Stage two: #647 Replace classification domain with new attribute

Replace the classification domain with a new attribute: `@subjectrefs`

Champion

Kristen James Eberlein, Eberlein Consulting LLC

Tracking information

Event	Date	Links
Initial suggestion	E-mail, 09 December 2021	Minutes, 04 January 2022
Stage 1 proposal accepted	11 January 2022	Minutes, 11 January 2022 GitHub issue
Stage 2 proposal submitted to TC for early feedback (not applicable to all proposals)	Not applicable	
Stage 2 proposal submitted to reviewers	E-mail, 07 March 2022	Carsten Brennecke, SAP Gershon Joseph, Precision Content I've also requested feedback from the following members of the larger DITA community: • Peyton Bentley, Kaplan • Joe Gelb, Zoomin Software • Pam Noreault, Ellucian • Shane Taylor, Cengage Group
Stage 2 proposal submitted to TC	E-mail, 17 March 2022 Revised: E-mail, 29 March 2022	
Stage 2 proposal discussed by TC	22 March 2022	
Stage 2 proposal approved by TC		

Original requirement or use case

The classification domain, introduced in DITA 1.2, is poorly designed, difficult to use, and was never widely implemented. However, it does provide the only mechanism for associating topic references with subjects that are defined in a subject scheme map. This proposal suggests removing the classification domain and replacing it with a new attribute: `@subjectrefs`.

Use cases

The new attribute will enable authors to easily associate a topic reference with a subject that is defined in a subject scheme map.

New terminology

Not applicable

Proposed solution

Remove the classification domain and the classification map. Add a new attribute, `@subjectrefs`, to the `<topicref>` element and specializations of `<topicref>`. This includes both elements in the base and the technical content edition.

> **Note** The `@subjectrefs` attribute has no meaning if it is specified on a key definition that does not reference a resource.

The `@subjectrefs` attribute will have the following characteristics:

- It will cascade.
- It will take multiple values, separated by white space. The values are one or more keys, defined in a subject scheme map.

While the DITA TC does not want to introduce any processing expectations for `@subjectrefs` in the DITA 2.0 time frame, it is possible that we will introduce them in the future.

Benefits

This proposal addresses the following questions:

Who will benefit from this feature?

- Any implementation that wants to classify content using subjects that are defined using a subject scheme map.
- The DITA TC and DITA specification editors, who will not have to maintain and support the classification domain.

What is the expected benefit?

This provides a significantly easier way to classify content than using the markup that is provided with the classification domain.

How many people probably will make use of this feature?

This is difficult to assess. It probably depends on whether current processors implement some OOB processing or whether early adopters contribute processing extensions to the DITA users community.

How much of a positive impact is expected for the users who will make use of the feature?

This is difficult to assess. Again, it depends on processor support. However, it might open the way for small implementations to develop simple approaches to faceted browsing or metadata-driven user experiences.

Technical requirements

See Proposed solution (2) above. The attribute will not contain translatable text, nor will it require any particular set up for current users of DITA.

Backwards compatibility

This proposal removes the classification domain and the classification map that was shipped with DITA 1.3. Implementing this proposal will require the following changes to DITA 2.0:

Removing the classification domain

- `doctypes/dtd/subjectScheme/classifyDomain.ent`
- `doctypes/dtd/subjectScheme/classifyDomain.mod`
- `doctypes/rng/subjectScheme/classifyDomain.rng`

In addition, the catalog files that currently reference the above files will need to be modified.

Removing the document-type shell for the classification map

(From the technical content package)

- doctypes/dtd/classificationMap directory
- doctypes/rng/classificationMap directory

In addition, the catalog files that currently reference the above files will need to be modified.

The classification domain and the document-type shell for the classification map will be loaded into the GitHub repository for specializations that are removed from the DITA standard.

Migration plan

Implementations that use the classification domain and classification map will need to do one of the following, if they want to move to DITA 2.0:

- Download the classification domain from the GitHub repository and integrate it into the relevant document-type shells. If the implementation currently uses the OOB OASIS-provided document-type shell for the classification map, they will need to move to using a custom document-type shell.
- Replace map markup that uses the classification domain with the new `@subjectrefs` attribute, as well as modifying any processing that is based on the classification domain.

Costs

This proposal will have an impact on the following groups:

Maintainers of the grammar files

Small effort to move files from the current GitHub repositories for the specification to the GitHub repository for specializations removed from the DITA standard.

Editors of the DITA specification

- Remove the classification elements topics
- Remove the "Classification maps" topic from the "DITA maps processing" chapter

Vendors of tools

Zoomin Software is the only company that uses the classification domain, as far as the DITA TC is aware. They will be affected in the same way as implementations that use the classification domain.

DITA community-at-large

By removing the classification domain and replacing it with a simple attribute, this change to the DITA specification should reduce the complexity of the DITA standard. Given how few companies have implemented the classification domain, the cost of migration will be limited to a handful of companies. The companies that currently use the classification domain, of which the TC are aware, are Kaplan and Elucian. Cengage also has expressed interest in deploying the classification domain, although they have indicated that they thought the `@subjectrefs` attribute would be an acceptable alternative.

Producing migration instructions or tools

These changes can be covered with minimal effort in the planned committee note about migrating from DITA 1.x to DITA 2.0.

Examples

This section contains examples of how the `@subjectrefs` attribute can be used. Each of the following examples uses the following subject scheme map:

```
<subjectScheme>
  <subjectdef keys="content-types">
    <subjectdef keys="conceptual-material"/>
    <subjectdef keys="reference"/>
    <subjectdef keys="tutorial"/>
  </subjectdef>
  <subjectdef keys="operating-systems">
    <subjectdef keys="linux"/>
    <subjectdef keys="macosx"/>
    <subjectdef keys="windows"/>
  </subjectdef>
  <subjectdef keys="user-tasks">
    <subjectdef keys="administering"/>
    <subjectdef keys="developing"/>
    <subjectdef keys="installing"/>
    <subjectdef keys="troubleshooting"/>
  </subjectdef>
</subjectScheme>
```

Figure 1: Example of the @subjectrefs attribute cascading

The following code sample shows how the `@subjectrefs` attribute cascades through a map branch. It indicates that all the topic references in the branch are associated with the subject of "installing".

```
<map>
<title>User assistance for the Acme Widget</title>
<!-- ... -->
<topicref keyref="install-overview" subjectrefs="installing">
  <topicref keyref="install-linux"/>
  <topicref keyref="install-macosx"/>
  <topicref keyref="install-windows"/>
</topicref>
<!-- ... -->
</map>
```

Note that the key definitions for the topics in the above DITA map are defined elsewhere.

Figure 2: Example of the @subjectrefs cascading and adding a value

The following code sample shows how the `@subjectrefs` attribute cascades through a map branch and adds its value to an existing instance of the `@subjectrefs` attribute:

```
<map>
<title>User assistance for the Acme Widget</title>
<!-- ... -->
<topicref keyref="install-overview" subjectrefs="installing">
  <topicref keyref="install-linux"/>
  <topicref keyref="install-macosx"/>
  <topicref keyref="install-windows"/>
  <topicref keyref="install-troubleshooting" subjectrefs="troubleshooting"/>
</topicref>
<!-- ... -->
</map>
```

The result is equivalent to the following markup:

```
<topicref keyref="install-troubleshooting" subjectrefs="installing troubleshooting"/>
```

Figure 3: Example of @subjectrefs and @cascade attribute interacting

In the following code sample, the value for the `@subjectrefs` attribute cascades through a map branch. However, the value of "installing" for the `@subjectrefs` attribute is not added to the topic reference for the `troubleshooting.dita` topic, since the topic reference specifies `cascade="nomerge"`.

```
<map>
<title>User assistance for the Acme Widget</title>
<!-- ... -->
<topicref keyref="install-overview" subjectrefs="installing">
  <topicref keyref="install-linux" subjectrefs="linux"/>
  <topicref keyref="install-macosx" subjectrefs="macosx"/>
  <topicref keyref="install-windows" subjectrefs="windows"/>
  <topicref href="troubleshooting.dita" cascade="nomerge" subjectrefs="troubleshooting"/>
</topicref>
<!-- ... -->
</map>
```

Note that if the topic reference to `troubleshooting.dita` did not specify a value for the `@subjectrefs` attribute, then the value of "installing" would cascade, since `cascade="nomerge"` would have no meaning.

Figure 4: Example of @subjectrefs specified on a key definition

The following code sample shows a `@subjectrefs` attribute specified on a key definition:

```
<keydef href="../developing-code-samples.dita" keys="dev-samples" subjectrefs="developing"/>
```

Hopefully users will use such markup carefully, as it clearly could introduce unexpected consequences if authors are unaware of `@subjectrefs` attributes that are specified on the key definition.