Stage 3: #670 Add impose-role attribute to topic references

Define a new attribute `@impose-role` that can be used on topicref elements. When specified on a topicref element with a value of "impose", and the topicref element is a reference to a map, this indicates that the role of the topicref overrides the role of the referenced map or topicref.

Champion

Provide information about the champion. If the proposal is submitted by a subcommittee, include the name of the point person. He or she should have prepared this proposal and thoroughly understand all of the content. The point person must be present at the TC calls when this proposal is discussed.

Tracking information: Stage two

Event	Date	Links
Initial suggestion	01 March 2022	Minutes, 01 March 2022
Stage 1 proposal accepted	15 March 2022	Minutes, 15 March 2022 https://github.com/oasis-tcs/dita/issues/670
Stage 2 proposal submitted to TC for early feedback (not applicable to all proposals)	24 June 2022	https://lists.oasis-open.org/archives/dita/202206/msg00057.html
Stage 2 proposal submitted to reviewers	28 June 2022	Carsten Brennecke, SAP Gershon Joseph, Precision Content
Stage 2 proposal submitted to TC	15 August 2022	
Stage 2 proposal discussed by TC	16 August 2022	https://lists.oasis-open.org/archives/dita/202208/msg00033.html
Stage 2 proposal approved by TC	23 August 2022	https://lists.oasis-open.org/archives/dita/202208/msg00050.html

Tracking information: Stage three

In the following table, insert cross references to e-mails to the TC list or meeting minutes:

- Be sure to use the public URL as the value of the `@href` attribute
- For link text for the cross references, use "E-mail, date month year", "GitHub issue", or "Minutes, date month year"

Event	Date	Links
Stage 3 proposal submitted to reviewers	<xref to e-mail sent to reviewers and cc'd to list>	<xref to e-mail sent to reviewers and cc'd to list>
Stage 3 proposal submitted to TC	<xref to e-mail sent to list>	
Stage 3 proposal discussed	<Date>	<xref to meeting minutes where discussed>

Event	Date	Links
Stage 3 proposal approved	<Date>	<xref to meeting minutes where discussed>

Approved technical requirements

Synopsis of the technical requirements as presented in the stage 2 proposal:

- Define a new `@impose-role` attribute that is available on `<topicref>` and its specializations
- Processing expectations that are associated with the new attribute:
  - When set to "impose", it declares the processing that is already described in 2.2.4.6.3 Cascading of roles from map to map
  - When set to "keeptarget", processors do not change the role of the referenced map or `<topicref>`.
- The attribute will typically have a default value set by the specialization

Dependencies or interrelated proposals

N/A

Modified grammar files

This proposal will add the `@impose-role` attribute to `<topicref>` and every specialization of `@topicref`.

Changes in base DTDs: https://github.com/oasis-tcs/dita/commit/cca6897ae802fe6fd4b2709f4781b21c1dc1027f

- Adds attribute to `map` module as an entity for easy reuse, with default set to "keeptarget"
- Fix the attribute as "keeptarget" for elements where "impose" makes no sense: `<ditavalref>`, `<topicsubject>`, `<topicapply>`, `<subjectref>`, `<keydef>`, `<mapref>`

Changes in base RNGs: https://github.com/oasis-tcs/dita/commit/b03661f13d55d153407ba0937b49b71e8573d4b3

- As above, adds attribute to `map` module as a definition for easy reuse, with default set to "keeptarget"
- Fix the attribute as "keeptarget" for elements where "impose" makes no sense: `<ditavalref>`, `<topicsubject>`, `<topicapply>`, `<subjectref>`, `<keydef>`, `<mapref>`

Modified terminology

n/a

Modified specification documentation

- The "attribute list" topic in the language reference will have a new entry for the `@impose-role` attribute. It will be grouped with "other" attributes (it is not a "common topicref attribute" as those generally inherit

and are also used on containers like `<map>` and `<reltable>`. See commit, definition list entry copied here:

@impose-role
> Specifies whether this element will impose its role on elements in a referenced map. The attribute is ignored if the target of the reference is not a map or branch of a map. The following values are valid:
>
> **keeptarget**
>> The role of the current reference is not imposed on the target of the reference. This is the default for the unspecialized `<topicref>` element and for many convenience elements such as `<keydef>`.
>
> **impose**
>> The rolel of the current reference is imposed on the target of the reference. For example, if a specialized topic reference `<chapter>` uses this value and references a map, a topic reference that resolves in place of the `<chapter>` will be treated as if it were a chapter.

- Add the attribute into the attribute section for topicref and each specialization; for most of these, add to the "attribute exception" section to list the fixed value of "keeptarget": https://github.com/oasis-tcs/dita/commit/6330f5c26913e933e1971ab24cb3b16adf640bd5

Migration plans for backwards incompatibilities

n/a

New architectural topic (not specifically for a new element)

Creates the following mostly-new topic, and revises the "example" topic that shows several examples of how chapter references would resolve. This is moving from the "cascading metadata" section in to the "Dita maps and their usage" section. Reflected in the following commit: https://github.com/oasis-tcs/dita/commit/bababb4c1647457a555f7116b37db7769095c2fb

Ideally the `<example>` sections in the main architectural topic here would be moved to example topics; however that's a little difficult to integrate before we rework the "dita maps and their usage" section so they are currently part of the overview topic.

Imposing roles when referencing a map

When specialized `<topicref>` elements reference a map, they might imply a semantic role for the referenced content. The `@impose-role` attribute provides a mechanism to declare that such references impose their original role on referenced content.

In many cases the `<topicref>` element is specialized in order to create a specific role for the reference. For example, the `<keydef>` element creates a new role for the reference, but does not create a role for the target of the reference. In other cases, the element is specialized to create a role for the target of the reference. For example, in the Bookmap specialization from the DITA Technical Communication specializations, the `<chapter>` element creates a role for the target of the reference: it declares that the referenced content is a chapter in the context of this map.

The declaration of roles can be harder to follow when the target of a reference is a map or branch of a map. In such cases, a `<topicref>` element can reference a map, which in turn references content. When resolving those references, processors need to know which roles created by the `<topicref>` elements need to be preserved for the content.

For example, assume that a specialized `<setupProject>` element indicates that the referenced content plays the "setup a project" role in a publication. This might result in special formatting or generated headings when the

content is rendered. If that element refers to a map instead of a topic, that specialized role still needs to be passed on to topics in the referenced map - regardless of what `<topicref>` elements might be used in that referenced map.

The `@impose-role` attribute provides a way for specialized elements to declare whether processors should use this behavior. This attribute is only evaluated when a `<topicref>` element refers to a map or branch of a map. In that case, it indicates whether the element provides a role for content that should be passed on to content in the referenced map.

The role created by a `<topicref>` is reflected by the `@class` hierarchy of the element. Processors that need to do something with the role do it based on that `@class` attribute. In the `<setupProject>` example above, that might be a `@class` attribute like `"- map/topicref taskmap/setupProject "`. Processors working with the reference know to render the referenced content based on that value. When `<setupProject>` instead pulls in content from another map, processors need to preserve that intent. Effectively, they need to preserve awareness of that `@class` attribute value for topics that are indirectly referenced through the other map.

Specialized topic references achieve this behavior by setting up a default value for the `@impose-role` attribute on the new element: `impose-role="impose"`.

When a role is imposed in this manner, it does not apply to all content referenced by the element. If a `<topicref>` refers to a branch of a map, the role is imposed only on the root element of that branch. If a `<topicref>` refers to an entire map, the role is imposed only on the highest-level topic references within that map. The role does not cascade to other nested referencs within the map.

For elements that do not create a role for the referenced content, the `@impose-role` attribute is defined with a default value indicating that the target of the reference keeps its original role: `impose-role="keeptarget"`. For example, the `<mapref>` element is a convenience element used to simplify references to other maps. It does not force the content in other maps to be treated as `<mapref>` - no special role is created for the referenced content. For this reason, it is defined in the grammar file with a fixed value of "keeptarget".

In some cases, preserving the role of a referencing element might result in out-of-context content. For example, a `<chapter>` element in one bookmap could pull in a `<part>` element from another bookmap, where that referenced `<part>` also contains nested `<chapter>` elements. Treating the `<part>` element as a `<chapter>` will result in a chapter that nests other chapters, which is not valid in bookmap and might not be understandable by processors. The result is implementation specific. Processors *MAY* choose to treat this as an error, issue a warning, or simply assign new roles to the problematic elements.

Defining a fixed role for a specialized element

In the Bookmap specialization from the OASIS DITA Technical Communications specializations, the `<chapter>` element creates a role for the referenced topic. In many contexts (such as a PDF version of the map), this will result in special formatting that identifies the topic as the start of a chapter.

When a chapter element refers to another map, topic references in that other map need to be treated as chapters in order to retain the structure of the book. The `@impose-role` attribute is set to a fixed value of "impose", which lets processors know that the role needs to be preserved for content in the other map.

In an RNG grammar file, this default value can be set as follows:

```
<optional>
  <attribute name="impose-role" a:defaultValue="keeptarget">
    <value>keeptarget</value>
  </attribute>
</optional>
```

In a DTD grammar file, this default value can be set as follows:

```
impose-role
                            (impose)
                                    'impose'
```

With these fixed values, a `<chapter>` element that refers to a map will impose the role of "chapter" as expected.

Imposing a role on a branch of a map

In this scenario, a specialized `<chapter>` element refers to a branch of another map. The chapter element does not need to set the `@impose-role` attribute directly, because it is defined with a default value in the XML grammar files. The element itself refers to a specific branch of the map, setting the `@format` attribute to indicate this is a map reference:

```
<bookmap>
  <!-- ... title, front matter, and other chapters -->
  <chapter href="reusemap.ditamap#examplebranch" format="ditamap"/>
  <!-- additional content -->
</bookmap>
```

The referenced map contains that branch along with other content:

```
<map>
  <title>Reusable map branches</title>
  <topicref> <!-- ... --> </topicref>
  <topicref href="parent.dita" id="examplebranch">
    <topicref href="child1.dita"/>
    <topicref href="child2.dita">
      <!-- more children -->
    </topicref>
  </topicref>
  <!-- ... more reusable branches -->
</map>
```

Because the `<chapter>` element is defined with a fixed value of "impose" for the `@impose-role` attribute, processors will impose the "chapter" role on the reference to `parent.dita` at the root of the referenced branch. The "chapter" role is not imposed on the child topics in that branch. While processors do not need to literally resolve the content in a normal map, the effective result is similar to this merged map:

```
<bookmap>
  <!-- ... title, front matter, and other chapters -->
  <chapter href="parent.dita">
    <topicref href="child1.dita"/>
    <topicref href="child2.dita">
      <!-- more children -->
    </topicref>
  </chapter>
  <!-- additional content -->
</bookmap>
```

Imposing a role on a referenced map

In this scenario, a specialized `<chapter>` element refers an entires submap. The chapter element does not need to set the `@impose-role` attribute directly, because it is defined with a default value in the XML grammar files. The element itself sets the `@format` attribute to indicate this is a map reference:

```
<bookmap>
  <!-- ... title, front matter, and other chapters -->
  <chapter href="nestedmap.ditamap" format="ditamap"/>
  <!-- additional content -->
</bookmap>
```

The referenced map contains three branches as children of of the root `<map>` element:

```
<map>
  <title>Reusable map branches</title>
  <topicref href="branch1.dita"> <!-- ... --> </topicref>
  <topicref href="branch2.dita">
    <topicref href="child1.dita"/>
    <topicref href="child2.dita">
```

```
      <!-- more children -->
    </topicref>
  </topicref>
  <topicref href="branch3.dita"> <!-- ... --> </topicref>
</map>
```

Because the `<chapter>` element is defined with a fixed value of "impose" for the `@impose-role` attribute, processors will impose the "chapter" role on the highest-level references within the nested map. This means the processors imposes the role of "chapter" on all three branches in the nested map. As with the previous example, the "chapter" role is not imposed on the child topics in each branch. While processors do not need to literally resolve the content in a normal map, the effective result is similar to this merged map:

```
<bookmap>
  <!-- ... title, front matter, and other chapters -->
  <chapter href="branch1.dita"> <!-- ... --> </chapter>
  <chapter href="branch2.dita">
    <topicref href="child1.dita"/>
    <topicref href="child2.dita">
      <!-- more children -->
    </topicref>
  </chapter>
  <chapter href="branch3.dita"> <!-- ... --> </chapter>
  <!-- additional content -->
</bookmap>
```

Example: How <topicref> roles are imposed on referenced maps

In this scenario, a specialized `<topicref>` element references content in another map.

Consider the scenario of a `<chapter>` element from the Bookmap specialization that references a DITA map. This scenario could take several forms:

Referenced map contains a single top-level <topicref> element
> The entire branch functions as if it were included in the bookmap. The "chapter" role is imposed on the branch, with the result that the top-level `<topicref>` element is processed as if it were the `<chapter>` element.

Referenced map contains multiple top-level <topicref> elements
> The "chapter" role is imposed on each top-level element in the referenced map. Each top-level `<topicref>` element is processed as if it were a `<chapter>` element.

Referenced map contains a single <appendix> element
> The "chapter" role is imposed on the `<appendix>` element, which is processed as it were a `<chapter>` element.

Referenced map contains a single <part> element, with nested <chapter> elements
> The "chapter" role is imposed on the `<part>` element, which is processed as it were a `<chapter>` element. Nested `<chapter>` elements might not be understandable by processors, which can treat this as an error or recover as they are able.

<chapter> element references a single <topicref> element rather than a map
> The "chapter" role is imposed on the referenced `<topicref>` element, which is processed as if it were a `<chapter>` element.