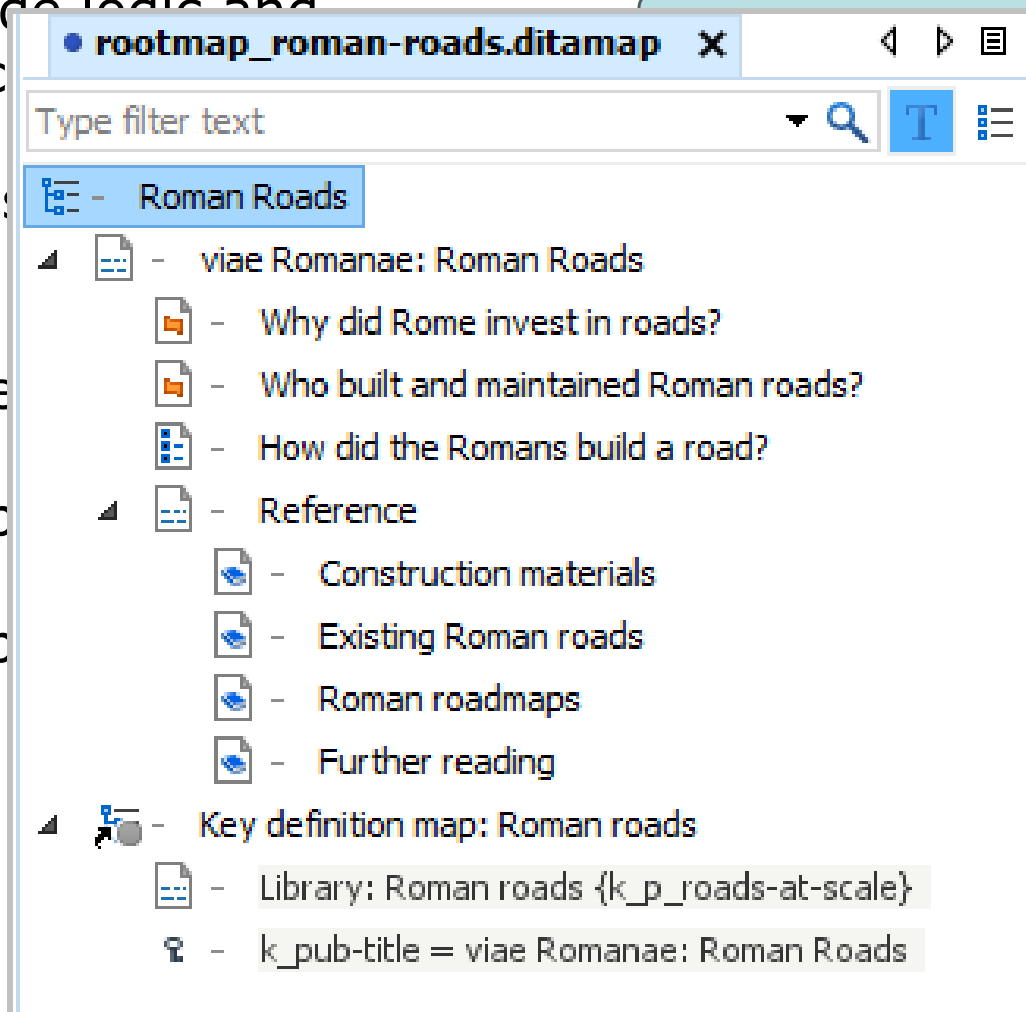
Maps provide logic and reference content

Topics focus on content types

Topics have

Topics support

Topics support



How is working in DITA different from working in a word processor (Word, GDoc)?



Feature	Word processor	DITA (OOTB)
Overhead	Low	Moderate
Authoring app	WP app	DITA editor
Review app	WP app	GitHub / Web review
Publishing app	WP app	DITA Open Toolkit
Artifact(s)	One document	Multiple topics and maps
Document format	Proprietary Binary	Open standard Text
On-screen styling	WYSIWYG	Not WYSIWYG Suggestive, not real
Content styling	Integrated CSS	Separate processor transforms



How is working in DITA different from working in a word processor (Word, GDoc)?



Feature	Word processor	DITA (OOTB)
Content ownership	1 writer	1+ writer(s)
Images/multimedia	Embedded	Referenced
Build automation	Limited	Robust
Engineering integration	Limited	Robust
Version control	One document One control thread	Multiple documents Multiple threads
Content reuse	Copying	By reference
Cross-document linking	Limited	Robust
Semantic markup	Limited	Robust
Real-time validation	None	Robust
Enterprise scalability	None	Robust



Options to get started



If you have little or no exposure to DITA . . .

Review the ACM SIGDOC resources on “Structured Authoring” website.

If you have campus IT support for GitHub . . .

See Dr. Carlos Evia’s GitHub Actions plug-ins.

If students have access to VS Code and can download apps . . .

Download the DitaCraft extension and DITA-OT.

If you have no \$\$ and can download applications . . .

Download VS Code, DITA-OT, DitaCraft, or XML Mind.

If you have no \$\$ and need a solution for < 30 days . . .

Download the Oxygen Editor 30-day evaluation license.



DEMO – DEMO - DEMO



No-cost options

- DITA authoring in a text editor + the DITA-OT.
- DITA authoring in XMLMind + the DITA-OT.
- DITA authoring in VS Code + DitaCraft extension + the DITA-OT.
- DITA authoring in 30-day Oxygen Editor license.
- DITA authoring in Heretto CCMS web instance (ACM SIGDOC pilot).

Low-cost options

- DITA authoring in academic Oxygen Editor license (starting at \$72).

Text editor + DITA-OT



```
rootmap_roman-roads.ditamap
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE map PUBLIC "-//OASIS//DTD DITA 2.x Map//EN"
3    "map.dtd">
4  <map id="roman_roads_map">
5    <title>Roman Roads</title>
6    <topicref href="roman-roads-overview.dita">
7      <topicref href="roman-roads-concept.dita"/>
8      <topicref href="roman-roads_whobuilt.dita"/>
9      <topicref href="roman-roads-task.dita"/>
10     <topicref href="roman-roads_reference-glue.dita">
11       <topicref href="roman-roads_materials.dita"/>
12       <topicref href="roman-roads-reference.dita"/>
13       <topicref href="roman-roads_map.dita"/>
14       <topicref href="roman-roads_further-reading.dita"/>
15     </topicref>
16   </topicref>
17   <topicref href="keymap_libraries.ditamap"
18     format="ditamap" processing-role="resource-only"/>
19 </map>
```



Command window/shell

```
C:\>..\dita-ot-4.4\bin\dita \  
-i "C:\roman_roads_dita\rootmap_roman-roads.ditamap" \  
-f pdf \  
-o intro-roman-roads
```



Association for
Computing Machinery

Advancing Computing as a Science & Profession

NOTE: DITA tools run on Windows, Mac, and Linux.

XMLMind editor + DITA-OT



The screenshot shows the XMLMind editor interface. The menu bar includes File, Select, Edit, Search, View, Tools, XML, Tab, Options, and Help. The toolbar contains icons for file operations (New, Open, Save, Print, Copy, Paste, Undo, Redo) and editing (Find & Replace, Remark). The active file is 'rootmap_roman-roads.ditamap'. The editor content shows the following XML code:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE map PUBLIC "-//OASIS//DTD DITA 2.x Map//EN"
3   "map.dtd">
4 <map id="roman_roads_map">
5   <title>Roman Roads</title>
6   <topicref href="roman-roads-overview.dita">
7     <topicref href="roman-roads-concept.dita"/>
8     <topicref href="roman-roads_whobuilt.dita"/>
9     <topicref href="roman-roads-task.dita"/>
10    <topicref href="roman-roads_reference-glue.dita">
11      <topicref href="roman-roads_materials.dita"/>
12      <topicref href="roman-roads-reference.dita"/>
13      <topicref href="roman-roads_map.dita"/>
14      <topicref href="roman-roads_further-reading.dita"/>
15    </topicref>
16  </topicref>
17  <topicref href="keymap_lib
18    format="ditamap" process
19 </map>
```

The screenshot shows the 'Download DITA-OT' page on the website www.dita-ot.org/download. The page features a navigation bar with links for 'DITA Open Toolkit', 'Code', 'Community', 'DITA-OT Day', 'Docs', 'Download', and 'Plugins'. The main heading is 'Download DITA-OT'. Below the heading, it states 'Current version: DITA-OT 4.4 released January 31, 2026'. A green button with a download icon and the text 'DITA-OT 4.4' is visible. At the bottom, it says 'For details on the changes in this version, see the DITA-OT 4.4 Release Notes.'

Command window/shell

```
C:\>..\dita-ot-4.4\bin\dita \  
-i "C:\roman_roads_dita\rootmap_roman-roads.ditamap" \  
-f pdf \  
-o intro-roman-roads
```



VS Code + DitaCraft extension + DITA-OT



The screenshot shows the VS Code editor interface with the DitaCraft extension. The main editor window displays the XML output of a DITA map, including the root `<map>` element and several `<topicref>` elements. A tooltip is visible over the `<topicref>` tag, providing a description and attributes. The DITA Preview window on the right shows the rendered content, including the title "Roman Roads" and a list of links.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE map PUBLIC "-//OASIS//DTD DITA 2.x Map//EN"
3   "map.dtd">
4 <map id="roman_roads_map">
5   <title>Roman Roads</title>
6   <topicref href="roman-roads-overview.dita">
7     <topicref href="roman-roads-concept.dita"/>
8     <topicref href="roman-roads_whobuilt.dita"/>
9     <topicref href="roman-roads-task.dita"/>
10    <topicref href="roman-roads_reference-glue.dita">
11      <topicref href="roman-roads_materials.dita"/>
12      <topicref href="roman-roads_reference-glue.dita"/>
13      <topicref href="roman-roads_reference-glue.dita"/>
14      <topicref href="roman-roads_reference-glue.dita"/>
15      <topicref href="roman-roads_reference-glue.dita"/>
16    </topicref>
17    <topicref href="keymap_libraries.ditamap"
18      format="ditamap" processing-role="resource-only"/>
19 </map>
```

DITA Preview

Roman Roads

- [viae Romanae: Roman Roads](#)
 - [Why did Rome invest in roads?](#)
 - [Who built and maintained Roman roads?](#)
 - [How did the Romans build a road?](#)
 - [Reference](#)
 - [Construction materials](#)
 - [Existing Roman roads](#)
 - [Roman roadmaps](#)
 - [Further reading](#)



Heretto web CCMS

The screenshot shows the Heretto web CCMS interface. At the top left is the Heretto logo. A search bar is located at the top right. Below the logo is a 'Create New' button and a breadcrumb path: master > content > ACM-workshop-2023. The main area is divided into two columns: 'Browse' and 'Search'. The 'Browse' column shows a tree view with folders like 'Inbox', 'Starred', 'Recent', 'Content', 'acm-sample-1 (5)', 'acm-test-1 (3)', 'ACM-workshop-2023 (5)', and 'images (1)'. The 'Search' column shows a list of files with checkboxes, star icons, and file names: 'images 1 documents, 0 folders', 'What is generative AI? bard_concept_generative-ai.dita', 'Introduction to Google Bard bard_map.ditamap', 'Sample questions for Google Bard bard_reference.dita', and 'Introduction to Google Bard bard_topic_intro.dita'.

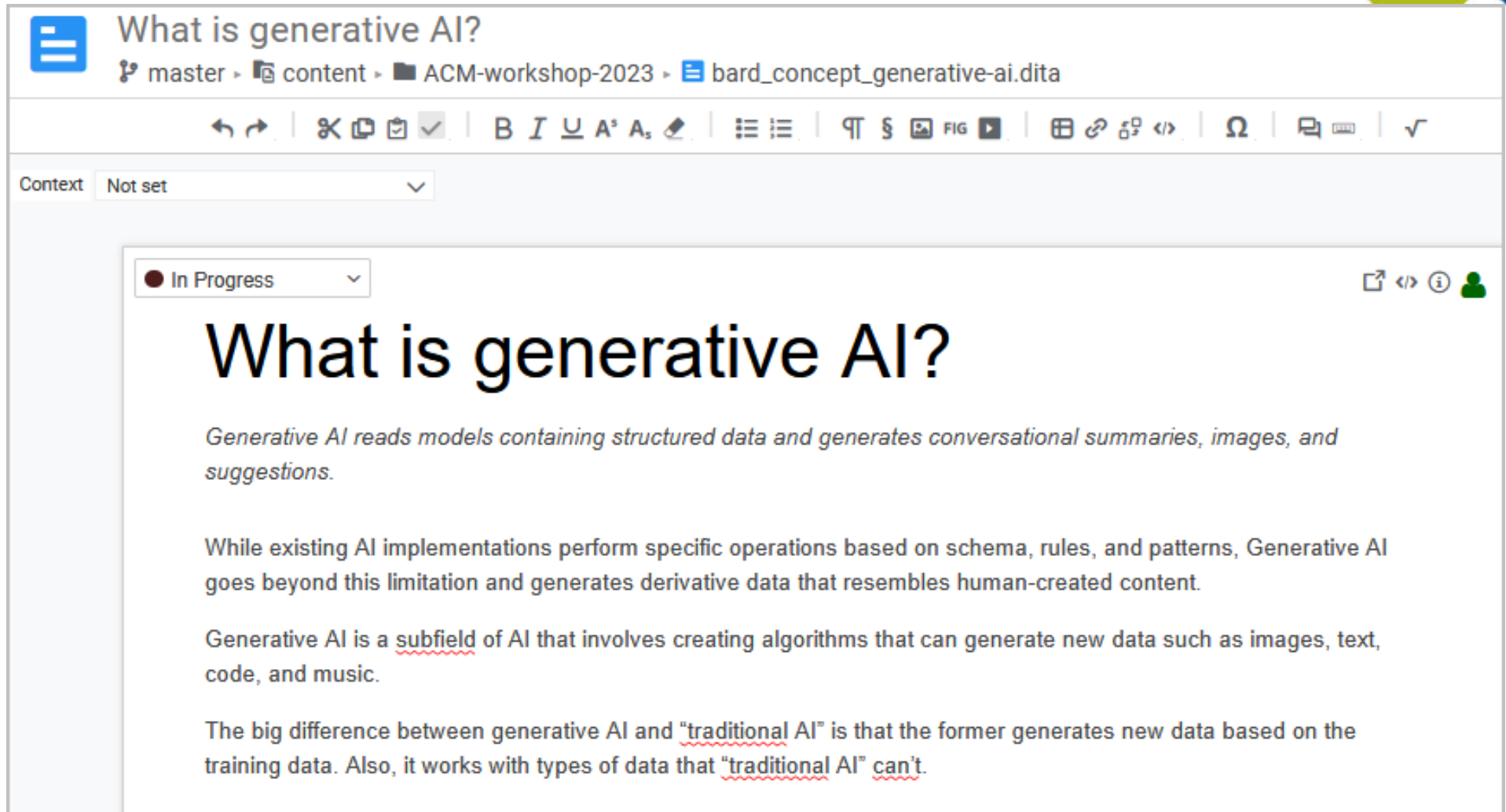
The screenshot shows the Heretto web CCMS navigation menu. At the top left is the Heretto logo. The menu items are: Content Library, Activity, Administration, Branching, Reports, Metadata, Collections, Taxonomy, Users, Groups, PDF Generator, DITA-OT Cloud, Publishing Configuration, and My Account.



Association for
Computing Machinery

Advancing Computing as a Science & Profession

Heretto web CCMS



The screenshot shows a web-based content management system (CCMS) editor. At the top, there is a navigation breadcrumb: master > content > ACM-workshop-2023 > bard_concept_generative-ai.dita. Below this is a rich text editor toolbar with various icons for undo, redo, bold, italic, underline, text color, background color, bulleted list, numbered list, link, unlink, code, and other functions. A 'Context' dropdown menu is set to 'Not set'. The main editing area has a status bar at the top left that says 'In Progress' and a top right corner with icons for share, code, info, and user. The document content is as follows:

What is generative AI?

Generative AI reads models containing structured data and generates conversational summaries, images, and suggestions.

While existing AI implementations perform specific operations based on schema, rules, and patterns, Generative AI goes beyond this limitation and generates derivative data that resembles human-created content.

Generative AI is a subfield of AI that involves creating algorithms that can generate new data such as images, text, code, and music.

The big difference between generative AI and "traditional AI" is that the former generates new data based on the training data. Also, it works with types of data that "traditional AI" can't.



Heretto web CCMS

What is generative AI?

master > content > ACM-workshop-2023 > bard_conce

Context Not set

In Progress

What is generative

Generative AI reads models containing structured data and ge suggestions.

While existing AI implementations perform specific operations goes beyond this limitation and generates derivative data that

Generative AI is a subfield of AI that involves creating algorithm code, and music.

The big difference between generative AI and "traditional AI" is training data. Also, it works with types of data that "traditional

What is generative AI?

con... > ... > bard_concept_generative-ai...

Details Activity

OVERVIEW

OUTPUT

ASSIGNMENTS

BRANCHES

RELEASES

LOCALES

LINKS

HISTORY

What is generative AI?

Generative AI reads models containing structured data and generates conversational summaries, images, and suggestions.

While existing AI implementations perform specific operations based on schemas, rules, and patterns, Generative AI goes beyond this limitation and generates derivative data that enables human-created content.

Generative AI is a subfield of AI that involves creating algorithms that can generate new data such as images, text, code, and music.

The big difference between generative AI and "traditional AI" is that the former generates new data based on the training data. Also, it works with types of data that "traditional AI" can't.

Type	Concept
Location	ACM-workshop-2023
Used by	1 object(s)
Uses	0 object(s)
Dependencies	0 object(s)
Modified	Sun, Oct 22, '23
Created	Sun, Oct 22, '23
Word count	100
Char count	789
Locked By	Collaboratively Locked
Status	In Progress
Owner	sjdoherty.acm@gmail.com



Oxygen Editor

The screenshot displays the Oxygen Editor interface for editing a DITA map. On the left, the **DITA Maps Manager** shows a tree view of the map structure, with the current map selected. The main editor window shows the content of the selected map, which is a task map titled "How did the Romans build a road?". The task map structure is visible, including a title, a short description, a prerequisite, and a list of pre-requisites. The **Attributes** panel on the right shows the current context and a list of attributes. The **Elements** panel on the right shows a list of elements available for use in the map.

DITA Maps Manager

Context: <Current map>

• rootmap_roman-roads.ditamap

Type filter text

- Roman Roads
 - viae Romanae: Roman Roads
 - Why did Rome invest in roads?
 - Who built and maintained Roman roads?
 - How did the Romans build a road?
 - Reference
 - Key definition map: Roman roads
 - Library: Roman roads {k_p_roads-at-scale}
 - k_pub-title = viae Romanae: Roman Roads

roman-roads-task.dita

task taskbody context

task title How did the Romans build a road? **title**

shortdesc **Short Description:** The process of planning and constructing a Roman road was consistent and repeatable across centuries, legions, and provinces. **shortdesc**

taskbody prereq **Before you begin:**

p **Pre-requisites**: Before a legion began its survey and excavation work, it required the following. **p**

ul

- **li** **p** Funding from the Roman central government. **p** **li**
- **li** **p** Approval from Roman military consuls and command. Typically one Roman legion was tasked with the construction of one road at a time. **p** **li**
- **li** **p** Available local knowledge. Roman engineers depended on the knowledge of locals in determining the optimal path of a road through local terrain. **p** **li**

Attributes

Attribute	Value
context	
audience	
base	
class	- topic/section ta...
conaction	
conkeyref	
conref	
conrefend	
deliveryTarget	
dir	
id	

Elements

- context
- abbreviated-form
- apiname
- audio
- b
- cite
- cmdname
- codeblock
- codeph
- data
- div
- dl
- draft-comment
- em

Text Grid Author

Oxygen Editor

The screenshot displays the Oxygen Editor interface for editing a DITA map. On the left, the **DITA Maps Manager** shows a tree view of the map structure, with the current map selected. The central editor window displays the content of the selected map, including a title, a short description, and a list of items. A large blue **LIVE DEMO** button is overlaid on the content. On the right, the **Attributes** and **Elements** panels are visible, showing the current context and the available attributes and elements for that context.

DITA Maps Manager

Context: <Current map>

- rootmap_roman-roads.ditamap
 - Roman Roads
 - viae Romanae: Roman Roads
 - Why did Rome invest in roads?
 - Who built and maintained Roman roads?
 - How did the Romans build a road?
 - Reference
 - Key definition map: Roman roads
 - Library: Roman roads {k_p_roads-at-scale}
 - k_pub-title = viae Romanae: Roman Roads

roman-roads-task.dita

task taskbody context

task title How did the Romans build a road? **title**

shortdesc **Short Description:** The process of planning and constructing a Roman road was consistent and repeatable across centuries, legions, and provinces.

LIVE DEMO

excavation work, it

ul

- li** **p** Funding from the Roman central government. **p** **li**
- li** **p** Approval from Roman military consuls and command. Typically one Roman legion was tasked with the construction of one road at a time. **p** **li**
- li** **p** Available local knowledge. Roman engineers depended on the knowledge of locals in determining the optimal path of a road through local terrain. **p** **li**

Attributes

Attribute	Value
context	
audience	
base	
class	- topic/section ta...
conaction	
conkeyref	
conref	
conrefend	
deliveryTarget	
dir	
id	

Elements

- context
- abbreviated-form
- apiname
- audio
- b
- cite
- cmdname
- codeblock
- codeph
- data
- div
- dl
- draft-comment
- em

Text **Grid** **Author**

Elements **Entities**

Upcoming ACM SIGDOC workshops



- 06-12-2026 **Introduction to Component Content Management**
Keith Schengili-Roberts
This hour-long workshop aims to provide students with an understanding of how a modern Component Content Management System (CCMS) is used. Keith will not only show the basics of how a technical writer would use the system by working and publishing DITA XML-based content, but also look at some of the administrative functions of the system.
- What is a Component Content Management System (CCMS)?
 - What is the typical technical infrastructure for a cloud-based CCMS?
 - How do technical writers work with and author content within a CCMS?
 - What are roles and groups within a CCMS?
 - How does a manager come up with production metrics within a CCMS?
- TBD-2026 **Hands-on workshop: Authoring in DITA**
Stan Doherty
- TBD-2026 **Using Committee resources to develop teaching plans for DITA**
TBD



Follow-up for ACM SIGDOC



1. Please review the resources that the Structured Authoring committee has developed thus far.

<https://acm-sigdoc-structured.org/1-curriculum-resources.html>

General concepts

The following resources provide introductions to many of the concepts underlying structured authoring in XML DITA and docs-as-code.

- **ACM-001: What is structured content?**

[gDoc](#) [DOCX](#) [PDF](#) [Feedback](#)

- **ACM-002: What is semantic markup?**

[gDoc](#) [DOCX](#) [PDF](#) [Feedback](#)

2. Email me to get a copy of these slides.

SJDoherty.ACM@gmail.com

3. Contact our Steering Committee to learn more about our work or to organize a Fall/2026 semester pilot.

Steering-Committee@acm-sigdoc-structured.org

4. Be well.



Association for
Computing Machinery

Advancing Computing as a Science & Profession



Questions

Answers

Stanley Doherty, Ph.D.
SJDoherty.ACM@gmail.com

<https://acm-sigdoc-structured.org/1-curriculum-resources.html>



Association for
Computing Machinery

Advancing Computing as a Science & Profession