The Internationalization Tag Set

Richard Ishida
W3C Internationalization Activity Lead
Background

The basic idea
Ways to use ITS
Data categories
Summary
Objectives of ITS

- Ensure that XML tag sets support international use
- Ensure that XML tag sets support localization needs
- Guide designers of XML tag sets and content developers away from translatability problems.
- Ensure that localizers and format converters can recognize the meaning of tags in a tag set.
The Internationalization Tag Set

Background
Who needs to hear this?

Schema developers
- affects the way they create document formats

Content authors
- using new tags for content development

Process support
- applying general rules to content for localization

Localizers
- recognize tags for translation tool and process use
A schema (with a small 's') describes the structure of an XML document. Some formats in which people write schemas include DTDs (Document Type Definitions), the W3C's XML Schema (with a capital 'S'), and RelaxNG. The ITS Best Practises document will contain examples in each of these formats of how to implement the ITS tagset.
# Background

## History

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>late 1990s</td>
<td>Richard Ishida explores ideas about developing XML formats for multilingual use</td>
</tr>
<tr>
<td>June 2000</td>
<td>Richard presents at LISA Forum presentation, Tokyo, “Localization Considerations in DTD Design”</td>
</tr>
<tr>
<td>late 2000</td>
<td>Steven Forth gets people together, and introduces Yves Savourel to the mix</td>
</tr>
<tr>
<td>2001</td>
<td>Richard and Yves publish “Requirements for Localizable DTD Design.”</td>
</tr>
<tr>
<td>Jan 2005</td>
<td>ITS Working Group starts under W3C Internationalization Activity, Yves Savourel chairs – requirements developed, then Working Drafts for tag set and best practices</td>
</tr>
<tr>
<td>May 2006</td>
<td>Last call review for tag set document</td>
</tr>
<tr>
<td>Now</td>
<td>Last call completed, commenters satisfied – very soon hope to move tag set to Candidate Recommendation</td>
</tr>
<tr>
<td>Next</td>
<td>Create tests and implementations and move tag set on to full Recommendation, then move best practices to WG Note</td>
</tr>
</tbody>
</table>
Background
ITS Working Group

Yves Savourel
Felix Sasaki
Sebastian Rahtz
Christian Lieske
Andrej Zydron
Damien Donlon
Martin Dürst
Diane Stoick
Najib Tounsi
Goutham Saha
Poonam Gupta
Francois Richard
Richard Ishida (part time)
The basic idea

Background

The basic idea

Ways to use ITS

Data categories

Summary
The Internationalization Tag Set

The basic idea
Supporting international usage

Characters as ordered in memory:
<br>The title says "מאניפסט ביניים, W3C" in Hebrew.</br>

The title says "W3C" in Hebrew.

Using the bidi algorithm only
The title says "מאניפסט ביניים, W3C" in Hebrew.

This is an example of the need for special markup to support features of certain languages or scripts.

At the top we have a quotation in Hebrew embedded in an English sentence. (The characters are shown in the order they appear in memory.)

The second line shows how we would expect the text to be displayed – with the characters 'W3C' to the left of the embedded quote.

Unfortunately, the Unicode bidirectional algorithm alone is not able to produce this result. The lower line shows the result you would get from the code at the top.
The title says "W3C" in Hebrew.

To achieve the desired result in XHTML you need to add the dir attribute around the quotation, with the value set to rtl. This establishes the directional context for the Hebrew quote, which enables the Unicode bidirectional algorithm to correctly align the words.
At the operator control panel, make sure the printing system is in Make-Ready mode. The MAKE-READY/RUN indicator should not be lit.

Press the START button to sound the horn. The MAKE READY/RUN indicator flashes.

At the third beep, press the START button again. The START indicator remains lit and paper movement begins.

When the web reaches minimum print speed, the test pattern prints.

Press the MAKE-READY/RUN button to place the printing system in Run mode and start printing the live test pages. The MAKE-READY/ RUN indicator should be lit.

This next example shows a need for markup to assist in the localization process.

Here is some documentation that refers to a hard panel. The text on the hard panel will not be translated. This means that the words START and MAKE READY/RUN should not be translated in the documentation.
START drücken, so dass die Hupe ertönt und die Anzeige MAKE READY/RUN blinkt.

This is what a German translation might look like. Note that the uitext element content has not been translated.
The basic idea

Markup to support localization

Press the START button to sound the horn. The MAKE READY/RUN indicator flashes.

<para>
  Press the <uitext>START</uitext> button to sound the horn. The <uitext>MAKE-READY/RUN</uitext> indicator flashes.
</para>

If the markup looks like this, there is nothing to indicate to a translator, translation tool or machine translation system that the words in the uitext elements should be left in English.
Press the START button to sound the horn. The MAKE READY/RUN indicator flashes.

It would help significantly if the translator or the tools were able to see something like a translate attribute with the value set to no to say that the uitext elements should not be translated.
Our final example in this section relates to a best practice rather than additional markup. The caption for the photo has been captured in an attribute value.

The problem with this is that it is now impossible to identify the Japanese text using xml:lang information, nor to apply the kind of bidirectional markup we saw earlier for Middle Eastern languages.
Volcanic eruptions have literally devastated large inhabited areas. During the 1914 eruption of Sakurajima in Kyushu, 687 houses in Kurokami were buried in hot ash. What remained of this shrine gate, previously five meters tall, was left as a reminder.

Kurokami maibutsu gate (腹五社神社黒神埋没鳥居), Sakurajima Island.

A much better approach is to use an element for the caption. You now have the possibility to apply metadata to the text, such as that shown above. You are also, by the way, able to apply abbreviation markup, additional styling, etc.
**Objectives of ITS**

- Ensure that XML tag sets support international use (e.g., bidirectional or ruby markup)
- Ensure that XML tag sets support localization needs (e.g., translation notes, etc.)
- Guide designers of XML tag sets and content developers away from translatability problems.
- Ensure that localizers and format converters can recognize the meaning of tags in a tag set.
In this section we look at local vs. global use of ITS markup.

We will use the idea of describing whether or not text should be translated, and show how you various ways in which you could achieve that using ITS markup.
ITS has a 'data category' called Translate. It can take the values 'yes' and 'no'. A data category is a kind of abstract concept, which is implemented using markup.
<para>
  Press the <uitext>START</uitext> button to sound the horn. The <uitext>MAKE-READY/ RUN</uitext> indicator flashes.
</para>

To illustrate how to apply the values of this data category in markup we will use our previous example of hard panel text that must be left untranslated.
We can express that we don’t want to translate the content of a particular element by using a `local` attribute, on the element itself, ITS suggests that you incorporate the `its:translate` attribute into your schema so that it can be used in this way in valid XML. The attribute can only have the values 'yes' and 'no'.
Ways to use ITS

Global use

<its:rules ... its:version="1.0">
  <its:translateRule selector="//uitext" translate="no"/>
</its:rules>

<para>
  Press the
  <uitext>START</uitext>
  button to sound the horn. The
  <uitext>MAKE-READY/ RUN</uitext>
  indicator flashes.
</para>

There is another way of applying this data category to the uitext elements, without using local markup. You can define **global** rules using elements and attributes defined in ITS. These elements and attributes would typically appear before the body of your content, though this is not required. They could appear anywhere your schema allows.

The example above shows a rule that selects all uitext elements in the document, using an XPath expression, and applies the value of 'no' to them. A processor reading this file that is ITS aware would then able to apply that rule to the content of the document.
We may be faced with a slightly different situation, where the schema already has an attribute that indicates that content should not be translated, but that uses its own attributes or elements to express that. It is not obvious to a processor that the change attribute in the example above fulfills the same role as our its:translate attribute. Using global rules, however, we can associate that attribute with the translate data category.

The markup at the top of the slide shows how you could do this. It selects all elements in the document with a change attribute with the value set to false, and using the translate attribute in the rule, asserts that this is equivalent to its:translate="no". There is then a similar declaration for change="true".

This can be useful for describing legacy content so that a translation process can understand what text to lock for translation.
We will now look at all the data categories that made it into the first version of ITS. We will look at how the concepts associated with that data category can be applied locally, then globally. We will list the various elements and/or attributes that ITS defines to handle that data category, then provide some examples of their use.
Data categories

Translate

Whether the content of an element or attribute should be translated or not.

Default: translate elements, do not translate attributes

In many cases, ITS defines an expectation for the default state. For example, it is assumed that all elements should be translated and all attributes should not be translated unless specified otherwise. This means that you do not need to use its:translate="yes" until you have to override a setting of its:translate="no" higher up the document tree.
We have already seen uses of the Translate data category. In this example we use an its:translate attribute with the value set to no to indicate that the content of the <panelmsg> element should not be translated.
Data categories

Translate

Global use:

<table>
<thead>
<tr>
<th>element</th>
<th>containing:</th>
</tr>
</thead>
<tbody>
<tr>
<td>translateRule</td>
<td>required, value is XPath expression selecting</td>
</tr>
<tr>
<td>selector</td>
<td>nodes to which rule applies</td>
</tr>
<tr>
<td>translate</td>
<td>required, value &quot;yes&quot; or &quot;no&quot;</td>
</tr>
</tbody>
</table>

<its:rules>
  ...
  <its:translateRule translate="yes" selector="//acronym/@title"/>
  ...
</its:rules>

Note: global rules needed to specify translate information for attribute text in content.

This example says that all title attributes on acronym elements in the document should be considered for translation. Remember that the default is that attributes are left untranslated. (Note that acronyms can change during translation, so title attributes that contain the expanded version of the acronym may also need to be translated.)

It is very useful to be able to say whether specific attributes are to be translated or not. This can only be done in ITS using global rules.
Localization Note

Communicate notes to localizers about a particular item of content, eg.
  how to translate something
  contextual usage of something, eg. ‘enabled’ here refers to ‘printer’

Signal to the translator the relative importance of that note.

Default: none
In the example on this slide we locally provide a note about the content of the first data element, and declare that note to be of type 'alert'. The effect of labelling a note with a type is not normatively defined, but as an example, a translation tool could use the 'alert' attribute value to ensure that the translator sees the note before completing the translation.

Further on in the example a note is provided for the second data element. This is just an informative note, so no type information has been given. (The default is 'description'.) A translation tool may make the translator aware that there is a note for this piece of content, but leave it up to the translator to decide whether or not they read it.
## Localization Note

### Global use:

```
<locNoteRule>
  @selector required, value is XPath expression selecting nodes to which rule applies
  @locNoteType required, value "description" or "alert"
  one of:
    <locNote>
      content, the note text
      @locNotePointer value, relative XPath expression to note location
      @locNoteRef value, URI pointing to note location
      @locNoteRefPointer value, relative XPath expression to location of pointer to the note text
```

Copyright © 2005 W3C (MIT, ERCIM, Keio)
Global use:

```xml
<head>
  <its:rules its:version="1.0" its:translate="no">
    <its:locNoteRule locNoteType="alert"
      selector="//msg[@class='DisableInfo']">
      <its:locNote>The variable {0} has three possible values: 'printer',
       'stacker' and 'stapler options'.</its:locNote>
    </its:locNoteRule>
  </its:rules>
</head>

<body>
  ...
  <msg class="DisableInfo">The {0} has been disabled.</msg>
  ...
</body>
```

This example shows global rules that select all msg elements with the class attribute value set to "DisableInfo" and associate them with a note that describes how the message is to be used. It also gives this note a type of 'alert'.

Note, also, that we used an its:translate="no" attribute locally on the ITS global rules markup to indicate that the note itself should not be translated.

[Note that using text such as "The X has been disabled" in this way is very bad practice! For more information about that see the W3C I18n article "Using Composite Messages".]
Data categories

Localization Note

Global use:

<prolog>
   <its:rules its:version="1.0">
      <its:translateRule selector="/msg/notes" translate="no"/>
      <its:locNoteRule locNoteType="description" selector="/msg/data"
      locNotePointer="../notes"/>
   </its:rules>
</prolog>

<body>
   <msg id="FileNotFound">
      <notes>Indicates that the resource file {0} could not be loaded.</notes>
      <data>Cannot find the file {0}.</data>
   </msg>
   <msg id="DivByZero">
      <notes>A division by 0 was going to be computed.</notes>
      <data>Invalid parameter.</data>
   </msg>
</body>

In this example we use the locNoteRule element to select all data elements that have a parent element msg, and point to the location of the note information for that data element. In this case we use a relative XPath expression to say that the note is contained in the notes element that has the same msg parent. You can see the notes and text for translation in the lower part of the example.

Note, also, that we use another global rule to state that all notes elements are to be left untranslated.
Terminology

Mark terms and optionally associate them with information, such as definitions.
Default: not a term
In this example we use local ITS markup to say that the word "motherboard" is a term, and point to its definition.
**Data categories**

**Terminology**

Global use:

```
<termRule> containing:
@ selector required, value is XPath expression selecting nodes to which rule applies
@ term required, value "yes" or "no"
optionally, one of:
  @ termInfoPointer value, relative XPath expression to info location
  @ termInfoRef value, URI pointing to info location
  @ termInfoRefPointer value, relative XPath expression to location of pointer to the information
```
Global use:

```xml
<its:rules its:version="1.0">
  <its:termRule selector="//term" term="yes" termInfoRefPointer="@target"/>
</its:rules>

<p>We may define <term target="#TDPV">discoursal point of view</term> as <gloss xml:id="TDPV">the relationship, expressed through discourse structure, between the implied author or some other addresser, and the fiction.</gloss></p>
</text>
```

This example uses a global rule to say that all the content of all term elements are terms, and that a pointer to the information describing that term is to be found in the target attribute on the term element.
Elements within text

Identify how an element behaves relative to its surrounding text, eg. for text segmentation purposes.

Default: not within text

The Elements Within Text data category is easy to understand if you think of an example of its use. One such use can be to indicate whether specific tags should or should not demarcate the boundaries of translation units. These are the small chunks of text that translation tools use to match against previous translations. The process of splitting the text up into such translation units is typically called segmentation.
In this example we say that b, em, and i elements do not constitute translation unit boundaries.

We use another rule to say that quote tags do not break up the text in which the quote is embedded into multiple translation units, but a translation tool may want to also do source matching on the content of the quote element on its own to try and find previous translations of the text.
Data categories

Directionality

Specify the base writing direction of blocks, embeddings and overrides for the Unicode bidirectional algorithm.

Default: left-to-right text

We have so far been looking at data categories that provide information that is particularly useful for localization of content. The remaining data categories should be included in schemas so that content authors can use those tags set to create content in languages such as those used in the Middle or Far East.
The its:dir attribute is used here locally to express that the directional context of the quote is right-to-left. This information is necessary if we want the text to be displayed correctly. We have already seen this example earlier in the presentation.
In the example on this slide, the content actually has markup to perform the same role as the its:dir attribute. We use a global rule to indicate that all direction attributes with a value of "rtlText" are equivalent to its:dir="rtl". (In an actual implementation, you would probably want to define rules for all possible values of the direction attribute – we kept things simple for the example.)
Ruby (or furigana in Japanese) is commonly used in Far Eastern scripts to provide phonetic transcriptions of the complex ideographic characters where such characters are obscure, or where the reader is assumed to not yet know how to read the character.

An approach to implementing ruby markup in a schema is described by the W3C's Ruby Annotation Specification.
### Ruby

**Local use:**

- `<rb>` required, contents ruby base text
- `<rp>` contains ruby parentheses
- `<rt>` contains the ruby text and an optional `<rspan>`
- `<rb>` ruby base container (for complex ruby)
- `<rtc>` ruby text container (for complex ruby)

The ITS specification provides elements that cover the usage described in the Ruby Annotation specification.

The example of use shows how you would use ITS markup to associate a run of hiragana text with kanji base text in Japanese. It includes markup for ruby parentheses, which provide a useful fallback approach if an application doesn't support ruby text display.
Global use:

<rubyRule> containing:
@ selector required, value is XPath expression selecting nodes containing ruby base text
@ rubyPointer optional, value is XPath expression
@ rpPointer optional, value is XPath expression
@ rbcPointer optional, value is XPath expression
@ rtcPointer optional, value is XPath expression
@ rbspanPointer optional, value is XPath expression
@ rtPointer optional, value is XPath expression
<rubyText> optional, value is ruby text

There are also a set of attributes and an element to point to or provide ruby markup in legacy content. No example is given here.
Language information

Express the language of a given piece of content.

Default: none
Our final data category provides a means of indicating that a legacy tag fulfills the same role as the xml:lang attribute in XML.

There is no local ITS usage for this data category. You should simply use xml:lang for this purpose.

This example says that for any element in the content, the language attribute plays the same role as, and uses the same values as, the xml:lang attribute would.
Summary

Background
The basic idea
Ways to use ITS
Data categories
Summary
Here, once again, we show the set of objectives for ITS that we showed at the beginning of the presentation.
What this means to me

- Provide implementations for Candidate Recommendation phase
- Use the features for localization projects
- Lobby schema designers to incorporate ITS markup, and use external definitions elsewhere
- Additional features in a version 2? It's up to you!

When we enter the Candidate Recommendation phase we will be looking for people to create implementations of ITS so that the specification can continue on to full Recommendation (standard) status.

You should also begin lobbying schema designers and localization tool developers to support ITS.

In this version of ITS we have only covered some of the initial requirements set that was developed by the Working Group. In addition, there may be requirements that we have not so far recognized. If we are to create an ITS version 2, however, we need people to use the current version and show their support for rechartering the ITS Working Group.
Always remember that community involvement is crucial to development of W3C specifications. The W3C does not simply decide in an ivory tower to develop specifications such as ITS and impose them on the public. The process only starts when we have support from the W3C member companies, experts and industry participants who will compose the Working Group. If you feel that this work is valuable, please consider participating in the Working Group.