

Eberlein Consulting

DITA training, information architecture, and
strategy

Subject scheme maps

Contents

Subject scheme maps and their usage.....	3
Subject scheme maps.....	3
Controlled values for attributes.....	3
Binding controlled values to an attribute.....	4
Scaling a list of controlled values to define a taxonomy.....	5

Subject scheme maps and their usage

Subject scheme maps can be used to define controlled values and taxonomies. The controlled values and taxonomic subjects can be bound to attributes and element and attribute pairs.

Subject scheme maps

A subject scheme map enables adopters to create custom controlled values and to manage metadata attribute values without having to write a DITA specialization. Subject scheme maps use key definition to define a collection of controlled values rather than a collection of topics.

A controlled value is a keyword that can be used as a value for an attribute. For example, the `@audience` attribute can take a value that identifies the users that are associated with a particular product. Typical values for a medical-equipment product line might include therapist, oncologist, physicist, and radiologist. In a subject scheme map, an information architect can define a list of these `@audience` values.

Authoring tools *SHOULD* use these lists of controlled values to provide lists from which authors can select values when they specify attribute values. If controlled values for a metadata attribute are defined using the subject scheme map, tools *MAY* give an organization a list of readable labels, a hierarchy of values to simplify selection, and a shared definition of the value.

Controlled values can be used to classify content for filtering and flagging at build time. They also can be used for retrieval and traversal of the content at run time in conjunction with information viewing application that provide such functionality.

Comment by jelovirta

What is the interaction between key scopes and `<subjectdef>`? Should subject scheme maps be processed with the same rules for key scopes as normal keys?

Controlled values for attributes

Subject scheme maps can define controlled values for DITA attributes.

Each controlled value is defined using a `<subjectdef>` element, which is a specialization of the `<topicref>` element. The `<subjectdef>` element is used to define both a category and a list of controlled values. The top-level `<subjectdef>` element defines the category, and the children `<subjectdef>` elements define the controlled values.

The subject definitions can include additional information within a `<topicmeta>` element to clarify the meaning of a value:

- The `<navtitle>` element can provide a more readable value name.
- The `<shortdesc>` can provide a definition.

In addition, the `<subjectdef>` element can use an `@href` attribute to refer to a more detailed definition of the subject.

An editor *MAY* support drilling down to the subject definition topic for a detailed explanation of the subject. Tools *MAY* produce a help file, PDF, or other readable catalog to help authors better understand the controlled values.

An enumeration of controlled values can be defined with hierarchical levels by nesting subject definitions. This has an impact on how processors perform filtering and flagging. Processors *SHOULD* be aware of the hierarchy defined by nesting and process accordingly.

Comment by Kristen Eberlein on 7 October 2014

Need more precise working for above normative statement.

Example: Controlled values that provide additional information about the subject

The following example illustrates how a subject definition can provide a richer level of information about a controlled value:

```
<subjectdef keys="terminology" href="terminology.dita">
  <subjectdef keys="rf2119" href="rfc-2119.dita">
    <topicmeta>
      <navtitle>RFC-2119 terminology</navtitle>
      <shortdesc>The normative terminology that the DITA TC uses for the DITA specification</shortdesc>
    </topicmeta>
  </subjectdef>
</subjectdef>
```

Example: A hierarchy of controlled values and conditional processing

The following example illustrates a set of controlled values that contains a hierarchy.

```
<subjectScheme>
  <subjectdef keys="users">
    <subjectdef keys="therapist">
      <subjectdef keys="novice"/>
      <subjectdef keys="expert"/>
    </subjectdef>
    <subjectdef keys="oncologist"/>
    <subjectdef keys="physicist"/>
    <subjectdef keys="radiologist"/>
  </subjectdef>
</subjectScheme>
```

Processors that are aware of the hierarchy that is defined in the subject scheme will handle filtering and flagging in the following ways:

- If "therapist" is excluded and neither "novice" and "expert" are explicitly included, processors automatically should exclude both "novice" and "expert."
- If "novice" is included and "therapist" is not explicitly excluded, processors automatically should include "therapist" since it applies to "novice".
- If "therapist" is flagged and "novice" is not explicitly flagged, processors automatically should flag "novice" since it is a type of therapist.

Binding controlled values to an attribute

The controlled values defined in a subject scheme map can be bound to an attribute or an element and attribute pair. This has an impact on the expected behavior for processors and authoring tools.

If a subject scheme is used and an enumeration is bound, tools *SHOULD* validate controlled values for attributes against the subject scheme map. For authoring tools, this prevents users from entering misspelled or undefined values. For processors, processors *SHOULD* check that all values listed for an attribute in a DITAVAL file are bound to the attribute by the scheme before filtering or flagging.

Comment by Kristen Eberlein on 7 October 2014

The above paragraph originally contained the injunction "As with all key definitions and references, the reference must appear in the highest map that makes use of the controlled values."

The `<enumerationdef>` element binds the set of controlled values to an attribute. An enumeration can specify an empty category without children. In this case, no value is valid for the attribute. An enumeration also can specify an optional default value by using the `<defaultSubject>`.

Comment by Kristen Eberlein on 7 October 2014

Is the above information covered in the *Language Reference*?

Example: Binding a list of controlled values to the @audience attribute

The following example illustrates the use of the `<subjectdef>` element to define controlled values for types of users. It also binds the controlled values to the @audience attribute:

```
<subjectScheme>
<!-- Define types of users -->
  <subjectdef keys="users">
    <subjectdef keys="therapist"/>
    <subjectdef keys="oncologist"/>
    <subjectdef keys="physicist"/>
    <subjectdef keys="radiologist"/>
  </subjectdef>

  <!-- Bind the "users" subject to the @audience attribute.
  This restricts the @audience attribute to the following
  values: therapist, oncologist, physicist, radiologist -->
  <enumerationdef>
    <attributedef name="audience"/>
    <subjectdef keyref="users"/>
  </enumerationdef>
</subjectScheme>
```

Note that "users" is not a valid value for the @audience attribute.

Comment by Kristen Eberlein on 7 October 2014

Add descriptive text about how subjectScheme-aware editors and processors *SHOULD* handle this.

Scaling a list of controlled values to define a taxonomy

Optional classification elements make it possible to create a taxonomy from a list of controlled values.

A taxonomy differs from a controlled values list primarily in the degree of precision with which the metadata values are defined. A set of controlled values lists sometimes is regarded as the simplest form of taxonomy. Regardless of whether the goal is a simple list of controlled values or a taxonomy:

- The same core elements are used: `<subjectScheme>`, `<subjectdef>`, and `<subjectref>`.
- A category and its subjects can have a binding that enumerates the values of a metadata attribute.

Beyond the core elements and the attribute binding elements, sophisticated taxonomies can take advantage of some optional elements. These optional elements make it possible to specify more precise relationships among subjects. The `<hasNarrower>`, `<hasPart>`, `<hasKind>`, `<hasInstance>`, and `<hasRelated>` elements specify the kind of relationship in a hierarchy between a container subject and its contained subjects.

While users who have access to sophisticated processing tools benefit from defining taxonomies with this level of precision, other users can safely ignore this advanced markup and define taxonomies with hierarchies of `subjectdef` elements that are not precise about the kind of relationship between the subjects.

Example: A taxonomy defined using subject scheme elements

The following example defines San Francisco as both an instance of a city and a geographic part of California.

```
<subjectScheme>
  <hasInstance>
    <subjectdef keys="city" navtitle="City">
      <subjectdef keys="la" navtitle="Los Angeles"/>
      <subjectdef keys="nyc" navtitle="New York City"/>
      <subjectdef keys="sf" navtitle="San Francisco"/>
    </subjectdef>
    <subjectdef keys="state" navtitle="State">
      <subjectdef keys="ca" navtitle="California"/>
      <subjectdef keys="ny" navtitle="New York"/>
    </subjectdef>
  </hasInstance>
  <hasPart>
    <subjectdef keys="place" navtitle="Place">
      <subjectdef keys="ca">
        <subjectdef keys="la"/>
        <subjectdef keys="sf"/>
      </subjectdef>
      <subjectdef keys="ny">
        <subjectdef keys="nyc"/>
      </subjectdef>
    </subjectdef>
  </hasPart>
</subjectScheme>
```

Sophisticated tools can use this scheme to associate content about San Francisco with related content about other California places or with related content about other cities (depending on the interests of the current user).

The scheme can also define relationships between subjects that are not hierarchical. For instance, cities sometimes have "sister city" relationships. The example scheme could add a `<subjectRelTable>` element to define these associative relationships, with a row for each sister-city pair and the two cities in different columns in the row.