[*Translating and the Computer 28*, November 2006 (London: Aslib, 2006)]

W3C Internationalization Tag Set - A Gentle Introduction -

Christian Lieske[2]
christian.lieske@sap.com
SAP AG
(www.sap.com)
Dietmar-Hopp-Allee 16
D-69190 Walldorf

Felix Sasaki
fsasaki@w3.org
World Wide Web Consortium
(www.w3.org)
Keio Research Institute at SFC
5322 Endo Fujisawa
Kanagawa
252-8520 Japan

Yves Savourel
ysavourel@translate.com
ENLASO Corp.
(www.translate.com)
4888 Pearl East Circle
Suite 300E
Boulder, CO 80301
United States

Abstract

XML has many built-in capabilities to support the worldwide use of content. Proper use of these capabilities for the purpose of internationalization (i18n) and localization (l10n), however, sometimes requires considerable expertise. This holds especially for developers of XML schemas, and producers of XML instances (such as authors or translators). The Internationalization Tag Set Working Group (ITS WG) of the World Wide Web Consortium (W3C) is working on a standard which makes it easier to create XML which is internationalized and can be localized effectively. The standard has two dimensions: On the one hand, the standard identifies concepts (such as "directionality") which are important for i18n and l10n. On the other hand, the standard defines implementations of these concepts (termed "ITS data categories") as a set of elements and attributes called the Internationalization Tag Set (ITS). This paper explains the ITS motivation/context and history, basic ideas and mechanisms. It provides information on how to use ITS with new as well as with existing XML-based content. Furthermore, the paper depicts some processing contexts (such as transformation for localization purposes) in which ITS can be used.

1 Introduction

This chapter sketches the problem space/context for the Internationalization Tag Set (ITS) Working Group of the World Wide Web Consortium (W3C), and describes the steps which the group so far has taken towards the goal of creating a standard. Furthermore, the chapter explains some general considerations/tasks which may surface when working in the realm of standards. The chapter closes by depicting the users and usage scenarios which the ITS Working Group is considering.

1.1 Overall Context

Content or software that is authored in one language (so-called "source language") is often made available in additional languages or adapted with regard to other cultural aspects. This is done through a process called localization (l10n), where the original material is translated and adapted to the target audience.

---

[1] Many of the concepts in this paper were developed within the W3C i18n ITS (Internationalization Tag Set]) Working Group. Many Working Group participants have contributed to this development. However, this paper may not express the opinion of the working group. The authors of this paper, not the WG, are responsible for any unclearness and mistakes in any parts of this paper.

[2] The author would like to thank his colleague Hendrik Achenbach for feedback on draft versions of this document.

Authoring (in the original language), localization or other content-related processes may need to take into account features of the local languages or scripts. For example, people authoring in languages such as Arabic, Hebrew, Persian or Urdu may need special devices (Unicode control characters or special markup) to demarcate directionality in mixed direction text in order to support for example proper rendering in a Web browser. Figure 1: Incorrect rendering due to missing Devices to Indicate Directionality exemplifies this by incorrectly rendering the position of the Arabic inlay.

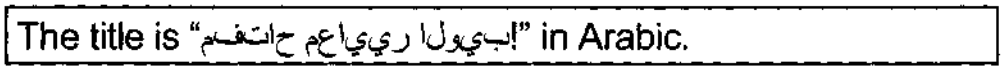

Figure 1: Incorrect rendering due to missing Devices to Indicate Directionality

From the viewpoints of feasibility, cost, and efficiency, it is important that the original material is suitable for localization. This is achieved by a combination of the following:

- Appropriate design and development of capabilities such an XML-based document format; the corresponding process is referred to as internationalization (i18n)

- Proper use of the capabilities during authoring (e.g. use of directionality markup when authoring HTML)

- Language- and style-related quality control

Another way to look at state-of-the-art internationalization and localization[3] of content is the following: It depends on standardized meta-data which is anticipated during the design of processes and formats, and put into place for example by authors.

Important note: ITS defines "data category" as an abstract concept for a particular type of information/meta-data for internationalization and localization of XML schemas and documents. The concept of a data category is independent of its implementation in an XML environment (e.g. using an element or attribute). The introductory sections of this paper use the terms "meta-data" and "data category" as quasi-synonyms. The authors have made all efforts, however, to use "data category" when referring to ITS, since that term is the official one used in the context of ITS.

Several important types of content have undergone interesting changes since the arrival of XML. Examples:

1. Structured documentation has adapted XML-based vocabularies such as the Darwin Information Typing Architecture (DITA; see [DITA])

2. Software-related content such as labels appearing on a Graphical User Interface has adapted XML-based vocabularies such as the extensible User Interface Language (see [XUL])

In cases, challenges and opportunities in the domain of XML internationalization and localization have become visible. Many of them relate to the standardized meta-data mentioned above.

[3] For a detailed explanation of the terms "localization" and "internationalization" (see *[geo-i18n-l10n])*.

1.1.1 Sample Challenges

1. It is easy to create proprietary, non-standard XML formats/vocabularies. The downside of this is the following: For each XML vocabulary, possibly all people, processes and tools have to be made aware of specifics. Examples:

   - Language information may be represented by any combination of attributes such as "lang", "xml:lang", elements such as "locale", "lang", "language", values such as "E", "English", "en", "en-us".

   - Content which may not need to be translated (but, however, has to be part of the content in order to provide for example context information for a translator) may be represented by sundry mechanisms such as a "context" attribute, or even an XML comment.

2. Translation may break code. If for example a button label gets translated (since the XML does not indicate that it should not be touched) the User Interface/application may fail to work (see Figure 2: Sample Code with both Translatable and Untranslatable Text[4]).

```
<!-- Sample XUL file -->
 <window xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
<box align="center">
<!—The "hello xFLy" should not be translated, the "Hello World" should be translated ->
   <button label="hello xFly" onclick="alert('Hello World')";/>
</box>
</window>
```

Figure 2: Sample Code with both Translatable and Untranslatable Text

1.1.2 Sample Opportunities

1. XML-based material can be much more easily twisted than binary content. It is for example easy to extract the value of certain attributes by means of a very basic XSL stylesheet.

2. Information which is helpful for translation/localization purposes often can be embedded or attached to XML-based material. This very often holds for both the specification of an XML vocabulary (e.g. an XSD) and for XML instances.

   Figure 3: Sample XSD with i18n/l10n Information exemplifies a note (related to a uniqueness constraint) for someone working with an XSD. Figure 4: Sample Instance with i18n/l10n Information exemplifies a note (related to uniqueness as well) for a translator working with an XML instance.

```
<xsd:schema                                xmlns:myITS="http://my.org/XML/2005/myITS"
 xmlns:xsd=http://www.w3.org/2001/XMLSchema                elementFormDefault="qualified"
attributeFormDefault="unqualified">
<xsd:element name="book">
<xsd:complexType>
    <xsd:sequence>
       <xsd:element ref="chapter"  maxOccurs="unbounded"/>
   </xsd:sequence>
```

[4] Please note that "untranslatable" in many cases has the semantic of "should not be translated".

```
</xsd:complexType>
</xsd:element>
<xsd:element name="chapter">
   <xsd:complexType>
     <xsd:attribute name='title" use="required"/>
   </xsd:complexType>
<xsd:unique name="titleldAndContent" mylTS:noteToLocEng="Please check that this constraint
can be enforced in our target languages">
<xsd:selector xpath="title"/>
<xsd:field xpath="@id"/>
<xsd:field xpath="."/>
</xsd:unique>
</xsd:element>
</xsd:schema>
```

Figure 3: Sample XSD with i18n/l10n Information

```
<book  xmlns:myITS="http://my.org/XML/2005/mvlTS">
<chapter  title='Table Creation" myITS:noteToTrans="Make sure that all chapters have different
titles"/>
<chapter title="Table Generation"/>
<chapter title="Table Making"/>
</book>
```

Figure 4: Sample Instance with i18n/l10n Information

The challenges and opportunities indicated in the remarks above have a common denominator: a standard, yet flexible set of elements and attributes that can be used with XML material to support the internationalization and localization. Such a set on the one hand can supply universally understood meta-data, and on the other hand can cater for individual needs. This insight has been the starting point for the work of the Internationalization Tag Set Working Group.

1.2 History of the W3C ITS Working Group

The idea that some kind of standardized XML vocabulary was needed to help with the worldwide use of XML dates back at least six years (see [XMLandLoc]). A first big step towards standardization was taken in autumn 2004 when the World Wide Web Consortium (W3C) published the charter for a Working Group (WG) within its Internationalization Activity.

The chartered W3C Working Group, the Internationalization Tag Set (ITS) Working Group (ITS WG), commenced work in February 2005. Three deliverables were targeted:

1. Requirements documents (see the information on the ITS WG home page [ITS WG Home])

2. Specification with corresponding XML markup vocabulary (i.e. the standard, yet flexible set of elements and attributes mentioned above)

3. Document on "Best Practices for XML Internationalization" (containing recommendations such as "Ensure that translatable text is stored in elements rather than attributes").

It is intended to develop the markup vocabulary specification into a Recommendation (i.e. an official W3C standard; see [W3C ReportMatLev] for an explanation of the maturity levels of W3C reports). The first milestone on the way towards that goal was a Working Draft of the specification which was first published in November 2005 (see Figure 5: Announcement of ITS First Public Working Draft on W3C Homepage).

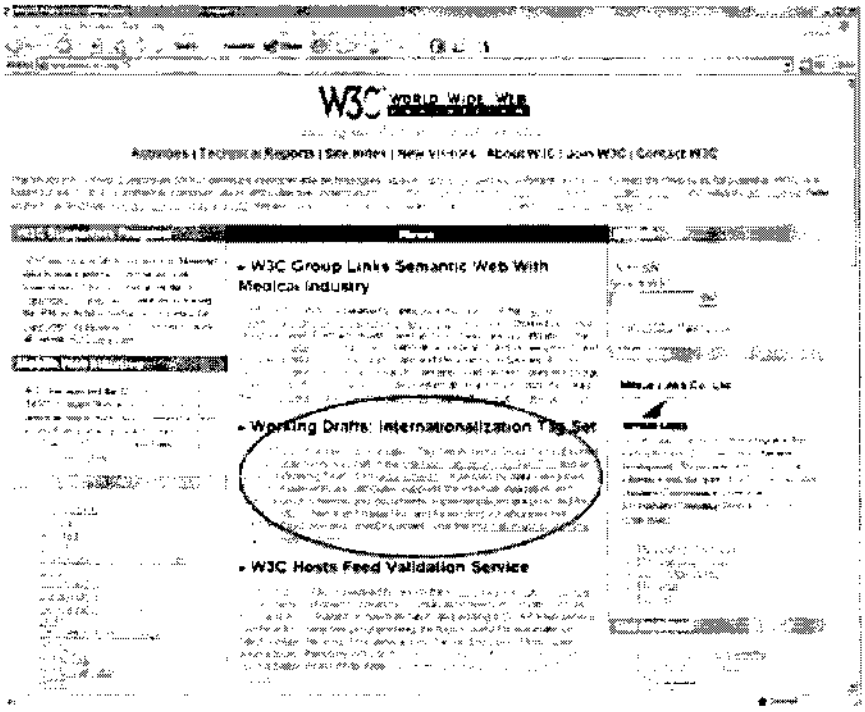

Figure 5: Announcement of ITS First Public Working Draft on W3C Homepage

The Last Call for the Working Draft closed in summer 2006, and the specification is currently moving towards becoming a Candidate Recommendation.

It is important to realize that the ITS WG is not primarily geared towards localization. Rather, the WG addresses internationalization issues, one of them being improving the process of localization. Furthermore, it has to be noted that while the information presented in this article is up-to-date at the time of writing (October 2007), the WG is still working on the specification and the other deliverables. Thus, notation or syntax in the specification may still change. Anyone working with ITS should make sure to refer to the latest WG deliverables, and check their official standing.

1.3 Important Questions and Tasks to Be Addressed

Working towards a set of elements and attributes for i18n/l10n meta-data needs to tackle questions and tasks like the following:

1. Which meta-data is necessary to support i18n and l10n?

   Example:

   note (categorized by roles such as "localization engineer" or "translator")

   ITS has approached this task by means of a set of requirements documents (see [ITS Reg]). Currently, not all of these requirements are addressed in the specification. The ITS Working Group covers some of the requirements in the

"Best Practices for XML Internationalization" and may cover additional ones in a future update or addition to the specification.

2. Which related standards exist?

   Example:

   A Best Current Practice Document (BCP) of the Internet Engineering Task Force (IETF) holds rules related to the syntax of language identifiers [langTags].

3. Which values should the meta-data is allowed to take?

   Example:

   The allowed values for the meta-data category 'original language' are those for 'xml:lang'.

4. How should the meta-data be represented?

   Examples:

   a. In XSD, notes should be represented by an attribute from the MYITS namespace. The attribute prefix shall be 'noteFor', and the part after the prefix should specify the role for which the note is meant (resulting in attribute names like 'mylTS:noteForLocEngineer').

   b. In XML instances, notes should be represented by the element 'note' from the MYITS namespace. The attribute 'type' shall specify the role for which the note is meant (resulting in things like <note type="ForLocEngineer">It is important ...</note>').

5. Which mechanism should be used to attach i18n/l10n-related information to content?

   Example:

   Any of the following mechanisms may be used (cf. mechanisms for attaching rendering-related information with CSS):

   a. Inline/locally:

```
<chapter title="Table Creation"  myITS:noteToTrans="Make sure that all chapters have different titles"/>
```

Figure 6: Attaching Information Inline/Locally

   b. In a special section/globally:

```
<root xmlns:myITS="http://www.w3.org/XML/2005/mylTS">
<loc>
   <mylTS:translate  value ="yes"   target ="\a\b | \\c"/>
</loc>
<body>
   ...
```

```
</body>
</root>
```

Figure 7: Attaching Information in a Special Section/Globally

c. Processing instruction which references information:

```
<?xml-locProps type="text/xsl" href="propsForDocTypeX.xsl"?>
```

Figure 8: Attaching Information by Means of a Processing Instruction

The question of how to attach meta-data is related to the constraint that in some cases minimal invasion (little modification), or even no-invasion (no modification) is called for. Example: Changes to existing XML content may not be feasible.

6. How should information proliferation work?

   Example:

   If 'translate="yes"' is specified on the root element of an XML instance, 'translate="yes" applies to all children unless it is overwritten at the child level.

7. Should there be any defaults?

   Example:

   The defaults for attributes should be 'translate="no"'.

8. How can existing meta-data be reused?

   Example:

   The element name for graphics you have in mind for the specification may be "gr". If people are using a different name (e.g. "inlineGraphics") in their vocabulary, you may want to include something like a mapping feature in your standard.

   The question related to reuse addresses the question how your standard can be used with existing popular markup schemes such as XHTML, DocBook, OpenOffice or DITA. It acknowledges for example the fact that one single markup in one schema language might be insufficient since the standard may need to work in heterogeneous environments:

   - Different types of XML formats (e.g. centered on text like OpenOffice, focused on code like XUL Mozilla, mixing prose and code like DocBook)

   - Existing and new schemas and processes (and for example should not break existing processing chains)

9. Which technologies would you like to consider?

   You may for example mandate that the specification has no dependency on technologies which are still under development, or that it fits with existing work in the W3C architecture.

10. Which "home" would be best to develop your standard?

    Sample "homes" are the World Wide Web Consortium, the Organization for the Advancement of Structured Information Systems (OASIS), the International Standards Organization (ISO).

11. Which rules do you want to enforce with regard to conformance and availability of implementations?

12. Would you like to provide test suites to implementers or bodies involved in conformance testing or even certification?

Working on a standard thus has at least two faces:

- On the one hand, conceptual work is required. For example, you have to figure out which meta-data (termed "data categories" in the context of the ITS WG) is needed for the purpose at hand

- On the other hand, work related to syntax and implementations of this meta-data e.g. as a set of XML elements, attributes and values is required

1.4 Users and Usages of ITS

ITS targets different types of users and usages. In order to support all of them, the information about what markup should be supported to enable worldwide use and effective localization of content is provided by the ITS specification in two ways: abstract in the data category descriptions, and concrete in the ITS schemas.

Out of the many potential user types of ITS, the following are of particular importance: schema developers (experts coding a so-called "host vocabulary"), tools vendors, content producers and information architects.

For the purpose of illustration, we will answer the question, how ITS can be used to indicate that certain parts of content should or should not be translated.

1.4.1 Schema Developers Who Start a Schema from Ground Up

This type of user will use ITS to find proposals for attribute and element names to be included in their new schema. Using the ITS attribute and element names may be helpful because it leads to easier recognition (by both schema users and processors) of the concepts represented. It is perfectly possible, however, for a schema developer to work with a proprietary set of attribute and element names. ITS sets out, first and foremost, to suggest markup that should be available to enable worldwide use and effective localization of content, and that the behavior of that markup meets established needs.

1.4.2 Schema Developers Who Work with an Existing Schema

This type of user will be working with existing schemas such as DocBook, DITA, or perhaps a proprietary schema. Developers working on this type of schema should check whether their schemas already support the ITS markup, and, where appropriate, add ITS markup to their schema.

In some cases, an existing schema may already contain markup equivalent to that recommended in ITS. In this case, it is not necessary to add duplicate markup since ITS provides mechanisms for relating ITS markup with markup in the host vocabulary which serves a similar purpose. Before using these mechanisms, however, the

developers should check that the behavior associated with the markup in their own schema is fully compatible with the expectations described in the ITS specification.

1.4.3 Vendors of Content-related Tools

This type of users encompasses providers of tools for authoring, translation or other flavors of content-related software solutions. It is important to ensure that such tools enable worldwide use and effective localization of content. For example, translation tools should prevent content marked up as not for translation from being changed or translated.

It is expected that ITS will make the job of vendors easier by standardizing the format and processing expectations of certain relevant markup items, and allowing them to more effectively identify how content should be handled.

1.4.4 Content Producers and Architects

This type of users comprises authors, translators and other types of content authors. They may use ITS to mark up specific bits of content.

Aside: ITS provides mechanisms for removing the burden of inserting markup from content producers by relating the ITS information to relevant bits of content in a global manner by means of a global, rule-based approach. This global work, however, may fall to information architects or localization engineers, rather than the content producers themselves.

A content producer may use an attribute on a particular element to say that the text in the element should not be translated (see Figure 9: Attribute Used by a Content Author).

```
<book>
<head>...</head>
<body>
  <p>And he said: you need a new
    <quote    its:translate= "no" >T-Model</quote>
  < /p>
</body>
</book>
```

Figure 9: Attribute Used by a Content Author

A content author or information architect may use markup at the top of an XML instance to identify a particular type of element or context in which the content should not be translated (see Figure 10: Rule Used by Content Architect).

```
<text>
<head>
<its:rules xmnls:its=  http://www.w3.org/2005/11/its">
<its:translateRule its:translate=”no” its:selector="//dt/>
<its:rules>
</head>
<body>...
<p>... <dl><dt>...</dt><dd>...</dd></dl></p>
</body>
</text>
```

Figure 10: Rule Used by Content Architect

The examples above can be likened to the use of CSS in XHTML: Using a "style" attribute, an XHTML content author may assign a color to a particular paragraph. That author could also have used the style element at the top of the page to say that all paragraphs of a particular class or in a particular context would be colored red.

2 Basic Concepts

This chapter explains which meta-data the ITS markup currently covers. In addition, it introduces the core principles of ITS. Please refer to the specification for (see [ITS Spec] for details).

2.1 Choice of Meta Data

Determining adequate meta-data may be conceived, as indicated in the preceding chapter, as one of the generic tasks related to the development of a standard. The ITS WG derived possible meta-data for i18n/l10n from an ITS-specific requirements document (see [ITS Req]). Not all requirements listed there resulted in meta-data for the first version of the ITS markup vocabulary[5]. Those which are not addressed are either covered in the WG document "Best Practices for XML Internationalization" (see [XML i18n BP]) or may be addressed in a future version of ITS.

Meta-data which ITS currently addresses questions such as the following:

1. What is the language that is being used in the content?

2. Do specific parts of the content need special attention since they are terms (and thus may require a very specific translation)?

3. Are some parts of the content written in a specific script (and thus may for example require rendering from right-to-left rather than left-to-right)?

4. Which types of markup do not affect linguistic integrity (and thus should not affect text flow)?

5. Is there some content which does not need to be translated or even must not be translated?

The ITS WG defines "data categories" as an abstract notion for meta-data for internationalization and localization of XML schemas and documents. In line with the questions mentioned above, the ITS WG thus has created data categories for concepts such as:

1. Identifying language/locale

2. Identifying terms

3. Supporting bidirectional text

4. Indicating translatability

5. Carrying localization notes

6. Providing annotation/Ruby markup

[5] The Working Group needed to address the most important data categories first.

7. Indicating segmentation

2.2 Core Principles

ITS recommends special purpose meta-data and suggests standard mechanisms for this meta-data. These standard mechanisms amongst other define how the meta-data can be attached to content. Furthermore, the mechanisms mandate defaults and state if and how meta-data can be overridden or inherited. ITS meta-data can be stored in a standalone file, or within an existing document (via the XML namespace mechanism). In some cases, existing information such as a set of notes for a localizer can be reused by ITS. Since each usage defines some specific requirements, ITS markup nevertheless may take different shapes.

2.2.1 Selection

Information (e.g. "translate this") captured by ITS markup (e.g. 'its:translate="yes"') always pertains to one or more element or attribute nodes. In a sense, ITS markup "selects" the XML node(s). Selection may be explicit or implicit, a distinction captured by ITS' two approaches to selection: local, and with global rules.

The two approaches cater for different needs and user types:

- Users like content authors need for example a simple way to work with the translatability data category in order to express whether the content of an element or attribute should be translated or not.

- Users like localization coordinators need an efficient way for managing translations of large document sets based on the same schema. This could be realized by a specification of defaults for translatability and exceptions from the defaults (e.g. all p elements should be translated, but not p elements inside of an index element).

The mechanisms defined for ITS selection resemble those defined for Cascading Stylesheets (CSS, see [CSS2]). The local approach can be compared to the "style" attribute in CSS, and the approach with global rules is similar to the "style" element in CSS.

In contrast to CSS, ITS uses XPath for identifying nodes. While the local approach puts ITS markup in the relevant element of the host vocabulary (e.g. the "author" element in DocBook) the rule-based, global approach puts the ITS markup in elements defined by ITS itself (namely the "rules" element).

The global, rule-based approach has the following benefits:

- Content authors do not have to concern themselves with creating additional markup or verifying that the markup was applied correctly. ITS data categories are associated with sets of XML nodes (for example all "p" elements in an XML instance)

- Changes can be done in a single location, rather than by searching and modifying the markup throughout a document (or documents, if the "rules" element is stored as an external entity)

- Attributes as well as elements can be selected

- It is possible to associate ITS markup with existing markup (for example the "term" element in DITA)

The difference between the local and global approach is fleshed out below by means of examples.

2.2.1.1 Example for the Local Approach

The document in Figure 11: ITS Markup on Elements in an XML Document (Local Approach) shows how a content author may use the ITS "translate" attribute to indicate that all content inside the "author" element should be protected from translation. Translation tools that are aware of the meaning of this attribute can then screen the relevant content from the translation process.

```
<dbk:article
xmlns:its="http://www.w3.org/2005/11/its"
xmlns:dbk="http://docbook.org/ns/docbook"
its:version="1.0">
<dbk:info>
   <dbk:title>An example article</dbk:title>
<dbk:author its:translate="no">
<dbk:personname>
   <dbk:firstname>John</dbk:firstname><dbk:sumame>Doe</dbk:sumame>
</dbk:personname>
<dbk:affiliation>
   <dbk:address><dbk:email>foo@example.com</dbk:email></dbk:address>
</dbk:affiliation>
</dbk:author>
</dbk:info>
</dbk:article>
```

Figure 11: ITS Markup on Elements in an XML Document (Local Approach)

For this to work, the schema developer will need to add the "translate" attribute to the schema as a common attribute or on all the relevant element definitions. Note how there is an expectation in this case that inheritance plays a part in identifying which content does have to be translated and which does not. Tools that process this content for translation will need to implement the expected inheritance.

2.2.1.2 Example for the Global Approach

The document in Figure 12: ITS Global Markup in an XML Document (Global, Rule-based Approach) shows a different approach to identifying non-translatable content, similar to that used with a "style" element in XHTML (see [XHTML 1.0], but using an ITS-defined element called "rules". It works as follows: A document can contain a "rules" element (placed where it does not impact the structure of the document, like in a "head" section). The "rules" contains one or more ITS rule elements (for example "translateRule"). Each of these specific elements contains a "selector" attribute. As its name suggests, this attribute selects the XML node or nodes to which a corresponding ITS information pertains. The values of ITS selector attributes are XPath absolute location paths. Information for the handling of namespaces in these path expressions is contained in the ITS element."ns" which is a child of "rules".

```
<myTopic
xmlns:its= "http://www.w3.org/2005/11/its" id="topic01" xml:lang="en-us">
<prolog>
<title>Using ITS</title>
<its:rules  its:version="1.0">
<its:ns prefix="n" uri="myNamescapeURI"/>
```

```
    <its:translateRule selector="//n:term" translate="no"/>
</its:rules>
</prolog>
<body><p>ITS defines <term>data category</term> as an abstract concept for a particular type of information for internationalization and localization of XML schemas and documents.</p></body>
</myTopic>
```

Figure 12: ITS Global Markup in an XML Document (Global, Rule-based Approach)

For this approach to work, the schema developer needs to add the "rules" element and associated markup to the schema.

In some cases this may allow the schema developer to avoid adding other ITS markup (such as a "translate" attribute) to the elements in the schema. However, it is likely that authors will want to use attributes on markup from time to time to override the general rule.

For the specification of the "translate" data category information, the contents of the "rules" element would normally be designed by an information architect familiar with the document format and familiar with, or working with someone familiar with, the needs of the localization group.

The commonality in both examples above is the markup 'translate="no"'. This piece of ITS markup can be interpreted as follows:

- it pertains to the "translate" data category
- the attribute "translate" holds a value of "no"

2.2.2 Information Proliferation

The power of the ITS selection mechanisms comes at a price: rules related to defaults, overriding/precedence, and inheritance, are needed. These rules may best be explained by way of an example (see Figure 13: Overriding and Inheritance) related to the "translate" data category.

```
<text  xmlns:its="http://www.w3.org/2005/11/its">
<head>
  <revision>Sep-10-2006 v5</revision>
  <author>Ealasaidh Mclan</author>
  <contact>ealasaidh@hogw.ac.uk</contact>
  <title its:translate="yes">The Origins of Modem Novel</title>
  <its:rules its:version="1.0">
    <its:translateRule translate="no" selector= /text/head"/>
  </its:rules>
</head>
<body>
<div xml:id="intro">
<head>lntroduction</head>
<p>lt would certainly be quite a <span its:translate="no">faux pas</span> to start a dissertation on the origin of modem novel without mentioning the <tl>Epic of Gilgamesh</tl>...</p>
</div>
</body>
</text>
```

Figure 13: Overriding and Inheritance

Some notes on the example above:

1. By default elements are translatable.

2. A "translateRule" element declared in the header overrides the default for the "head" element inside "text" and for all its children.

3. Because the "title" element is actually translatable/needs to be translated, the global rule needs to be overridden by a local "its:translate='yes'".

The inheritance part related to proliferation may be compared to the cascadation rules in CSS for which also relationships between inline style attributes, style elements and possibly external files have been specified.

2.2.3 Rules Files

One good thing about ITS rules is the following: They can easily be reused. Thus, it becomes easy to apply rules to collections of documents. The information proliferation mechanisms sketched above allow, however, to "cancel" rules on the level of an individual document or even an individual XML node.

Two steps are necessary for easy reuse of rules:

a. create a file which contains the rules (see Figure 14: External Rules File (with Meta-Data for Rules))

```
<myFormatInfo xmlns:its=”http://www.w3.org/2005/11/its" >
<desc>ITS rules used by the Open University</desc>
<hostVoc>http://www.tei-c.org/ns/1.0</hostVoc>
<rulesId>98ECED99DF63D511B1250008C784EFB1</rulesId>
<rulesVersion>v 1.81 2006/03/28 07:43:21 </rulesVersion>...
<its:rules its:version="1.0">
  <its:translateRule selector= "//header" translate="no">
  <its:translateRule selector= "//term" translate="no"/>
  <its:termRule selector="//term" term="yes"/>
  <its:withinTextRule  withinText="yes" selector="//term | //b"/>
</its:rules>
</myFormatInfo>
```

Figure 14: External Rules File (with Meta-Data for Rules)

b. associate the rules file with content by doing one of the following

   - link to an external rules file using the XLink "href attribute, as shown in

```
<myDoc xmlns:its="http://www.w3.org/2005/11/its" xmlns:xlink="http://www.w3.org/1999/xlink">
<header>
<its:rules its:version="1.0" xlink:href="EX-link-external-rules-1.xml"></its:rules>
<author>Theo Brumble</author>
<lastUpdate>Apr-01-2006</lastUpdate>
</header>
<body>
```

```
<p>A <term>Palouse horse</term> has a spotted coat.</p>
</body>

</myDoc>
```

Figure 15: Link to External Rules File

- Use a tool-specific mechanism (for example, for a command-line tool: provide the paths of both the XML document to process and its corresponding rules file).

2.2.4 Adding Information or Pointing to Existing Information

Very often, data redundancy is bad or at least cumbersome. For some data categories, ITS thus provides special attributes to add or point to information about the selected nodes. These rules may best be explained by way of an example (see the examples below) related to the "localization note" data category which allows to add note data to contents.

The note data can take one of four different forms:

1. A "its:locNote" element inside "its:LocNoteRule" element.

2. A pointer to a node that contains the note. This is done with the "locNotePointer" which holds an XPath expression relative to the position of the node selected by the "selector" attribute. This is very handy for XML formats that have their own notes and comments constructs.

3. A reference to the note. That reference must be an Internationalized Resource Identifier (IRI). For example, the location of an external file. This is done with the "locNoteRef" attribute.

4. A pointer to a node that contains a reference to the note. This is done with a "locNoteRefPointer" attribute, which holds an XPath expression relative to the position of the node selected by the "selector" (just like "locNotePointer"). But the content of the pointed node contains an IRI reference to the note instead of the note itself (just like "locNoteRef").

   You can also use localization notes within the document itself. You can use either the "its:locNote" attribute to hold a note that applies to the element where the attribute declared, or the "its:locNoteRef attribute to store an IRI referencing the note. In addition, the "its:locNoteType" attribute can be used to indicate the type of note ("description" or "alert", the first being the default).

The functionalities of adding information and pointing to existing information are mutually exclusive. That is to say, attributes for pointing and adding must not appear at the same "rule" element.

2.2.4.1 Example: "locNote" Element

The "locNoteRule" element associates the content of the "locNote" element with the message with the identifier "DisableInfo" and flags it as important. This would also work if the rule was in an external file, allowing to provide notes without modifying the source document.

```
<myRes xmlns:its="http://www.w3.org/2005/11/its" >
<head>
<its:rules its:version="1,0" its:translate="no">
        <its:locNoteRule locNoteType="alert"selector="//msg[@id='DisableInfo']">
   <its:locNote>The variable {0} has three possible values: 'printer', 'stacker' and 'stapler
options'.</its:locNote>
   </its:locNoteRule>
</its:rules>
</head>
<body>
   <msg id="DisableInfo">The {0} has been disabled.</msg>
</body>

</myRes>
```

Figure 16: "locNote" Element

2.2.4.2 Example: locNotePointer Attribute

The "locNotePointer" attribute is a relative XPath expression pointing to a node that holds the note.

```
<Res xmlns:its="http://www.w3.org/2005/11 /its" >
<prolog>
<its:rules its:version="1.0">
  <its.translateRule selector="//msg/notes" translate="no"/>
  <its locNoteRule locNote="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>
</Res>
```

Figure 17: "locNotePointer" Attribute

2.2.4.3 Example: "locNoteRef Attribute

The "locNoteRule" element specifies that the message with the identifier 'NotFound' has a corresponding explanation note in an external file. The URI for the exact location of the note is stored in the "locNoteRef" attribute.

```
<myRes xmlns:its="http://www.w3.org/2005/11/its">
<head>
<its:rules its:version="1.0">
  <its:locNoteRule  locNoteType="description"    selector= "//msg[@id-NotFound']"
locNoteRef="ErrorsInfo.html/#NotFound"/>
</its:rules>
```

```
</head>
<body><msg id="NotFound">Cannot find {0} on {1}.</msg></body>
</myRes>
```

Figure 18: "locNoteRef Attribute

2.2.4.4 Example: "locNoteRefPointer" Attribute

The "locNoteRefPointer" attribute contains a relative XPath expression pointing to a node that holds the URI referring to the location of the note.

```
<data xmlns:its="http://www.w3.org/2005/11/its" >
<prolog>
<its:rules its:version="1.0">
  <its:locNoteRule locNoteType="description" selector="//data"
locNoteRefPointer="../@noteFile"/>
</its:rules>
</prolog>
<body>
<string id="FileNotFound" noteFile="Comments.html/#FileNotFound">
<data>Cannot find the file {0}.</data>
</string>
<string id="DivByZero" noteFile="Comments.html/#DivByZero">
<data>Invalid parameter.</data>
</string>
</body>

</data>
```

Figure 19: "locNoteRefPointer" Attribute

The local "its:locNote" attribute is the only occurrence where ITS uses an attribute to hold potentially translatable text (because, after all, translation notes can be translatable content). Storing translatable text in an attribute is normally frowned upon for many valid reasons, but in this case, the need for the ITS markup to not interfere with the rest of the document structure (as it would if it were an element) outweighs this internationalization consideration.

3 ITS in Action

This chapter shows how to use ITS in two important usage scenarios: enhancing/complementing material based on standards for XML-based content, and preparing material for streamlined translation.

3.1 Enhancing/Complementing Standards-based Content

An important usage scenario for ITS is related to material based on standards for XML-based content. In this scenario, ITS enhances/complements well-known vocabularies like those developed under the auspices of the Organization for the Advancement of Structured Information Systems (OASIS). Popular examples of these vocabularies are the Darwin Information Typing Architecture (DITA), DocBook and Open Document.

When applying ITS in this usage scenario, especially two types of general considerations pertaining to the relationship between ITS and a so-called "host vocabulary" like DITA need attention:

1. Distribution of labor: It has to decided which mechanisms (e.g. encoding of language identifiers) should be taken from ITS, and which should be taken from the host vocabulary

2. Semantics and mechanisms of the host vocabulary: It is necessary to check that the semantics and mechanisms (e.g. defaults and inheritance) of the two vocabularies which are involved work together properly

3.1.1 General Considerations

Four questions need to be addressed when applying ITS to a host vocabulary. They will be exemplified using DITA:

1. What is the benefit of applying ITS?

Sample answer: Add a missing data category such as Ruby

2. How should or could ITS be applied?

Sample answer: Use the local approach for supplying ITS markup

3. Should or could existing markup be "reused"?

Sample answer: Stick to DITA's "translate" attribute rather than introduce "its:translate"

4. Which caveats exist?

Sample answer: DITA's information proliferation works not in all cases like that of XML 1.0

In the context of the DITA, ITS can enhance or complement as follows:

1. Add meta-data such as script directionality and Ruby annotations

2. Designate translatable content more easily or more complete (see Figure 20: Using ITS to indicate translatability of DITA elements and attributes)

```
<!- Translatable attribute (some are deprecated) -->
<its:translateRule selector="//@alt" translate="yes"/>
<its:translateRule selector="//topicgroup/@navtitle" translate="yes"/>
<!- Non-translatable elements -->
<its:translateRule selector="//draft-comment//*" translate="no"/>
<its:translateRule selector= "//draft-comment/descendant-or-self::*/@*" translate="no"/>
```

Figure 20: Using ITS to indicate translatability of DITA elements and attributes

3. Centralize meta-data such as which DITA elements mark terms, or which elements do not affect linguistic integrity (see Figure 21: Centralizing Meta-data for DITA with ITS)

```
<|-- Terminology -->
<its:termRule selector="//term //dt/>
<|-- Elements within text (inline) -->
<its:withinText withinText="yes"
```

```
selector=" //keyword | //b | //i | //sub | //sup |..." />
```

Figure 21: Centralizing Meta-data for DITA with ITS

A commonality of the examples above is the following: The association between ITS markup and the DITA markup is made in a somewhat private fashion. By contrast, possible public associations are the following:

1. The DITA community is working on updates to DITA. ITS might become part of such an update.

2. The DITA community is constantly generating custom versions of DITA (e.g. for e-learning material) which use DITA's specialization, customization, and generalization mechanisms. ITS might be the domain of such a custom version.

Several caveats are obvious when applying ITS to DITA:

1. DITA's specialization, customization, generalization mechanisms have to be compared carefully with the precedence, inheritance, and defaults defined in ITS

2. Inclusion in DITA is handled by means of the proprietary "conref' mechanism and may not easily fit with ITS' selection

3. Proliferation rules (e.g. for language information) are defined between DITA maps and other types of DITA objects (such as DITA concepts)

3.1.2 Applying ITS to DITA: A Hands-on Example

Let's flesh out the general considerations of the preceding section, and turn a hands-on example which shows how to apply ITS to DITA.

DITA already offers the "translate" attribute to specify whether an element is translatable or not. Thus, we are faced with a question which relates to the distribution of labor between the host vocabulary DITA and ITS: Should we replace DITA's "translate" attribute by "its:translate"? A possible answer to this question is: Use DITA's "translate" attribute as it was intended, and, in addition, make sure that an ITS-enabled tool can see this attribute as an equivalent of "its:translate". This can be done easily (see Figure 22: Routing DITA Translatability to ITS).

```
<its:rules xmnl:its="http://www.w3.org/2005/11/its"   its:version="1.0">
   <its:translateRules selector="//*[translate='no']" translate="no"/>
   <its:translateRules selector="//*[translate='yes']" translate="yes"/>
</its:rules>
```

Figure 22: Routing DITA Translatability to ITS

The two rules simply state that any element with a DITA "translate" attribute set to "no" is not translatable, and conversely for translate="yes".

DITA also already offers elements (for example "term" and "dt") to specify that a certain string is a term. Thus, there is no urgent need to work with "its:term". Rather, we can decide to do the same as in the "translate" case: Use what DITA is offering, and create a rule which allows ITS-enabled tools to see it as an equivalent of "its:term" (see Figure 23: Specifying DITA Elements as ITS Terms).

```
<its:termRule selector=”//term | //dt"/>
```

Figure 23: Specifying DITA Elements as ITS Terms

A sample DITA "topic" (one of the higher-level DITA object types) with the rules defined above only slightly differs from an ordinary DITA topic (see Figure 24: Inserting ITS Rules in a DITA Topic).

```
<topic xmlns:its="http://www.w3.org/2005/11 /its" id="myTopic">
<title>The ITS Topic</title>
<prolog>
<its:rules its:version="1.0">
   <its:translateRule selector="//*[@translate='no']" translate="no"/>
   <its:translateRule selector="//*[@translate='yes']" translate="yes"/>
    <its:termRule selector="//term | //dt"/>
</its:rules>
</prolog>
<body>
<dl>
<dlentry id="tDataCat">
   <dt>Data category</dt>
    <dd>ITS defines <term>data category</term> as an abstract concept for
a particular type of information related to internationalization and
localization of XML schemas and documents.</dd>
</dlentry>
</dl>
<p><ph translate="no" xml:lang="fr">Et voila !</ph>.</p>
</body>
</topic>
```

Figure 24: Inserting ITS Rules in a DITA Topic

3.2 ITS and translation

Translation of XML-based content very often can either work on the native format, or on an intermediate/interchange format. In both cases, ITS can be used to prepare the content for streamlined translation.

3.2.1 ITS and Native Format Translation

If used with a native format like DITA, ITS can enhance or complement the translation-related features of the host vocabulary. A file with ITS rules can for example carry information which is needed in order to configure XML-aware localization tools. To be specific, the "element within text" data category of ITS can capture the information which tags are "internal" in terms of the SDL/Trados TagEditor (an editor which translators use).

3.2.2 ITS and Translation Interchange Format

If used with an interchange format like the XML Localization Interchange File Format (XLIFF), ITS information can be used in filters which extract content from the native file format (see Figure 25: Using the Extract&Merge Paradigm in Translation Processes).

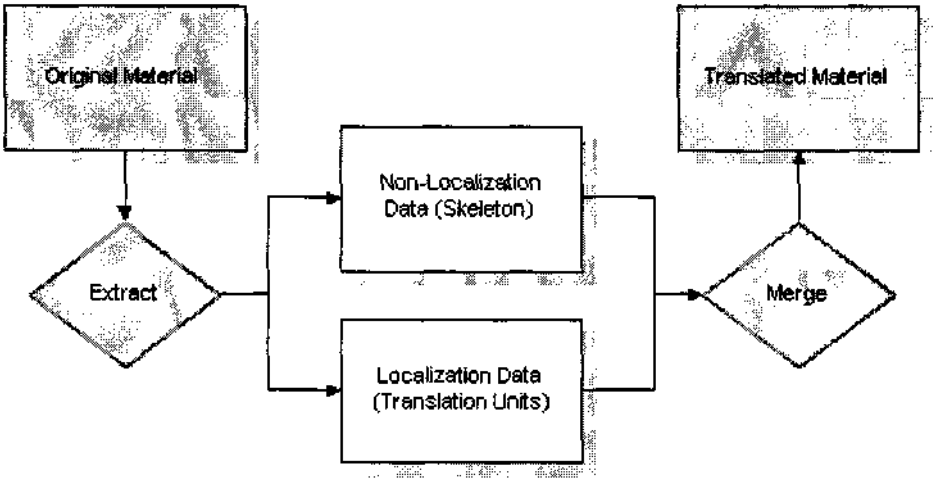

Figure 25: Using the Extract&Merge Paradigm in Translation Processes

The "Elements within Text" data category of ITS can for example capture the information that the content of a certain element belongs to the flow of its parent element, and thus should be placed in the same XLIFF "source" element as the content of the parent element (see Figure 26: Indicating Text Flow in DITA with ITS and Figure 27: Representing Text Flow Correctly in XLIFF based on ITS).

```
<concept id="myConcept" xml:lang="en-us">
<prolog>
<its:rules xmlns:its="http://www.w3.org/2005/11/its" its:version="1.0">
   <its:withinTextRule  withinText="yes" selector="//term"/>
</its:rules>
</prolog>
<title>Horses around the World</title>
<conbody>
   <p>A <term>Palouse horse</term> is the same as an Appalosa.</p>
</conbody>
</concept>
```

Figure 26: Indicating Text Flow in DITA with ITS

```
<xliff xmlns="urn:oasis:names:tc:xliff:document:1.2"   xmlns:okp="okapi-framework:xliff-extensions"
version="1.2">
<file original="\8C8B41275DFACDE.xml" source-language="en" target-language="fr"
 datatype="xml" okp:settings="okf_xml" okp:encoding="utf-8">
<header>
   <tool tool-id="-1397714189" tool-name="Okapi.Utilities.Set03" tool-version="1.0.1.0"/>
</header>
<body>
<trans-unit id="1">
   <source xml:lang="en">Horses around the World</source>
    <target xml-lang="fr"> Horses around the World </target>
</trans-unit>
<trans-unit  id="2">
   <source xml:lang="en">A      <bpt      id="1">&lt;term></bpt>Palouse      horses<ept
id="1"&it;/term></ept> is the same as an Appalosa.</source>
    <target xml:lang="fr">A      <bpt      id="1">&lt;term></bpt>Palouse      horses<ept
id="1">&lt;/term></ept> is the same as an Appalosa.</target>
</trans-unit>
</body>
```

```
</file>
</xliff>
```

Figure 27: Representing Text Flow Correctly in XLIFF based on ITS

4 References/Further Reading

Much of the information about referenced work (indicated by [ ]) is available on the home page of the ITS WG (see http://www.w3.org/International/its/).

[ITS Spec] http://www.w3.org/TR/its

[XMLandLoc] "XML Technologies and the Localization Process", Multilingual #35, Volume 11, Issue 7, pages 62-67

[geo-i18n-l10n] http://www.w3.org/International/questions/qa-i18n

[langTags] http://www.w3.org/International/articles/language-tags/Overview.en.php

[W3C ReportMatLevel] http://www.w3.org/2005/10/Process-20051014/tr.html#maturity-levels

[ITS WG Home] http://www.w3.org/International/its/

[DITA] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=dita

[XUL] http://www.xulplanet.com/

[ITS Req] http://www.w3.org/TR/itsreq/

[XML i18n BP] http://www.w3.org/TR/xml-i18n-bp/

[CSS2] http://www.w3.org/TR/CSS21/

[XHTML 1.0] http://www.w3.org/TR/xhtml1/

[XLIFF] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xliff