DITA 1.3 proposed feature #14000

Contents

DITA 1.3 proposed feature #14000

Provide a domain that provides the ruby markup from HTML5 as used by Japanese and other ideographic languages.

Date and version information

- Proposal Completed: 1 Oct 2012
- Champion: Eliot Kimber,
- Email discussion: *https://lists.oasis-open.org/archives/dita/201201/msg00061.html*.

Original requirement

In Japanese, the pronunciation of ideographic characters cannot always (or often) be known from context. The ideographic characters are annotated with their phonetic transliteration, a "ruby", which is rendered above or beside or following the ideographs. This is standard Japanese typography.

Use cases

Japanese and other ideographic or similar languages where correct pronunciation may require phonetic annotation.

Benefits

- Who benefits: DITA users who create Japanese- and similar ideographic language content.
- Expected benefit: Enables correct typographic rendering of ruby annotations.
- Potential users: At a minimum, all or most companies that write in or localize to Japanese.
- Degree of positive impact: Significant. Makes producing Japanese documents with proper ruby annotation possible without the need to implement a custom vocabulary module.

Costs

Costs:

- Maintainers of the DTDs and XSDs: Adds a new vocabulary module, which must be integrated into the appropriate shell document types.
- Editors of the DITA specification:
  - How many new topics will be required? 1 new reference topic.
  - How many existing topics will need to be edited? TBD, but may be none.
  - Will the feature require substantial changes to the information architecture of the DITA specification? No architectural change.
- Vendors of tools: Processors that render DITA content visually should provide appropriate rendering of ruby content. For HTML-based outputs, ruby support is built in to most, if not all, browsers.
- DITA community-at-large. Will this feature add to the perception that DITA is becoming too complex? Will it be simple for end users to understand?

  This feature adds a new optional vocabulary module. Users who need it will appreciate having it readily available. Users who do not need it may safely ignore it. The general architecture and semantics of DITA are not affected by this proposal.

Technical requirements

Define a new vocabulary module, rubyDomain, that defines the <ruby> element type with the same content as defined in HTML5.

ruby	Contains the subelements <rb>, <rb>, and <rt>. See the HTML5 definition of the <ruby> element.

rubyDomain.ent:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- =============================================================
     DITA Ruby Domain

     Defines equivalent of HTML ruby elements for marking up
     Japanese language documents.

     Copyright (c) 2012 OASIS Open

     ============================================================= -->


<!-- ============================================================= -->
<!--                   Ruby DOMAIN ENTITIES                  -->
<!-- ============================================================= -->

<!ENTITY % ruby-d-ph
   " ruby
   "
>



<!ENTITY   ruby-d-att
  "(topic ruby-d)"
>


<!-- ================== End DITA For Publishers Ruby Domain Entities
 ============ -->
```

rubyDomain.mod:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- =============================================================
     DITA Ruby Domain

     Defines equivalent of HTML ruby elements for marking up
     Japanese language documents.

     Copyright (c) 2012 OASIS Open

     ============================================================= -->

 <!ENTITY % ruby           "ruby" >

 <!ENTITY % rb             "rb" >
 <!ENTITY % rp             "rp" >
 <!ENTITY % rt             "rt" >


<!-- ============================================================= -->
<!--                   ELEMENT NAME ENTITIES                       -->
<!-- ============================================================= -->


<!-- ============================================================= -->
```

```
<!--                    ELEMENT DECLARATIONS                       -->
<!-- ============================================================= -->


<!-- In order to support HTML5, which allows a mix of PCDATA, other phrase-
     level elements, and <rt> and <rp>, the content model must allow
     %ph;, which means that the DTD allows <ruby> within <ruby>. However,
     <ruby> should *not* be used within <ruby>, per the HTML
     constraints on <ruby>. Likewise, if <rp> is used, it should be
     used as <rp>(</rp><rt>...</rt><rp>)</rp> per the HTML5 spec.
  -->
<!ENTITY % ruby.content
"
  (%ph.cnt; |
   %rb; |
   %rp; |
   %rt;)*
  "
>
<!ENTITY % ruby.attributes
 "
   %id-atts;
  %localization-atts;
  base
    CDATA
    #IMPLIED
  %base-attribute-extensions;
  outputclass
    CDATA
    'ruby'
 "
>
<!ELEMENT ruby %ruby.content; >
<!ATTLIST ruby %ruby.attributes; >

<!ENTITY % rb.content
"
  (#PCDATA
  )*
  "
>
<!ENTITY % rb.attributes
 "
   %id-atts;
  %localization-atts;
  base
    CDATA
    #IMPLIED
  %base-attribute-extensions;
  outputclass
    CDATA
    'rb'
 "
>
<!ELEMENT rb %rb.content; >
<!ATTLIST rb %rb.attributes; >

<!ENTITY % rp.content
"
  (#PCDATA
  )*
  "
>
<!ENTITY % rp.attributes
```

```
 "
   %id-atts;
  %localization-atts;
  base
    CDATA
    #IMPLIED
  %base-attribute-extensions;
  outputclass
    CDATA
    'rp'
 "
>
<!ELEMENT rp %rp.content; >
<!ATTLIST rp %rp.attributes; >

<!ENTITY % rt.content
"
  (#PCDATA
  )*
  "
>
<!ENTITY % rt.attributes
 "
   %id-atts;
  %localization-atts;
  base
    CDATA
    #IMPLIED
  %base-attribute-extensions;
  outputclass
    CDATA
    'rt'
 "
>
<!ELEMENT rt %rt.content; >
<!ATTLIST rt %rt.attributes; >



<!-- ============================================================= -->
<!--                    SPECIALIZATION ATTRIBUTE DECLARATIONS      -->
<!-- ============================================================= -->

<!ATTLIST ruby              %global-atts;  class CDATA "+ topic/ph    ruby-
d/ruby ">
<!ATTLIST rb                %global-atts;  class CDATA "+ topic/ph    ruby-
d/rb ">
<!ATTLIST rp                %global-atts;  class CDATA "+ topic/ph    ruby-
d/rp ">
<!ATTLIST rt                %global-atts;  class CDATA "+ topic/ph    ruby-
d/rt ">


<!-- ================== End Ruby Domain ==================== -->
```

Figure 1: DTD Syntax domain module declarations

rubyDomainMod.xsd:

```
TBD
```

Figure 2: XSD domain module declarations

Note: These declarations are not complete. Waiting to work out details of how to declare equivalent of <xs:any> in RNC syntax.

rubyDomainMod.rnc:

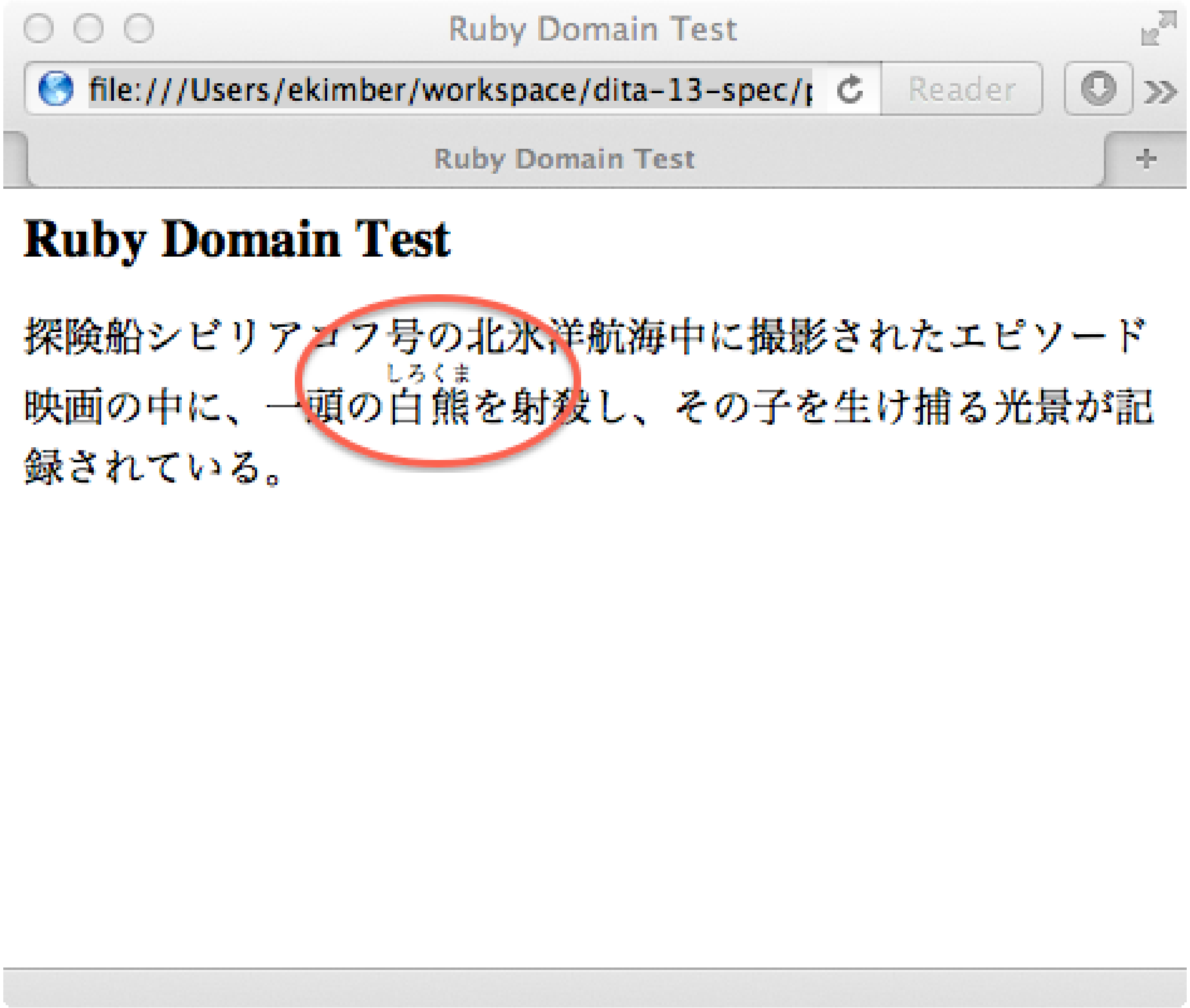
```
TBD
```

Figure 3: RelaxNG Compact domain module declarations

Examples

```
<topic id="ruby-test-topic-01">
  <title>Ruby Domain Test</title>
  <body>
    <p> #####################################<ruby>
      <rb>##</rb>
      <rp>#</rp>
      <rt>####</rt>
      <rp>#</rp>
    </ruby>########################</p>
  </body>
</topic>
```

Figure 4: Sample topic with ruby markup.



Ruby Domain Test

探険船シビリアコフ号の北氷洋航海中に撮影されたエピソード
しろくま
映画の中に、一頭の白熊を射殺し、その子を生け捕る光景が記
録されている。

Figure 5: Ruby as rendered in HTML by Safari browser