DITA 2.0 proposed feature #292: Add @colspan, @rowspan, @headers, @scope, and <title> to <simpletable>

The `<simpletable>` element currently does not allow custom definition of the number of columns or rows that a cell should span. Likewise, the current `<simpletable>` content model does not include a caption or title for a table. Allowing column/row spanning and title will make the `<simpletable>` element more similar to the HTML table, which should make it more usable for authors working in web contexts and also more compatible with content structures in HDITA (the Lightweight DITA authoring format that is based on HTML5).

Date and version information

Include the following information:

Date that this feature proposal was completed

??

Champion of the proposal

Carlos Evia

Links to any previous versions of the proposal

N/A

Links to minutes where this proposal was discussed at stage 1 and moved to stage 2

https://lists.oasis-open.org/archives/dita/201908/msg00027.html

Reviewers for Stage 2 proposal

Robert Anderson and Eliot Kimber

Links to e-mail discussion that resulted in new versions of the proposal

https://lists.oasis-open.org/archives/dita/201907/msg00100.html

Link to the GitHub issue

https://github.com/oasis-tcs/dita/issues/292

Original requirement or use case

Presented July 30 2019:

We've had two table models in DITA for a long time - CALS and simple table. The CALS model is meant to handle even the most complex table markup, but can be cumbersome to use or specialize. Simple table was meant to handle tables that do not need the complexity, and is intentionally easy to specialize. Outside of DITA, HTML5 has one of the most commonly used table models. The HTML table maps very easily to DITA's simple table, except that the simple table is missing two very common features of the HTML5 table -- row/column spanning and table captions. This poses a problem for tool compatibility, especially for Lightweight DITA, where HTML is the common table model in both HDITA and MDITA.

Rather than add a third table model to support HTML table markup, we propose extending the markup for `<simpletable>` to allow it to support common HTML table features. This could be done in a backwards-compatible fashion - existing `<simpletable>` markup would remain valid, we would only be adding options to support those features. We would still have a complex (CALS) model and simple model, this just means that our simple table would map more easily to the widely supported HTML5 table.

The value of `<simpletable>` for specialization purposes would be preserved, and we would gain greater compatibility with the dominant table markup standard in today's web and CMS tooling.

New terminology

N/A

Proposed solution

Add the existing DITA `<title>` element as optional first child of `<simpletable>` to mimic the behavior of `<caption>` in the HTML5 `<table>` element. In the base topic, `<title>` is already contained by `<data>`, `<example>`, `<fig>`, `<figgroup>`, `<linklist>`, `<section>`, `<table>`, and `<topic>`. The element can remain unchanged when contained by `<simpletable>`.

Add the new `@colspan` and `@rowspan` optional attributes to `<stentry>`. `@colspan` indicates the number of columns that the cell is to span inside a `<simpletable>`, and `@rowspan` indicates the number of rows that the cell is to span inside a `<simpletable>`. Add the optional attributes `@headers` and `@scope` to `<stentry>` for full accessibility of simple tables.

Benefits

Who will benefit from this feature?

DITA and LwDITA authors

What is the expected benefit?

The `<simpletable>` element will be easier to map to `<table>` in HTML5, which makes it relevant for authors creating content for Web publication. The CALS `<table>` will still be available for more complex tabular environments.

How many people probably will make use of this feature?

Many, mainly those authoring content for Web publication.

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

Significant.

1. Adding `<title>` to `<simpletable>` will make it a more accessible element and a more complete alternative to `<table>`
2. Allowing `@rowspan` and `@colspan` in `<simpletable>` will make it more compatible with the authoring formats of LwDITA, which should attract new users.
3. Adding `@headers` and `@scope` to `<stentry>` will enhance accessibility of non-trivial simple tables.

Technical requirements

Provide a detailed description of how the solution will work. Be sure to include the following details:

Adding new elements or attributes

Adding an element

- Adds `<title>` as optional first child of `<simpletable>`
- The block element will contain translatable text and expected to behave like `<title>` does in `<data>`, `<example>`, `<fig>`, `<figgroup>`, `<linklist>`, `<section>`, `<table>`, and `<topic>`.

Adding an attribute

- Adds `@colspan`, `@rowspan`, `@headers`, and `@scope` as optional attributes to `<stentry>`
- These attributes will not contain translatable text
- The value for `@colspan` must be a valid non-negative integer greater than zero
- The value for `@rowspan` must be a valid non-negative integer. As its namesake attribute from the HTML5 spec indicates, a value of 0 (zero) in an instance of `@rowspan` means that the cell is to span all the remaining rows in the table environment.
- The attributes headers and scope should follow the structure that they already have under `<entry>` in CALS tables.

Processing impact

Minimal. Processors would just need to acknowledge `<title>`, which already exists in many DITA environments, in `<simpletable>`. The new attributes for spanning rows and columns will not require major processing changes for processors generating HTML tables.

Overall usability

The proposed changes will actually increase the usability of `<simpletable>`, as it will more closely resemble the widely used `<table>` element in HTML. It will also increase its accessibility and compatibility with LwDITA authoring formats.

Backwards compatibility

The proposed changes will not affect backwards compatibility because the `<title>` element and `@rowspan` and `@colspan` attributes will be optional in `<simpletable>`. Existing `<simpletable>` versions will not need the proposed changes.

Was this change previously announced in an earlier version of DITA?

No.

Removing a document type that was shipped in DITA 1.3?

No.

Removing a domain that was shipped in DITA 1.3?

No.

Removing a domain from a document type shell was shipped in DITA 1.3?

No.

Removing or renaming an element that was shipped in DITA 1.3?

No.

Removing or renaming an attribute that was shipped in DITA 1.3?

No.

Changing the meaning of an element or attribute in a way that would disallow existing usage?

No.

Changing a content model by removing something that was previously allowed, or by requiring something that was not?

No.

Changing specialization ancestry?

No.

Removing or replacing a processing feature that was defined in DITA 1.3?

No.

Are element or attribute groups being renamed or shuffled?

No

Costs

Outline the impact (time and effort) of the feature on the following groups.

Maintainers of the grammar files

Small. Update content models and attribute list declarations and any related tests.

Editors of the DITA specification

- The `<simpletable>` topic will need to be edited to include the optional caption/title and attributes for spanning rows and columns
- Topics for attributes will need to be edited to included the new `@rowspan` and `@colspan`

Vendors of tools

- Small to moderate. For processors that generate HTML tables the cost is small. For processors that generate other table formats there will be some additional work, probably to adapt existing CALS table handling for row and column spans to simple table processing. Editors that currently assume simple tables do not do spanning may need to be updated to allow authoring of spanned rows and columns.

DITA community-at-large

- Will this feature add to the perception that DITA is becoming too complex?
  - Probably the opposite. `<simpletable>` can become a more popular element with these additions.
- Will it be simple for end users to understand?
  - Yes, the syntax proposed is very similar to widely used HTML attributes.

Examples

Source:

```
<simpletable>
    <title>Food log for today</title>
<sthead>
      <stentry>Meal</stentry>
      <stentry>Food</stentry>
    </sthead>
    <strow>
      <stentry>Breakfast</stentry>
      <stentry>Eggs</stentry>
    </strow>
    <strow>
      <stentry>Lunch</stentry>
      <stentry>Salad</stentry>
    </strow>
    <strow>
      <stentry>Dinner</stentry>
      <stentry>Chicken</stentry>
    </strow>
    </simpletable>
```

Figure 1: Example of <title> element

Formatted output:

Table 1: Food log for today

Meal	Food
Breakfast	Eggs
Lunch	Salad
Dinner	Chicken

Source:

```
<simpletable>
    <title>Food log for today</title>
    <sthead>
      <stentry>Meal</stentry>
      <stentry>Food</stentry>
    </sthead>
    <strow>
      <stentry colspan="2">Fasting period</stentry>
    </strow>
    <strow>
      <stentry>Lunch</stentry>
      <stentry>Salad</stentry>
    </strow>
    <strow>
```

```
      <stentry>Dinner</stentry>
      <stentry>Chicken</stentry>
    </strow>
    </simpletable>
```

Figure 2: Example of @colspan attribute

Formatted output:

Table 2: Food log for today

Meal	Food
Fasting period	
Lunch	Salad
Dinner	Chicken

Source:

```
<simpletable>
<title>Food log for today</title>
<sthead>
  <stentry>Meal</stentry>
  <stentry>Food</stentry>
</sthead>
<strow>
  <stentry>Breakfast</stentry>
  <stentry>Eggs</stentry>
</strow>
<strow>
  <stentry>Lunch</stentry>
  <stentry rowspan="2">Chicken</stentry>
</strow>
<strow>
  <stentry>Dinner</stentry>
</strow>
</simpletable>
```

Figure 3: Example of @rowspan attribute

Formatted output:

Table 3: Food log for today

Meal	Food
Breakfast	Eggs
Lunch	Chicken
Dinner	