Allow example element in more places #297

The `<example>` element is not allowed in several elements, which means that content cannot be used properly (or tagged semantically) in certain contexts. Adding `<example>` to more contexts solves the issue.

Date and version information

Date that this feature proposal was completed

11 SEP 2019

Champion of the proposal

Scott Hudson

Links to any previous versions of the proposal

https://lists.oasis-open.org/archives/dita/201906/msg00026.html

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

https://lists.oasis-open.org/archives/dita/201907/msg00060.html

Links to email discussion that resulted in new versions of the proposal

https://lists.oasis-open.org/archives/dita/201907/msg00060.html

Link to the GitHub issue

https://github.com/oasis-tcs/dita/issues/297

Original requirement or use case

The content model of `<example>` is much too restrictive, limiting how and where content can be semantically marked as exemplary content. As a result, writers often hardcode paragraphs with "For example:" or other non-semantic constructions.

By allowing `<example>` in more places, content can be properly tagged with the `<example>` semantic.

Proposed to list on 10 JUN 2019. Discussed at 11JUN 2019 TC meeting.

Use cases

When implemented, authors can include example content as part of most block elements. For example:

```
<section id="section_EngineOutClimboutAnti-IceOnD11">
<title>Engine-out climbout, anti-ice on (D11)</title>
<sectiondiv product="JetPlan">
 <p><synph>D11=</synph></p>
 <example><codeph>D11=1LE</codeph></example>
 </sectiondiv>
<example><title>Scenario: Engine-out, anti-ice on</title>
<p>Enables you to set a divert climb airspeed for the climbout portion
 of the flight under an engine-out, anti-ice on, driftdown scenario.
The value is the name of the table data available for the aircraft.</p>
<codeblock>1LE</codeblock></example></section>
```

Proposed solution

Allow `<example>` everywhere that `<fig>` can go today, except do not allow `<example>` directly within `<example>`. Also allow `<example>` as a child of `<fig>`.

Benefits

Address the following questions:

Who benefits from this feature?

All writers and information consumers.

What is the expected benefit?

Allowing `<example>` in more contexts enables better semantic accuracy in labeling example content. This feature improves styling and search with better semantic accuracy. The feature discourages "tag abuse," which forces inconsistent styling of example content.

How many people probably can use this feature?

Everyone.

How much of a positive effect is expected for the users who can make use of the feature?

Significant.

1. This feature allows `<example>` to exist where it does today (1.3 use of the element is fully backward compatible).
2. This feature allows `<example>` in most block contexts.
3. This feature would not create a new context for titled areas, since the additional contexts already allow for titled elements. This suggests processors should be able to cope with the new context without much difficulty.
4. This feature avoids the "why would they do this" condition since it prevents nesting of `<example>` as a child of `<example>`.

Technical requirements

Renaming or refactoring elements and attributes

Refactoring of <example>

Add example to the following contexts:

- `<abstract>`
- `<body>`
- `<bodydiv>`
- `<conbody>`
- `<conbodydiv>`
- `<dd>`
- `<div>`
- `<draft-comment>`
- `<entry>`
- `<fn>`
- `<itemgroup>`
- `<li>`
- `<lq>`
- `<note>`
- `<p>`
- `<pd>`
- `<refbody>`
- `<refbodydiv>`
- `<section>`
- `<sectiondiv>`
- `<stentry>`
- `<taskbody>`

Processing impact

Minimal. All `@domains`- and `@class`-based processing of specialized elements and attributes will continue to function exactly as it has before. Processors may need to be adjusted to account for the additional contexts of `<example>`, but these contexts already allow for titled blocks.

Overall usability

Current DITA users, and current document type shells, will be unaffected. Usability of authoring with specializations is ultimately improved due to content models that more closely model business requirements.

Backwards compatibility

Backward compatibility is not affected, since the `<example>` element is an addition to existing content models.

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?

The solution allows the `<example>` element directly within the following elements: <abstract>, <body>, <bodydiv>, <conbody>, <conbodydiv>, <dd>, <div>, <draft-comment>, <entry>,<fn>, <itemgroup>, <li>, <lq>, <note>, <p>, <pd>, <refbody>, <refbodydiv>, <section>,<sectiondiv>, <stentry>, <taskbody>.

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?

This solution can be achieved by adding `<example>` to the following common element sets in `dita/doctypes/rng/base/commonElementsMod.rng` :

- basic.block
- basic.block.notbl
- basic.block.nonote
- basic.block.nopara
- basic.block.nolq

```
<define name="basic.block">
        <a:documentation>Elements common to most body-like contexts</a:documentation>
        <choice>
          <ref name="dl"/>
          <ref name="div" dita:since="1.3"/>
          <ref name="example"/>
          <ref name="fig"/>
          <ref name="image"/>
          <ref name="lines"/>
          <ref name="lq"/>
          <ref name="note"/>
          <ref name="object"/>
          <ref name="ol"/>
          <ref name="p"/>
          <ref name="pre"/>
          <ref name="simpletable"/>
          <ref name="sl"/>
          <ref name="table"/>
          <ref name="ul"/>
        </choice>
      </define>
      <define name="basic.block.notbl">
        <choice>
          <ref name="dl"/>
          <ref name="div" dita:since="1.3"/>
          <ref name="example"/>
          <ref name="fig"/>
          <ref name="image"/>
          <ref name="lines"/>
          <ref name="lq"/>
          <ref name="note"/>
          <ref name="object"/>
          <ref name="ol"/>
          <ref name="p"/>
          <ref name="pre"/>
          <ref name="sl"/>
          <ref name="ul"/>
        </choice>
      </define>
      <define name="basic.block.nonote">
        <choice>
          <ref name="dl"/>
          <ref name="div" dita:since="1.3"/>
          <ref name="example"/>
          <ref name="fig"/>
          <ref name="image"/>
          <ref name="lines"/>
          <ref name="lq"/>
          <ref name="object"/>
          <ref name="ol"/>
          <ref name="p"/>
          <ref name="pre"/>
          <ref name="simpletable"/>
          <ref name="sl"/>
          <ref name="table"/>
          <ref name="ul"/>
        </choice>
      </define>
      <define name="basic.block.nopara">
        <choice>
          <ref name="dl"/>
          <ref name="div" dita:since="1.3"/>
          <ref name="example"/>
          <ref name="fig"/>
          <ref name="image"/>
          <ref name="lines"/>
          <ref name="lq"/>
          <ref name="note"/>
          <ref name="object"/>
          <ref name="ol"/>
          <ref name="pre"/>
          <ref name="simpletable"/>
          <ref name="sl"/>
          <ref name="table"/>
          <ref name="ul"/>
        </choice>
      </define>
      <define name="basic.block.nolq">
        <choice>
          <ref name="dl"/>
```

```
          <ref name="div" dita:since="1.3"/>
          <ref name="example"/>
          <ref name="fig"/>
          <ref name="image"/>
          <ref name="lines"/>
          <ref name="note"/>
          <ref name="object"/>
          <ref name="ol"/>
          <ref name="p"/>
          <ref name="pre"/>
          <ref name="simpletable"/>
          <ref name="sl"/>
          <ref name="table"/>
          <ref name="ul"/>
        </choice>
      </define>
```

The solution requires the addition of a new common element set:

```
<define name="basic.block.noex">
        <a:documentation>Elements common to most body-like contexts</a:documentation>
        <choice>
          <ref name="dl"/>
          <ref name="div" dita:since="1.3"/>
          <ref name="fig"/>
          <ref name="image"/>
          <ref name="lines"/>
          <ref name="lq"/>
          <ref name="note"/>
          <ref name="object"/>
          <ref name="ol"/>
          <ref name="p"/>
          <ref name="pre"/>
          <ref name="simpletable"/>
          <ref name="sl"/>
          <ref name="table"/>
          <ref name="ul"/>
        </choice>
      </define>
```

The solution requires the modification of the example.cnt model in `dita/doctypes/rng/base/topicMod.rng` to use the new common element set (to prevent nesting of example):

```
<define name="example.cnt">
      <choice>
        <text/>
        <ref name="basic.block.noex"/>
        <ref name="basic.ph"/>
        <ref name="data.elements.incl"/>
        <ref name="foreign.unknown.incl"/>
        <ref name="title"/>
        <ref name="txt.incl"/>
      </choice>
    </define>
```

Migration plan

No existing content requires migration, since the `<example>` element is an addition to existing content models.

Costs

Maintainers of the grammar files

Grammar files need to be updated as outlined in this proposal. Downstream schema generation should be unaffected.

Editors of the DITA specification

Since the content model appendix is auto-generated, no additional changes to the DITA specification is needed.

Vendors of tools

Tool vendors may need to update functionality if they have unique support for display or processing of the `<example>` elements. Otherwise, no changes should be necessary.

DITA community-at-large

The solution should improve authoring and user experience with the addition of more places that `<example>` is allowed.

Examples

See Use cases on page 1, above.