[contents]

Internationalization Tag Set (ITS) Version 1.0

W3C Recommendation 03 April 2007

This version:
http://www.w3.org/TR/2007/REC-its-20070403/
Latest version:
http://www.w3.org/TR/its/
Previous version:
http://www.w3.org/TR/2007/PR-its-20070226/

Editors:
Christian Lieske, SAP AG
Felix Sasaki, W3C

Please refer to the **errata** for this document, which may include some normative corrections.

See also **translations**.

This document is also available in these non-normative formats: ODD/XML document, self-contained zipped archive, and XHTML Diff markup to publication from 26 February 2007.

Abstract

This document defines data categories and their implementation as a set of elements and attributes called the *Internationalization Tag Set (ITS)*. ITS is designed to be used with schemas to support the internationalization and localization of schemas and documents. An implementation is provided for three schema languages: XML DTD, XML Schema and RELAX NG.

Status of this Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

This document defines data categories and their implementation as a set of elements and attributes called the Internationalization Tag Set (ITS). ITS is designed to be used with schemas to support the internationalization and localization of schemas and documents. An implementation is provided for three schema languages: XML DTD, XML Schema and RELAX NG.

This is a Recommendation of the W3C. It has been developed by the W3C Internationalization Tag Set (ITS) Working Group, which is part of the Internationalization Activity.

This document has been reviewed by W3C Members, by software developers, and by other W3C groups and interested parties, and is endorsed by the Director as a W3C Recommendation. It is a stable document and may be used as reference material or cited from another document. W3C's role in making the Recommendation is to draw attention to the specification and to promote its widespread deployment. This enhances the functionality and interoperability of the Web.

Please make comments about this document using W3C's public Bugzilla system. We recommend using Bugzilla for making comments (instructions can be found at How to use the Issues Tracking System for the ITS Tagset Working Draft). If this is not feasible, comments may also be sent to www-i18n-comments@w3.org. Use "[Comment on ITS WD]" in the subject line of your email. A list of ITS tagset related comments and issues in Bugzilla and the www- i18n-comments archives are publicly available.

This document incorporates minor changes made against the Proposed Recommendation of 21 November 2006; please see the Revision Log for details. The implementation report and test suite provide further implementation information.

This document was produced by a group operating under the 5 February 2004 W3C Patent Policy. W3C maintains a public list of any patent disclosures made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains Essential Claim(s) must disclose the information in accordance with section 6 of the W3C Patent Policy.

Table of Contents

Appendices

---

1 Introduction

This section is informative.

ITS is a technology to easily create XML which is internationalized and can be localized effectively. On the one hand, the ITS specification identifies concepts (such as "directionality") which are important for internationalization and localization. On the other hand, the ITS specification defines implementations of these concepts (termed "ITS data categories") as a set of elements and attributes called the Internationalization Tag Set (ITS). The document provides implementations for three schema languages: XML DTD [XML 1.0], XML Schema [XML Schema] and RELAX NG [RELAX NG].

This document aims to realize many of the ideas formulated in [Localizable DTDs].

Requirements for this document are formulated in [ITS REQ]. Not all requirements listed there are addressed in this document. Those which are not addressed here are either covered in [XML i18n BP] or may be addressed in a future version of this specification.

This document covers the following requirements:

- R002 - span-like element, see span
- R006 - identifying language/locale, see Section 6.7: Language Information
- R007 - identifying Terms, see Section 6.4: Terminology
- R008 - purpose specification/mapping, see Section 5.5: Associating ITS Data Categories with Existing Markup
- R011 - bidirectional text support, see Section 6.5: Directionality
- R012 - indicator of translatability, see Section 6.2: Translate
- R014 - limited impact, see Section 5.5: Associating ITS Data Categories with Existing Markup
- R017 - localization notes, see Section 6.3: Localization Note
- R020 - annotation markup, see Section 6.6: Ruby
- R025 - elements and segmentation, see Section 6.8: Elements Within Text

The following requirements will be addressed in [XML i18n BP]:

- R003 - CDATA Section
- R004 - Unique Identifier
- R005 - Handling of Entities
- R015 - Attributes and Translatable Text
- R016 - Naming Scheme
- R019 - Multilingual Documents
- R022 - Nested Elements

The Working Group decided not to cover the following requirements at this time to be able to focus on the most important ones.

- R001 - Indicator of Constraints
- R009 - Content Style
- R010 - Link to Internal/External Text
- R013 - Metrics Count
- R018 - Handling of White-Spaces
- R021 - Identifying Date and Time
- R023 - Linguistic Markup
- R024 - Variables
- R026 - Associated Objects

1.1 Motivation for ITS

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, where the original material is translated and adapted to the target audience.

In addition, document formats expressed by schemas may be used by people in different parts of the world, and these people may need special markup to support the local language or script. For example, people authoring in languages such as Arabic, Hebrew, Persian or Urdu need special markup to specify directionality in mixed direction text.

From the viewpoints of feasibility, cost, and efficiency, it is important that the original material should be suitable for localization. This is achieved by appropriate design and development, and the corresponding process is referred to as internationalization. For a detailed explanation of the terms "localization" and "internationalization", see [l10n i18n].

The increasing usage of XML as a medium for documentation-related content (e.g. DocBook and DITA as formats for writing structured documentation, well suited to computer hardware and software manuals) and software-related content (e.g. the eXtensible User Interface Language [XUL]) creates challenges and opportunities in the domain of XML internationalization and localization.

1.1.1 Typical Problems

The following examples sketch one of the issues that currently hinder efficient XML-related localization: the lack of a standard, declarative mechanism which identifies which parts of an XML document need to be translated. Tools often cannot automatically do this identification.

Example 1: Document with partially translatable content

In this document it is difficult to make distinction between the `string` elements that are translatable and the ones that are not. Only the addition of flags could resolve the issue.

```
<resources>
 <section id="Homepage">
  <arguments>
   <string>page</string>
   <string>childlist</string>
  </arguments>
  <variables>
```

```
   <string>POLICY</string>
   <string>Corporate Policy</string>
  </variables>
  <keyvalue_pairs>
   <string>Page</string>
   <string>ABC Corporation - Policy Repository</string>
   <string>Footer_Last</string>
   <string>Pages</string>
   <string>bgColor</string>
   <string>NavajoWhite</string>
   <string>title</string>
   <string>List of Available Policies</string>
  </keyvalue_pairs>
 </section>
</resources>
```

[Source file: EX-motivation-its-1.xml]

Example 2: Document with partially translatable content

Even when metadata are available to identify non-translatable text, the conditions may be quite complex and not directly indicated with a simple flag. Here, for instance, only the text in the nodes matching the expression `//component[@type!='image']/data [@type='text']` is translatable.

```
<dialogue xml:lang="en-gb">
 <rsrc id="123">
  <component id="456" type="image">
   <data type="text">images/cancel.gif</data>
   <data type="coordinates">12,20,50,14</data>
  </component>
  <component id="789" type="caption">
   <data type="text">Cancel</data>
   <data type="coordinates">12,34,50,14</data>
  </component>
  <component id="792" type="string">
   <data type="text">Number of files: </data>
  </component>
 </rsrc>
</dialogue>
```

[Source file: EX-motivation-its-2.xml]

1.2 Users and Usages of ITS

1.2.1 Potential Users of ITS

The ITS specification aims to provide different types of users with information about what markup should be supported to enable worldwide use and effective internationalization and localization of content. The following paragraphs sketch these different types of users, and their usage of ITS.

- Schema developers who start a schema from ground up

  This type of user will find proposals for attribute and element names to be included in their new schema (also called "host vocabulary"). Using the attribute and element names proposed in the ITS specification may be helpful because it leads to easier recognition of the concepts represented by both schema users and processors. It is perfectly possible, however, for a schema developer to develop his own set of attribute and element names. The specification sets out, first and foremost, to ensure that the required markup is available, and that the behavior of that markup meets established needs.
- Schema developers who work with an existing schema

  This type of user will be working with schemas such as DocBook, DITA, or perhaps a proprietary schema. The ITS Working Group has sought input from experts developing widely used formats such as the ones mentioned.

  > **Note:**
  >
  > The question "How to use ITS with existing popular markup schemes?" is covered in more details (including examples) in a separate document: [XML i18n BP].

  Developers working on existing schemas should check whether their schemas support the markup proposed in this specification, and, where appropriate, add the markup proposed here 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 associating ITS markup with markup in the host vocabulary which serves a similar purpose (see Section 5.5: Associating ITS Data Categories with Existing Markup). The developer should, however, check that the behavior associated with the markup in their own schema is fully compatible with the expectations described in this specification.
- Vendors of content-related tools

  This type of user includes companies which provide 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 hoped that the ITS specification 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.
- Content producers

  This type of user comprises authors, translators and other types of content author. The markup proposed in this specification may be used by them to mark up specific bits of content. Aside: The burden of inserting markup can be removed from content producers by relating the ITS information to relevant bits of content in a global manner (see global, rule-based approach). This global work, however, may fall to information architects, rather than the content producers themselves.

In order to support all of these users, the information about what markup should be supported to enable worldwide use and effective localization of content is provided in this specification in two ways:

- abstractly in the data category descriptions: Section 6: Description of Data Categories
- concretely in the ITS schemas: Appendix D: Schemas for ITS

1.2.2 Ways to Use ITS

The ITS specification proposes several mechanisms for supporting worldwide use and effective internationalization and localization of content. We will sketch them below by looking at them from the perspectives of certain user types. For the purpose of illustration, we will demonstrate how ITS can indicate that certain parts of content should or should not be translated.

- A content author uses an attribute on a particular element to say that the text in the element should not be translated.

Example 3: Use of ITS by content author

The `its:translate="no"` attributes indicate that the `path` and the `cmd` elements should not be translated.

```
<help
  xmlns:its="http://www.w3.org/2005/11/its"
  its:version="1.0">
 <head>
  <title>Building the Zebulon Toolkit</title>
 </head>
 <body>
  <p>To re-compile all the modules of the Zebulon toolkit you need to go in the
    <path
     its:translate="no">\Zebulon\Current Source\binary</path> directory.
    Then from there, run batch file <cmd
     its:translate="no">Build.bat</cmd>.</p>
 </body>
</help>
```

[Source file: EX-ways-to-use-its-1.xml]

- A content author or information architect uses markup at the top of the document to identify a particular type of element or context in which the content should not be translated.

Example 4: Use of ITS by information architect

The **translateRule** element is used in the header of the document to indicate that none of the `path` or `cmd` elements should be translated.

```
<help
  xmlns:its="http://www.w3.org/2005/11/its"
  its:version="1.0">
 <head>
  <title>Building the Zebulon Toolkit</title>
  <its:rules version="1.0">
   <its:translateRule selector="//path | //cmd" translate="no"/>
  </its:rules>
 </head>
 <body>
  <p>To re-compile all the modules of the Zebulon toolkit you need to go in the
    <path>\Zebulon\Current Source\binary</path> directory.
    Then from there, run batch file <cmd>Build.bat</cmd>.</p>
 </body>
</help>
```

[Source file: EX-ways-to-use-its-2.xml]

- A processor may insert markup at the top of the document which links to ITS information outside of the document.

Example 5: Use of ITS by processor

A **rules** element is inserted in the header of the document. It has a XLink **href** attribute used to link to an ITS external rule document.

```
<help
  xmlns:its="http://www.w3.org/2005/11/its"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  its:version="1.0">
 <head>
  <title>Building the Zebulon Toolkit</title>
  <its:rules version="1.0" xlink:href="EX-ways-to-use-its-4.xml" xlink:type="simple"/>
 </head>
 <body>
  <p>To re-compile all the modules of the Zebulon toolkit you need to go in the
    <path>\Zebulon\Current Source\binary</path> directory.
    Then from there, run batch file <cmd>Build.bat</cmd>.</p>
 </body>
</help>
```

[Source file: EX-ways-to-use-its-3.xml]

Example 6: ITS rule file shared by different documents

The **rules** element contains several ITS rules that are common to different documents. One of them is a **translateRule** element that indicates that no `path` or `cmd` element should be translated.

```
<its:rules
  xmlns:its="http://www.w3.org/2005/11/its"  version="1.0">
 <its:translateRule selector="//path | //cmd" translate="no"/>
</its:rules>
```

[Source file: EX-ways-to-use-its-4.xml]

- A schema developer integrates ITS markup declarations in his schema to allow users to indicate that specific parts of the content should not be translated.

Example 7: An XSD schema with ITS declaration

The declarations for the **translate** attribute is added to a group of common attributes `commonAtts`. This allows to use the **translate** attribute within the documents like in Example 3.

```
<xs:schema
  xmlns:its="http://www.w3.org/2005/11/its"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"  elementFormDefault="qualified">
 <xs:import namespace="http://www.w3.org/2005/11/its" schemaLocation="its.xsd"/>
 <xs:attributeGroup name="commonAtts">
  <xs:attributeGroup ref="its:att.local.with-ns.attribute.translate"/>
  <xs:attribute name="id" type="xs:ID" use="optional"/>
 </xs:attributeGroup>
 <xs:element name="help">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="head">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="title" type="xs:string"/>
      </xs:sequence>
      <xs:attributeGroup ref="commonAtts"/>
     </xs:complexType>
    </xs:element>
    <xs:element name="body">
     <xs:complexType>
      <xs:choice minOccurs="1" maxOccurs="unbounded">
       <xs:element name="p">
        <xs:complexType mixed="true">
         <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:element ref="path"/>
          <xs:element ref="cmd"/>
         </xs:choice>
         <xs:attributeGroup ref="commonAtts"/>
        </xs:complexType>
       </xs:element>
      </xs:choice>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
   <xs:attributeGroup ref="its:att.version.attribute.version"/>
  </xs:complexType>
 </xs:element>
 <xs:element name="path">
  <xs:complexType mixed="true">
   <xs:attributeGroup ref="commonAtts"/>
  </xs:complexType>
 </xs:element>
 <xs:element name="cmd">
  <xs:complexType mixed="true">
   <xs:attributeGroup ref="commonAtts"/>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

[Source file: EX-ways-to-use-its-5.xsd]

The first two approaches above can be likened to the use of CSS in [XHTML 1.0]. 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.

1.3 Out of Scope

This standard does not cover all mechanisms and data formats (sometimes called **Localization Properties**), which might be needed for configuring localization workflows or tools to process a specific format. However, these mechanisms and data formats may be implemented using the framework described in this standard.

Note:

"XML localization properties" is a generic term to name the mechanisms and data formats that allow localization tools to be configured in order to process a specific XML format. Examples of "XML localization properties" are the Trados "DTD Settings" file, and the SDLX "Analysis" file.

1.4 Important Design Principles

Abstraction via *data categories*: ITS defines data categories as an abstract notion for information for internationalization and localization of XML schemas and documents. This abstraction is helpful in realizing independence from a particular implementation using for example an element or attribute. See Section 3.3: Data category for a definition of the term data categories, Section 6: Description of Data Categories for the definition of the various ITS data categories, and subsections in Section 6: Description of Data Categories for the data category implementations.

Powerful *selection mechanism:* For ITS markup which appears in an XML instance, it has to be clearly defined to which XML nodes the ITS-related information pertains. Thus, ITS defines selection mechanisms to specify to what parts of an XML document an ITS data category and its values should be applied. Selection relies on the information which is given in the XML Information Set [XML Infoset]. ITS applications may implement inclusion mechanisms such as XInclude or DITA's [DITA 1.0] conref.

Content authors need, for example, a simple way to work with the Translate data category in order to express whether the content of an element or attribute should be translated or not. Localization coordinators, on the other hand, need an efficient way of managing translations of large document sets based on the same schema. This could by realized by a specification of defaults for the Translate data category and exceptions from the defaults (e.g. all `p` elements should be translated, but not `p` elements inside of an `index` element).

To meet these requirements this specification introduces mechanisms that add ITS information to XML documents, see Section 5: Processing of ITS information. These mechanisms also provide a means for specifying ITS information for attributes (a task for which no standard means yet exists).

The ITS selection mechanisms allows you to provide information about content locally (specified at the XML node to which it pertains) or globally (specified in another part of the document). Global selection mechanisms can be in the same document, or in a separate file.

No dedicated extensibility: It may be useful or necessary to extend the set of information available for internationalization or localization purposes beyond what is provided by ITS. This specification does not define a dedicated extension mechanism, since ordinary XML mechanisms (e.g. XML Namespaces [XML Names]) may be used.

Ease of integration:

- ITS follows the example from section 4 of [XLink 1.0], by providing mostly global attributes for the implementation of ITS data categories. Avoiding elements for ITS purposes as much as possible ensures ease of integration into existing markup schemes, see section 3.14 in [ITS REQ]. Only for some requirements do additional child elements have to be used, see for example Section 6.6: Ruby.
- ITS has no dependency on technologies which are still under development
- ITS fits with existing work in the W3C architecture (e.g. use of [XPath 1.0] for the selection mechanism)

1.5 Development of this Specification

This specification has been developed using the ODD (*One Document Does it all*) language of the Text Encoding Initiative ([TEI]). This is a literate programming language for writing XML schemas, with three characteristics:

1. The element and attribute set is specified using an XML vocabulary which includes support for macros (like DTD entities, or schema patterns), a hierarchical class system for attributes and elements, and creation of modules.
2. The content models for elements and attributes are written using embedded RELAX NG XML notation.
3. Documentation for elements, attributes, value lists etc. is written inline, along with examples and other supporting material.

XSLT transformations are provided by the TEI to create documentation into HTML, XSL FO or LaTeX forms, and to generate RELAX NG documents and DTD. From the RELAX NG documents, James Clark's trang can be used to create XML Schema documents.

2 Basic Concepts

This section is informative.

2.1 Selection

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

The mechanisms defined for ITS selection resemble those defined in [CSS 2.1]. The local approach can be compared to the `style` attribute in HTML/XHTML, and the approach with global rules is similar to the `style` element in HTML/XHTML. In contrast to CSS, ITS uses XPath for identifying nodes. Thus,

- 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)

ITS markup can be used with XML documents (e.g. a DocBook article), or schemas (e.g. an XML Schema document for a proprietary document format). Since each usage defines some specific requirements, ITS markup may take different shapes.

The following two examples sketch the distinction between the local and global approaches.

2.1.1 Local Approach

The document in Example 8 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.

Example 8: ITS markup on elements in an XML document (local approach)

```
<dbk:article
  xmlns:its="http://www.w3.org/2005/11/its"
  xmlns:dbk="http://docbook.org/ns/docbook"
  its:version="1.0" version="5.0" xml:lang="en">
 <dbk:info>
  <dbk:title>An example article</dbk:title>
  <dbk:author
    its:translate="no">
   <dbk:personname>
    <dbk:firstname>John</dbk:firstname>
    <dbk:surname>Doe</dbk:surname>
   </dbk:personname>
   <dbk:affiliation>
    <dbk:address>
     <dbk:email>foo@example.com</dbk:email>
    </dbk:address>
   </dbk:affiliation>
  </dbk:author>
 </dbk:info>
 <dbk:para>This is a short article.</dbk:para>
</dbk:article>
```

[Source file: EX-basic-concepts-1.xml]

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.

2.1.2 Global Approach

The document in Example 9 shows a different approach to identifying non-translatable content, similar to that used with a `style` element in [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). It 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 taken from namespace declarations [XML Names] at the current rules element.

> **Note:**
>
> Caveat Related to XSLT-based Processing of ITS Selector Attributes
>
> The values of ITS **selector** attributes are XPath absolute location paths. Accordingly, the following is a legitimate value:
>
> myElement/descendant-or-self::*/@*
>
> Unfortunately, values like this cause trouble when they are used in XSLT-based processing of ITS where the values of the ITS **selector** attributes are used as values of `match` attributes of XSLT templates. The reason for this is the following: `match` attributes may only contain a restriction/subset of XPath expressions, so-called patterns.
>
> Basically the following restrictions hold for patterns:
>
> - only axes "child" or "attribute" allowed
> - "//" or "/" possible
> - id() or key() function possible
> - predicates possible
>
> Using only XSLT patterns in ITS **selector** attributes helps to avoid this issue. In many cases, this is possible by using patterns with predicates. The value above may for example be rewritten as follows:
>
> *[self::myElement]/@* | myElement//*/@*

Example 9: ITS global markup in an XML document (rule-based approach)

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

[Source file: EX-basic-concepts-2.xml]

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 an **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 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 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)
- ITS data categories can designate attribute values as well as elements.
- It is possible to associate ITS markup with existing markup (for example the `term` element in DITA)

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"

The ITS **selector** attribute allows:

- ITS data category attributes to appear in global rules (even outside of an XML document or schema)
- ITS data categories attributes to pertain to sets of XML nodes (for example all `p` elements in an XML document)
- ITS markup to pertain to attributes
- ITS markup to associate with existing markup (for example the `term` element in DITA)

2.2 Overriding and Inheritance

The power of the ITS selection mechanisms comes at a price: rules related to overriding/precedence, and inheritance, have to be established.

The document in Example 10 shows how inheritance and overriding work for the Translate data category. By default elements are translatable. Here, the **translateRule** element declared in the header overrides the default for the `head` element inside `text` and for all its children. Because the `title` element is actually translatable, the global rule needs to be overridden by a local `its:translate="yes"`. Note that the global rule is processed first, regardless of its position inside the document. In the main body of the document, the default applies, and here it is `its:translate="no"` that is used to set "faux pas" as non-translatable.

Example 10: Overriding and Inheritance

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

[Source file: EX-basic-concepts-3.xml]

2.3 Adding Information or Pointing to Existing Information

For some data categories, special attributes add or point to information about the selected nodes. For example, the Localization Note data category can add information to selected nodes (using a **locNote** element), or point at existing information elsewhere in the document (using a **locNotePointer** attribute).

The functionality of adding information to the selected nodes is available for each data category except Language Information. Pointing to existing information is not possible for data categories that express *a closed set of values*; that is: Translate, Directionality and Elements Within Text.

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.

3 Notation and Terminology

This section is normative.

3.1 Notation

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119].

The namespace URI that MUST be used by implementations of this specification is:

```
http://www.w3.org/2005/11/its
```

The namespace prefix used in this specification for this URI is "its". It is recommended that implementations of this specification use this prefix.

In addition, the following namespaces are used in this document:

- `http://www.w3.org/2001/XMLSchema` for the XML Schema namespace, here used with the prefix "xs"
- `http://relaxng.org/ns/structure/1.0` for the RELAX NG namespace, here used with the prefix "rng"

- `http://www.w3.org/1999/xlink` for the XLink namespace, here used with the prefix "xlink"

3.2 Schema Language

[Definition: **Schema language** refers in this specification to an XML-related modeling or validation language such as XML DTD, XML Schema or RELAX NG.]

Note:

This specification provides schemas in the format of XML DTD, XML Schema or RELAX NG. However, these schemas are only non-normative; conformance for ITS markup declarations defines only mandatory positions of ITS declarations in schemas. This makes it possible to use ITS with any schema language that allows for using these positions.

3.3 Data category

[Definition: ITS defines **data category** as an abstract concept for a particular type of information 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).

For each data category, ITS distinguishes between the following:

- the prose description, see Section 6: Description of Data Categories
- schema language independent formalization, see the "markup declarations" subsections in Section 6: Description of Data Categories
- schema language specific implementations, see Appendix D: Schemas for ITS

Example 11: A data category and its implementation

The Translate data category conveys information as to whether a piece of content should be translated or not.

The simplest formalization of this prose description on a schema language independent level is a **translate** attribute with two possible values: "yes" and "no". An implementation on a schema language specific level would be the declaration of the **translate** attribute in, for example, an XML DTD, an XML Schema document or an RELAX NG document. A different implementation would be a **translateRule** element that allows for specifying global rules about the Translate data category.

3.4 Selection

[Definition: **selection** encompasses mechanisms to specify to what parts of an XML document an ITS data category and its values should be applied to.] Selection is discussed in detail in Section 5: Processing of ITS information. Selection can be applied globally, see Section 5.2.1: Global, Rule-based Selection, and locally, see Section 5.2.2: Local Selection in an XML Document. As for global selection, ITS information can be added to the selected nodes, or it can point to existing information which is related to selected nodes.

Selection relies on the information that is given in the XML Information Set [XML Infoset]. ITS applications MAY implement inclusion mechanisms such as XInclude or DITA's [DITA 1.0] conref.

Note:

The selection of the ITS data categories applies to textual values contained within element or attribute nodes. In some cases these nodes form pointers to other resources; a well-known example is the `src` attribute on the `img` element in HTML. The ITS Translate data category applies to the text of the pointer itself, not the object to which it points. Thus in the following example, the translation information specified via the **translateRule** element applies to the filename "instructions.jpg", and is not an instruction to open the graphic and change the words therein.

Example 12: Selecting the text of a pointer to an external object

```
<text
  xmlns:its="http://www.w3.org/2005/11/its" >
 <its:rules version="1.0">
  <its:translateRule translate="yes" selector="//p/img/@src"/>
 </its:rules> ...
 <p>As you can see in
  <img src="instructions.jpg"/>,
  the truth is not always out there.</p>
</text>
```

[Source file: EX-notation-terminology-1.xml]

3.5 Usage of Internationalized Resource Identifiers in ITS

The attributes **href**, **locNoteRef** and **termInfoRef** which contain resource identifiers MUST allow the usage of Internationalized Resource Identifiers (IRIs, [RFC 3987] or its successor) to ease the adoption of ITS in international application scenarios.

Note:

The ITS schemas in Appendix D: Schemas for ITS are not normative. Hence this specification defines no validation requirements for IRI values in ITS markup. For processing of these values, relying on IRIs imposes no specific requirements. The reason is that the processing happens on the info set level [XML Infoset], where no difference between IRIs and URIs exists.

4 Conformance

This section is normative.

The usage of the term *conformance clause* in this section is in compliance with [QAFRAMEWORK].

This specification defines two types of conformance: conformance of 1) ITS markup declarations , and conformance of 2) processing expectations for ITS Markup. These conformance types complement each other. An implementation of this specification MAY use them separately or together.

4.1 Conformance Type 1: ITS Markup Declarations

Description: ITS markup declarations encompass all declarations that are part of the Internationalization Tag Set. They do not concern the *usage* of the markup in XML documents. Such markup is subject to the conformance clauses in Section 4.2: Conformance Type 2: The Processing Expectations for ITS Markup.

Definitions related to this conformance type: ITS markup declarations are defined in various subsections in Section 5: Processing of ITS information and Section 6: Description of Data Categories (e.g. Section 6.3.3: Markup Declarations for Localization Note) in a schema language independent manner, relying on the ODD language. Their occurrence in other sections of this document is typographically marked via bold face and color.

Who uses this conformance type: Schema designers integrating ITS markup declarations into a schema. All conformance clauses for this conformance type concern the position of ITS markup declarations in that schema, and their status as mandatory or optional.

Conformance clauses:

- *1-1:* At least one of the following MUST be in the schema:
  - **rules** element
  - one of the local ITS attributes
  - **span** element
  - **ruby** element
- *1-2:* If the **rules** element is used, it MUST be part of the content model of at least one element declared in the schema. It SHOULD be in a content model for meta information, if this is available in that schema (e.g. the `head` element in [XHTML 1.0]).
- *1-3:* If the **ruby** element is used, it SHOULD be declared as an inline element.
- *1-4:* If the **span** element is used, it SHOULD be declared as an inline element.

Full implementations of this conformance type will implement all markup declarations for ITS. Statements related to this conformance type MUST list all markup declarations they implement.

Examples: Examples of the usage of ITS markup declarations in various existing schemas are given in a separate document [XML i18n BP].

> **Note:**
>
> Since the ITS markup declarations are schema language independent, each schema language can use its own, possibly multiple, mechanisms to implement the conformance clauses for ITS markup declarations. For example, an XML DTD can use parameter entities to encapsulate the ITS local attributes, or declare them directly for each element. The appropriate steps to integrate ITS into a schema depend on the design of this schema (e.g. whether it already has a customization layer that uses parameter entities). The ITS schemas in the format of XML DTD, XML Schema and RELAX NG in Appendix D: Schemas for ITS are only informative examples.

4.2 Conformance Type 2: The Processing Expectations for ITS Markup

Description: Processors need to compute the ITS information that pertains to a node in an XML document. The ITS processing expectations define how the computation has to be carried out. Correct computation involves support for selection mechanism, defaults / inheritance / overriding characteristics, and precedence. The markup MAY be valid against a schema which conforms to the clauses in Section 4.1: Conformance Type 1: ITS Markup Declarations.

Definitions related to this conformance type: The processing expectations for ITS markup make use of selection mechanisms defined in Section 5: Processing of ITS information. The individual data categories defined in Section 6: Description of Data Categories have defaults / inheritance / overriding characteristics, and allow for using ITS markup in various positions (global and local).

Who uses this conformance type: Applications that need to process for internationalization or localization the nodes captured by a data category. Examples of this type of application are: ITS markup-aware editors, or translation tools that make use of ITS markup to filter translatable text as an input to the localization process.

> **Note:**
>
> Application-specific processing (that is processing that goes beyond the computation of ITS information for a node) such as automated filtering of translatable content based on the Translate data category is not covered by the conformance clauses below.
>
> **Note:**
>
> The ITS Working group provides a test suite to help implementers to write applications that support the ITS specifications. The test suite provides pairs of input and output files.

Conformance clauses:

- *2-1:* A processor MUST implement at least one data category. For each implemented data category, the following MUST be taken into account:
  - *2-1-1:* processing of at least one selection mechanism (global or local).

  - *2-1-2:* the default selections for the data category.
  - *2-1-3:* the precedence definitions for selections defined in Section 5.4: Precedence between Selections, for the type of selections it processes.
- *2-2:* If an application claims to process ITS markup for the global selection mechanism, it MUST process an XLink **href** attribute found on a **rules** elements.

Statements related to this conformance type MUST list all data categories they implement, and for each data category which type of selection they support.

5 Processing of ITS information

This section is normative.

5.1 Indicating the Version of ITS

The version of the ITS schema defined in this specification is "1.0". The version is indicated by the ITS **version** attribute. This attribute is mandatory for the **rules** element, where it MUST be in no namespace. If there is no **rules** element in an XML document, a prefixed ITS **version** attribute (e.g. `its:version`) MUST be provided at the root element of the document. If there is both a **version** attribute at the root element and a **rules** element in a document, they MUST NOT specify different versions.

Each XML document can have a different version. That is: if external rules are linked via an XLink **href** attribute on the **rules** element, they can specify a different version than the **rules** element.

5.2 Locations of Data Categories

ITS data categories can appear in two places:

- Global rules: the selection is realized within a **rules** element. It contains rule elements for each data category. Each rule element has a **selector** attribute and possibly other attributes. The **selector** attribute contains an AbsoluteLocationPath as described in XPath 1.0 or its successor.
- Locally in a document: the selection is realized using ITS local attributes, which are attached to an element node, or the **span** or **ruby** element. There is no additional **selector** attribute. The default selection for each data category defines whether the selection covers attributes and child elements. See Section 6.1: Position, Defaults, Inheritance and Overriding of Data Categories.

The two locations are described in detail below.

5.2.1 Global, Rule-based Selection

Global, rule-based selection is implemented using the **rules** element. It contains zero or more rule elements. Each rule element has a mandatory **selector** attribute. This attribute and all other possible attributes on rule elements are in the empty namespace and used without a prefix.

If there is more than one **rules** element in an XML document, the rules from each section are to be processed at the same precedence level. The **rules** sections are to be read in document order, and the ITS rules with them processed sequentially. The versions of these **rules** elements MUST NOT be different.

Depending on the data category and its usage, there are additional attributes for adding information to the selected nodes, or for pointing to existing information in the document. For example, the Localization Note data category can be used for adding notes to selected nodes, or for pointing to existing notes in the document. For the former purpose, a **locNote** element can be used. For the latter purpose, a **locNotePointer** attribute can be used.

Each data category allows you to add information to the selected nodes except for language information. Pointing to existing information is not possible for data categories that express *a closed set of values*, that is: Translate, Directionality and Elements Within Text.

The functionalities of adding information and pointing to existing information are *mutually exclusive*. That is: markup for pointing and adding MUST NOT appear in the same rule element.

Another difference between adding and pointing is the usage of XPath:

- The value of the **selector** attribute MUST be an XPath expression which starts with "`/`". That is, it must be an AbsoluteLocationPath as described in XPath 1.0 or its successor. This ensures that the selection is not relative to a specific location. The resulting nodes MUST be either element or attribute nodes.
- Attributes that point to existing information in the document, i.e. attributes whose name ends in `...Pointer`, MUST use a RelativeLocationPath as described in XPath 1.0 or its successor. The XPath expression is evaluated relative to the nodes selected by the selector attribute. The following attributes point to existing information: **locNotePointer**, **locNoteRefPointer**, **termInfoPointer**, **termInfoRefPointer**, **rubyPointer**, **rtPointer**, **rpPointer**, **rbcPointer**, **rtcPointer**, **rbspanPointer**, **langPointer**.

If namespaces [XML Names] are used in XPath expressions in the **selector** attribute or the pointing attributes, the following rules MUST be applied while processing XPath:

1. For each prefix, there MUST be an **xmlns** attribute at the same rule element which allows to resolve the namespace URI of the prefix.
2. Element and attribute names without a prefix are interpreted as having no namespace.
3. To avoid a conflict with rule 2., default namespaces MUST NOT be used in the XPath expressions.

Example 13: XPath expressions with namespaces

The `term` element from the TEI is in a namespace `http://www.tei-c.org/ns/1.0`.

```
<its:rules
  xmlns:its="http://www.w3.org/2005/11/its"
  xmlns:tei="http://www.tei-c.org/ns/1.0" version="1.0">
 <its:termRule selector="//tei:term" term="yes"/>
</its:rules>
```

[Source file: EX-selection-global-1.xml]

Example 14: XPath expressions without namespaces

The qterm element from DocBook is in no namespace.

```
<its:rules
  xmlns:its="http://www.w3.org/2005/11/its"  version="1.0">
 <its:termRule selector="//qterm" term="yes"/>
</its:rules>
```

[Source file: EX-selection-global-2.xml]

Global rules can appear in the XML document they will be applied to, or in a separate XML document. The precedence of their processing depends on these variations. See also Section 5.4: Precedence between Selections.

Markup for global, rule-based selection is defined as follows.

rules

[1]	rules	::=	element its:rules { rules.content, rules.attributes }
[2]	rules.content	::=	(translateRule | locNoteRule | termRule | dirRule | rubyRule | langRule | withinTextRule)*
[3]	rules.attributes	::=	attribute version { xsd:float }, attribute xlink:href { xsd:anyURI }?, attribute xlink:type { "simple" }?

att.selector

[4]	att.selector.attributes	::=	att.selector.attribute.selector
[5]	att.selector.attribute.selector	::=	attribute selector { string }

att.version

[6]	att.version.attributes	::=	att.version.attribute.version
[7]	att.version.attribute.version	::=	attribute its:version { xsd:float }

5.2.2 Local Selection in an XML Document

Local selection in XML documents is realized with local ITS attributes, the **ruby** element, or the **span** element. **span** serves just as a carrier for the local ITS attributes and a container for **ruby**.

The content model of **span** permits arbitrary nesting of ruby markup, since the **rb** and **rt** elements themselves can contain **span**. An application of ruby MUST not use such arbitrary nesting.

The data category determines what is being selected. The necessary data category specific defaults are described in Section 6.1: Position, Defaults, Inheritance and Overriding of Data Categories.

Example 15: Defaults for various data categories

By default the content of all elements in a document is translatable. The attribute its:translate="no" in the head element means that the content of this element, including child elements, should not be translated. The attribute its:translate="yes" in the title element means that the content of this element, should be translated (overriding the its:translate="no" in head). Attribute values of the selected elements or their children are not affected by local **translate** attributes. By default they are not translatable.

The default directionality of a document is left-to-right. The its:dir="rtl" in the quote element means that the directionality of the content of this element, including child elements and attributes, is right-to-left. Note that xml:lang indicates only the language, not the directionality.

```
<text
  xmlns:its="http://www.w3.org/2005/11/its"
  its:version="1.0" xml:lang="en">
 <head
   its:translate="no">
  <author>Sven Corneliusson</author>
  <date>2006-09-26T17:34:04Z</date>
  <title
    its:translate="yes" role="header">Bidirectional Text</title>
 </head>
 <body>
  <par>In Arabic, the title <quote xml:lang="ar"
     its:dir="rtl">نشاط التدويل، W3C</quote>
     means <quote>Internationalization Activity, W3C</quote>.</par>
 </body>
</text>
```

[Source file: EX-selection-local-1.xml]

Markup for local selection is defined as follows. The attribute group att.local.no-ns.attributes contains ITS attributes in no namespace and is used with the ITS elements **span**, **locNote**, **ruby**, **rb**, **rt**, **rbc**, **rtc** and **rp**. The attribute group att.local.with-ns.attributes contains namespace qualified ITS attributes and is used with elements from different namespaces.

att.local.no-ns

[8]	att.local.no-ns.attributes	::=	att.local.no-ns.attribute.translate, att.local.no-ns.attribute.locNote, att.local.no-ns.attribute.locNoteType, att.local.no-ns.attribute.locNoteRef, att.local.no-ns.attribute.termInfoRef, att.local.no-ns.attribute.term, att.local.no-ns.attribute.dir
[9]	att.local.no-ns.attribute.translate	::=	attribute translate { "yes" | "no" }?

[10]	`att.local.no-ns.attribute.locNote`	::= `attribute locNote { string }?`
[11]	`att.local.no-ns.attribute.locNoteType`	::= `attribute locNoteType { "alert" | "description" }?`
[12]	`att.local.no-ns.attribute.locNoteRef`	::= `attribute locNoteRef { xsd:anyURI }?`
[13]	`att.local.no-ns.attribute.termInfoRef`	::= `attribute termInfoRef { xsd:anyURI }?`
[14]	`att.local.no-ns.attribute.term`	::= `attribute term { "yes" | "no" }?`
[15]	`att.local.no-ns.attribute.dir`	::= `attribute dir { "ltr" | "rtl" | "lro" | "rlo" }?`

att.local.with-ns

[16]	`att.local.with-ns.attributes`	::= `att.local.with-ns.attribute.translate`, `att.local.with-ns.attribute.locNote`, `att.local.with-ns.attribute.locNoteType`, `att.local.with-ns.attribute.locNoteRef`, `att.local.with-ns.attribute.termInfoRef`, `att.local.with-ns.attribute.term`, `att.local.with-ns.attribute.dir`
[17]	`att.local.with-ns.attribute.translate`	::= `attribute its:translate { "yes" | "no" }?`
[18]	`att.local.with-ns.attribute.locNote`	::= `attribute its:locNote { string }?`
[19]	`att.local.with-ns.attribute.locNoteType`	::= `attribute its:locNoteType { "alert" | "description" }?`
[20]	`att.local.with-ns.attribute.locNoteRef`	::= `attribute its:locNoteRef { xsd:anyURI }?`
[21]	`att.local.with-ns.attribute.termInfoRef`	::= `attribute its:termInfoRef { xsd:anyURI }?`
[22]	`att.local.with-ns.attribute.term`	::= `attribute its:term { "yes" | "no" }?`
[23]	`att.local.with-ns.attribute.dir`	::= `attribute its:dir { "ltr" | "rtl" | "lro" | "rlo" }?`

span

[24]	`span`	::= `element its:span { span.content, span.attributes }`
[25]	`span.content`	::= `( text | ruby | span )*`
[26]	`span.attributes`	::= `att.local.no-ns.attributes`

5.3 Link to External Rules

One way to associate a document with a set of external ITS rules is to use the optional XLink [XLink 1.0] **href** attribute in the **rules** element, accompanied by the XLink **type** attribute with the value "simple". The referenced document must be a valid XML document containing at most one **rules** element. That **rules** element can be the root element or anywhere within the document tree (for example, the document could be an XML Schema).

The rules contained in the referenced document MUST be processed as if they were at the top of the **rules** element with the XLink **href** attribute.

Example 16: External file EX-link-external-rules-1.xml with global rules:

The example demonstrates how metadata can be added to ITS 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 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>
```

[Source file: EX-link-external-rules-1.xml]

Example 17: Document with a link to EX-link-external-rules-1.xml

```
<myDoc
  xmlns:its="http://www.w3.org/2005/11/its"
  xmlns:xlink="http://www.w3.org/1999/xlink" >
 <header>
  <its:rules version="1.0" xlink:href="EX-link-external-rules-1.xml" xlink:type="simple">
   <its:translateRule selector="//term" translate="yes"/>
  </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>
```

[Source file: EX-link-external-rules-2.xml]

The result of processing the two documents above is the same as processing the following document.

Example 18: Document with identical rules as in the case of included rules

```
<myDoc
  xmlns:its="http://www.w3.org/2005/11/its" >
 <header>
  <its:rules 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:translateRule selector="//term" translate="yes"/>
  </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>
```

[Source file: EX-link-external-rules-3.xml]

Applications processing global ITS markup MUST recognize the XLink **href** attribute in the **rules** element; they MUST load the corresponding referenced document and process its rules element before processing the content of the **rules** element where the original XLink **href** attribute is.

External rules may also have links to other external rules. The linking mechanism is recursive, the deepest rules being overridden by the top-most rules, if any.

5.4 Precedence between Selections

The following precedence order is defined for selections of ITS information in various positions (the first item in the list has the highest precedence):

1. Implicit local selection in documents (ITS local attributes on a specific element)
2. Global selections in documents (using a **rules** element)

   Inside each **rules** element the precedence order is:

   1. Any rules inside the rules element
   2. Any rules linked via the XLink **href** attribute

   Note:

   If identical selections are defined in different rules elements within one document, the selection defined by the last takes precedence.

   Note:

   ITS doesn't define precedence related to rules defined or linked based on non-ITS mechanisms (such as processing instructions for linking rules).
3. Selections via defaults for data categories, see Section 6.1: Position, Defaults, Inheritance and Overriding of Data Categories

In case of conflicts between global selections via multiple rule elements, the last selector has higher precedence.

Note:

The precedence order fulfills the same purpose as the built-in template rules of [XSLT 1.0].

Example 19: Conflicts between selections of ITS information which are resolved using the precedence order

The two elements `title` and `author` of this document should be treated as separate content when inside a `prolog` element, but as part of the content of their parent element otherwise. In order to make this distinction two **withinTextRule** elements are used:

The first rule specifies that `title` and `author` in general should be treated as an element within text. This overrides the default.

The second rule indicates that when `title` or `author` are found in a `prolog` element their content should be treated separately. This is normally the default, but the rule is needed to override the first rule.

```
<text
  xmlns:its="http://www.w3.org/2005/11/its" >
 <prolog>
  <its:rules version="1.0">
   <its:withinTextRule withinText="yes" selector="//title|//author"/>
   <its:withinTextRule withinText="no" selector="//prolog/title|//prolog/author"/>
  </its:rules>
  <title>Designing User Interfaces</title>
  <author>Janice Prakash</author>
  <keywords>user interface, ui, software interface</keywords>
 </prolog>
 <body>
  <p>The book <title>Of Mice and Screens</title> by <author>Aldus
     Brandywine</author> is one of the best introductions to the vast
    topic of designing user interfaces.</p>
 </body>
</text>
```

[Source file: EX-selection-precedence-1.xml]

5.5 Associating ITS Data Categories with Existing Markup

Some markup schemes provide markup which can be used to express ITS data categories. ITS data categories can be associated with such existing markup, using the global selection mechanism described in Section 5.2.1: Global, Rule-based Selection.

Associating existing markup with ITS data categories can be done only if the processing expectations of the host markup are the same as, or greater than, those of ITS.

Example 20: Association of the ITS data categories Translate and Terminology with DITA 1.0 markup

```
<topic
  xmlns:its="http://www.w3.org/2005/11/its"  id="myTopic">
 <title>The ITS Topic</title>
 <prolog>
  <its:rules version="1.0">
   <its:translateRule selector="//*[@translate='no']" translate="no"/>
   <its:translateRule selector="//*[@translate='yes']" translate="yes"/>
   <its:termRule selector="//term | //dt" term="yes"/>
  </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>For the implementation of ITS, apply the rules in the order:</p>
  <ul>
   <li>Defaults</li>
   <li>Rules in external files</li>
   <li>Rules in the document</li>
   <li>Local attributes</li>
  </ul>
  <p>
   <ph translate="no" xml:lang="fr">Et voilà !</ph>.</p>
 </body>
</topic>
```

[Source file: EX-associating-its-with-existing-markup-1.xml]

Global rules can be associated with a given XML document using different means:

- By using an **rules** element in the document itself:
  - with the rules directly inside the document, as shown in Example 20
  - with a link to an external rules file using the XLink **href** attribute, as shown in Example 16
- By associating the rules and the document through a tool-specific mechanism. For example, for a command-line tool: providing the paths of both the XML document to process and its corresponding external rules file.

6 Description of Data Categories

This section is normative.

6.1 Position, Defaults, Inheritance and Overriding of Data Categories

The following table summarizes for each data category which selection, default value, and inheritance and overriding behavior applies.

- *Default values* apply if both local or global selection are absent. The default value for the Translate data category for example mandates that elements are translatable, and attributes are not translatable if there is no **translateRule** element and no **translate** attribute available.
- *Inheritance* describes whether ITS information is applicable to child elements of nodes and attributes related to these nodes or their child notes. The inheritance for the Translate data category for example mandates that all child elements of nodes are translatable whereas all attributes related to these the nodes or their child notes are not translatable.
- *Overriding* describes whether ITS information can be overridden or not. Overriding is only applicable for data categories with inheritance. Overriding thus is not applicable for the Terminology and the Ruby data category.

Data category	Local Usage	Global, rule-based selection	Global adding of information	Global pointing to existing information	Default Values	Inheritance	Overriding
Translate	Yes	Yes	Yes	No	`translate="yes"` for elements, and `translate="no"` for attributes	Textual content of element, *including* content of child elements, but *excluding* attributes	Yes
Localization Note	Yes	Yes	Yes	Yes	None	Textual content of element, *including* content of child elements, but *excluding* attributes	Yes
Terminology	Yes	Yes	Yes	Yes	`term="no"`	None	Not applicable
Directionality	Yes	Yes	Yes	No	`dir="ltr"`	Textual content of element, *including* attributes and child elements	Yes
Ruby	Yes	Yes	Yes	Yes	None	None	Not applicable
Language	No	Yes	No	Yes	None	Textual content of element, *including* attributes and	Yes

Information						child elements	
Elements Within Text	No	Yes	Yes	No	`withinText="no"`	None	Not applicable

Example 21: Defaults, inheritance and overriding behavior of data categories

In this example, the content of all the `data` elements is translatable because the default for the Translate data category in elements is "yes". The content of `revision` and **locNote** is not translatable because the default is overridden by the local `its:translate="no"` attribute in the `prolog` element, and that value is inherited by all the children of `prolog`.

The localization note for the two first `data` elements is the text defined globally with the **locNoteRule** element. And this note is overridden for the last `data` element by the local `its:locNote` attribute.

```
<Res
  xmlns:its="http://www.w3.org/2005/11/its"
  its:version="1.0">
 <prolog
   its:translate="no">
  <revision>Sep-07-2006</revision>
  <its:rules version="1.0">
   <its:translateRule selector="//msg/notes" translate="no"/>
   <its:locNoteRule locNoteType="description" selector="//msg/data">
    <its:locNote>The variable {0} is the name of the host.</its:locNote>
   </its:locNoteRule>
  </its:rules>
 </prolog>
 <body>
  <msg id="HostNotFound">
   <data>Host {0} cannot be found.</data>
  </msg>
  <msg id="HostDisconnected">
   <data>The connection with {0} has been lost.</data>
  </msg>
  <msg id="FileNotFound">
   <data
     its:locNote="{0} is a filename">{0} not found.</data>
  </msg>
 </body>
</Res>
```

[Source file: EX-datacat-behavior-1.xml]

Note:

The data categories differ with respect to defaults. This is due to existing standards and practices. It is common practice for example that information about translation refers only to textual content of an element. Thus, the default selection for the Translate data category is the textual content.

6.2 Translate

6.2.1 Definition

The Translate data category expresses information about whether the content of an element or attribute should be translated or not. The values of this data category are "yes" (translatable) or "no" (not translatable).

6.2.2 Implementation

The Translate data category can be expressed with global rules, or locally on an individual element. The information applies to the textual content of the element, *including* child elements, but *excluding* attributes. The default is that elements are translatable and attributes are not.

GLOBAL: The **translateRule** element contains the following:

- A required **selector** attribute. It contains an XPath expression which selects the nodes to which this rule applies.
- A required **translate** attribute with the value "yes" or "no".

Example 22: The Translate data category expressed globally

The **translateRule** element specifies that the elements `code` must not be translated.

```
<its:rules
  xmlns:its="http://www.w3.org/2005/11/its"  version="1.0">
 <its:translateRule translate="no" selector="//code"/>
</its:rules>
```

[Source file: EX-translate-selector-1.xml]

LOCAL: The following local markup is available for the Translate data category:

- A **translate** attribute with the value "yes" or "no".

Note:

It is not possible to override the Translate data category settings of attributes using local markup. This limitation is consistent with the advised practice of not using translatable attributes.

Example 23: The Translate data category expressed locally

The local `its:translate="no"` specifies that the content of `panelmsg` must not be translated.

```
<messages
  xmlns:its="http://www.w3.org/2005/11/its"
  its:version="1.0">
 <msg num="123">Click Resume Button on Status Display or
    <panelmsg
    its:translate="no">CONTINUE</panelmsg> Button
    on printer panel</msg>
</messages>
```

[Source file: EX-translate-selector-2.xml]

6.2.3 Markup Declarations for Translate

translateRule

[27] `translateRule` ::= `element its:translateRule { translateRule.content, translateRule.attributes }`

[28] `translateRule.content` ::= `empty`

[29] `translateRule.attributes` ::= `att.selector.attributes, attribute translate { "yes" | "no" }`

att.translate

[30] `att.translate.attributes` ::= `att.translate.attribute.translate`

[31] `att.translate.attribute.translate` ::= `attribute its:translate { "yes" | "no" }?`

6.3 Localization Note

6.3.1 Definition

The Localization Note data category is used to communicate notes to localizers about a particular item of content.

This data category can be used for several purposes, including, but not limited to:

- Tell the translator how to translate parts of the content
- Expand on the meaning or contextual usage of a specific element, such as what a variable refers to or how a string will be used on the user interface
- Clarify ambiguity and show relationships between items sufficiently to allow correct translation (e.g. in many languages it is impossible to translate the word "*enabled*" in isolation without knowing the gender, number and case of the thing it refers to.)
- Indicate why a piece of text is emphasized (important, sarcastic, etc.)

Two types of informative notes are needed:

- An alert contains information that the translator must read before translating a piece of text. Example: an instruction to the translator to leave parts of the text in the source language.
- A description provides useful background information that the translator will refer to only if they wish. Example: a clarification of ambiguity in the source text.

Editing tools may offer an easy way to create this type of information. Translation tools can be made to recognize the difference between these two types of localization notes, and present the information to translators in different ways.

6.3.2 Implementation

The Localization Note data category can be expressed with global rules, or locally on an individual element. The information applies to the textual content of the element, *including* child elements, but *excluding* attributes.

GLOBAL: The **locNoteRule** element contains the following:

- A required **selector** attribute. It contains an XPath expression which selects the nodes to which this rule applies.
- A required **locNoteType** attribute with the value "description" or "alert".
- Exactly one of the following:
  - A **locNote** element that contains the note itself and allows for local ITS markup.
  - A **locNotePointer** attribute that contains a relative XPath expression pointing to a node that holds the localization note.
  - A **locNoteRef** attribute that contains a URI referring to the location of the localization note.
  - A **locNoteRefPointer** attribute that contains a relative XPath expression pointing to a node that holds the URI referring to the location of the localization note.

Example 24: The 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 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>
```

[Source file: EX-locNote-element-1.xml]

Example 25: The 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 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>
</Res>
```

[Source file: EX-locNotePointer-attribute-1.xml]

Example 26: The 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 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>
```

[Source file: EX-locNoteRef-attribute-1.xml]

Example 27: The 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.

```
<dataFile
  xmlns:its="http://www.w3.org/2005/11/its" >
 <prolog>
  <its:rules 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>
</dataFile>
```

[Source file: EX-locNoteRefPointer-attribute-1.xml]

LOCAL: The following local markup is available for the Localization Note data category:

- One of the following:
  - A **locNote** attribute that contains the note itself.
  - A **locNoteRef** attribute that contains a URI referring to the location of the localization note.
- An optional **locNoteType** attribute with the value "description" or "alert". If the **locNoteType** attribute is not present, the type of localization note will be assumed to be "description".

Example 28: The Localization Note data category expressed locally

```
<msgList
  xmlns:its="http://www.w3.org/2005/11/its"  xml:space="preserve"
  its:version="1.0">
 <data name="LISTFILTERS_VARIANT"
   its:locNote="Keep the leading space!"
   its:locNoteType="alert">
  <value> Variant {0} = {1} ({2})</value>
 </data>
 <data
   its:locNote="%1\$s is the original text's date in the format YYYY-MM-DD HH:MM always in GMT">
  <value>Translated from English content dated <span id="version-info">%1\$s</span> GMT.</value>
 </data>
```

```
</msgList>
```

[Source file: EX-locNote-selector-2.xml]

Note:

It is generally recommended to avoid using attributes to store text, however, in this specific case, the need to provide the notes without interfering with the structure of the host document is outweighing the drawbacks of using an attribute.

6.3.3 Markup Declarations for Localization Note

locNoteRule

[32]	`locNoteRule`	::=	`element its:locNoteRule { locNoteRule.content, locNoteRule.attributes }`
[33]	`locNoteRule.content`	::=	`locNote?`
[34]	`locNoteRule.attributes`	::=	`att.selector.attributes, attribute locNotePointer { string }?, attribute locNoteType { "alert" | "description" }, attribute locNoteRef { xsd:anyURI }?, attribute locNoteRefPointer { string }?`

locNote

[35]	`locNote`	::=	`element its:locNote { locNote.content, locNote.attributes }`
[36]	`locNote.content`	::=	`( text | ruby | span )*`
[37]	`locNote.attributes`	::=	`att.local.no-ns.attributes`

att.locNote

[38]	`att.locNote.attributes`	::=	`att.locNote.attribute.locNote, att.locNote.attribute.locNoteType, att.locNote.attribute.locNoteRef`
[39]	`att.locNote.attribute.locNote`	::=	`attribute its:locNote { string }?`
[40]	`att.locNote.attribute.locNoteType`	::=	`attribute its:locNoteType { "alert" | "description" }?`
[41]	`att.locNote.attribute.locNoteRef`	::=	`attribute its:locNoteRef { xsd:anyURI }?`

6.4 Terminology

6.4.1 Definition

The Terminology data category is used to mark terms and optionally associate them with information, such as definitions. This helps to increase consistency across different parts of the documentation. It is also helpful for translation.

Note:

Existing terminology standards such as [ISO 12200] and its derived formats are about coding terminology data, while the ITS Terminology data category simply allows to identify terms in XML documents and optionally to point to corresponding information.

6.4.2 Implementation

The Terminology data category can be expressed with global rules, or locally on an individual element. There is no inheritance. The default is that neither elements nor attributes are terms.

GLOBAL: The **termRule** element contains the following:

- A required **selector** attribute. It contains an XPath expression which selects the nodes to which this rule applies.
- A required **term** attribute with the value "yes" or "no".
- Exactly one of the following:
  - A **termInfoPointer** attribute that contains a relative XPath expression pointing to a node that holds the terminology information.
  - A **termInfoRef** attribute that contains a URI referring to the resource providing information about the term.
  - A **termInfoRefPointer** attribute that contains a relative XPath expression pointing to a node that holds the URI referring to the location of the terminology information.

Example 29: Usage of the termInfoPointer attribute

```
<text
  xmlns:its="http://www.w3.org/2005/11/its" >
 <its:rules version="1.0">
  <its:termRule selector="//term" term="yes" termInfoPointer="id(@def)"/>
 </its:rules>
 <p>We may define <term def="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>
```

[Source file: EX-terms-selector-1.xml]

Example 30: Usage of the termInfoRef attribute

```
<text
  xmlns:its="http://www.w3.org/2005/11/its" >
 <its:rules version="1.0">
  <its:termRule selector="//term[1]" term="yes" termInfoRef="#TDPV"/>
 </its:rules>
 <p>We may define <term>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>
```

[Source file: EX-terms-selector-2.xml]

Example 31: Usage of the termInfoRefPointer attribute

```
<text
  xmlns:its="http://www.w3.org/2005/11/its" >
 <its:rules 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>
```

[Source file: EX-terms-selector-3.xml]

LOCAL: The following local markup is available for the Terminology data category:

- A **term** attribute with the value "yes" or "no".
- An optional **termInfoRef** attribute that contains a URI referring to the resource providing information about the term.

Example 32: The Terminology data category expressed locally

```
<book
  xmlns:its="http://www.w3.org/2005/11/its"
  its:version="1.0">
 <head>...</head>
 <body> ...  <p>And he said: you need a
   new <quote
     its:term="yes">motherboard</quote>
  </p> ...
  </body>
</book>
```

[Source file: EX-terms-selector-4.xml]

6.4.3 Markup Declarations for Terminology

termRule

```
[42] termRule            ::= element its:termRule { termRule.content, termRule.attributes }
[43] termRule.content    ::= empty
[44] termRule.attributes ::= att.selector.attributes, attribute term { "yes" | "no" }, attribute termInfoRef
                             { xsd:anyURI }?, attribute termInfoRefPointer { string }?, attribute termInfoPointer
                             { string }?
```

att.term

```
[45] att.term.attributes            ::= att.term.attribute.termInfoRef, att.term.attribute.term
[46] att.term.attribute.termInfoRef ::= attribute its:termInfoRef { xsd:anyURI }?
[47] att.term.attribute.term        ::= attribute its:term { "yes" | "no" }?
```

6.5 Directionality

6.5.1 Definition

The Directionality data category allows the user to specify the base writing direction of blocks, embeddings and overrides for the Unicode bidirectional algorithm. It has four values: "ltr", "rtl", "lro" and "rlo".

> **Note:**
>
> ITS defines only the values of the Directionality data category and their inheritance. The behavior of text labeled in this way may vary, according to the implementation. Implementers are encouraged, however, to model the behavior on that described in the CSS 2.1 specification or its successor. In such a case, the effect of the data category's values would correspond to the following CSS rules:
>
> - Data category value: "ltr" (left-to-right text)
>
>   CSS rule: `*[dir="ltr"] { unicode-bidi: embed; direction: ltr}`
> - Data category value: "rtl" (right-to-left text)
>
>   CSS rule: `*[dir="rtl"] { unicode-bidi: embed; direction: rtl}`
> - Data category value: "rlo" (left-to-right override)
>
>   CSS rule: `*[dir="lro"] { unicode-bidi: bidi-override; direction: ltr}`
> - Data category value: "rlo" (right-to-left text)
>
>   CSS rule: `*[dir="rlo"] { unicode-bidi: bidi-override; direction: rtl}`
>
> More information about how to use this data category is provided by [Bidi Article].

6.5.2 Implementation

The Directionality data category can be expressed with global rules, or locally on an individual element. The information applies to the textual content of the element, *including* child elements and attributes. The default is that both elements and attributes have the directionality of left-to-right.

GLOBAL: The **dirRule** element contains the following:

- A required **selector** attribute. It contains an XPath expression which selects the nodes to which this rule applies.
- A required **dir** attribute with the value "ltr", "rtl", "lro" or "rlo".

Example 33: Document which needs global rules for directionality

In this document the right-to-left directionality is marked using a `direction` attribute with a value "rtlText".

```
<text xml:lang="en">
 <body>
  <par>In Hebrew, the title <quote xml:lang="he" direction="rtlText">פעילות הבינאום, W3C</quote>
     means <quote>Internationalization Activity, W3C</quote>.</par>
 </body>
</text>
```

[Source file: EX-dir-selector-1.xml]

Example 34: The Directionality data category expressed with global rules

The **dirRule** element indicates that all elements with an attribute `direction="rtlText"` have right-to-left content.

```
<its:rules
  xmlns:its="http://www.w3.org/2005/11/its"  version="1.0">
 <its:dirRule dir="rtl" selector="//*[@direction='rtlText']"/>
</its:rules>
```

[Source file: EX-dir-selector-2.xml]

LOCAL: The following local markup is available for the Directionality data category:

- A **dir** attribute with the value "ltr", "rtl", "lro" or "rlo".

Example 35: The Directionality data category expressed locally

On the first `quote` element, the `its:dir="rtl"` attribute indicates a right-to-left content.

```
<text
  xmlns:its="http://www.w3.org/2005/11/its"  xml:lang="en"
  its:version="1.0">
 <body>
  <par>In Arabic, the title <quote xml:lang="ar"
     its:dir="rtl">نشاط التدويل، W3C</quote>
     means <quote>Internationalization Activity, W3C</quote>.</par>
 </body>
</text>
```

[Source file: EX-dir-selector-3.xml]

6.5.3 Markup Declarations for Directionality

dirRule

```
[48] dirRule            ::= element its:dirRule { dirRule.content, dirRule.attributes }
[49] dirRule.content    ::= empty
[50] dirRule.attributes ::= att.selector.attributes, attribute dir { "ltr" | "rtl" | "lro" | "rlo" }
```

att.dir

```
[51] att.dir.attributes    ::= att.dir.attribute.dir
[52] att.dir.attribute.dir ::= attribute its:dir { "ltr" | "rtl" | "lro" | "rlo" }?
```

6.6 Ruby

6.6.1 Definition

The Ruby data category is used for a run of text that is associated with another run of text, referred to as the base text. Ruby text is used to provide a short annotation of the associated base text. It is most often used to provide a reading (pronunciation) guide.

6.6.2 Implementation

The Ruby data category can be expressed with global rules, or locally. There is no inheritance.

GLOBAL: The **rubyRule** element contains the following:

- A required **selector** attribute. It contains an XPath expression which selects the nodes to which this rule applies. This is the ruby base text.
- An optional **rubyPointer** attribute that contains a relative XPath expression pointing to a node that corresponds to the ruby element.
- An optional **rpPointer** attribute that contains a relative XPath expression pointing to a node that corresponds to the ruby parenthesis.
- An optional **rbcPointer** attribute that contains a relative XPath expression pointing to a node that corresponds to the ruby base container.
- An optional **rtcPointer** attribute that contains a relative XPath expression pointing to a node that corresponds to the ruby text container.
- An optional **rbspanPointer** attribute that contains a relative XPath expression pointing to a node that corresponds to the **rbspan** attribute.
- An optional **rubyText** element that contains the ruby text.
- An optional **rtPointer** attribute that contains a relative XPath expression pointing to a node that corresponds to the ruby text.

Note:

Where legacy formats do not contain ruby markup, it is still possible to associate ruby text with a specified range of document content using the **rubyRule** element.

Example 36: Adding ruby text with a rubyRule element

```
<text
  xmlns:its="http://www.w3.org/2005/11/its" >
 <head> ...
   <its:rules version="1.0">
   <its:rubyRule selector="/text/body/img[1]/@alt">
    <its:rubyText>World Wide Web Consortium</its:rubyText>
   </its:rubyRule>
  </its:rules>
 </head>
 <body>
  <img src="w3c_home.png" alt="W3C"/> ...
  </body>
</text>
```

[Source file: EX-ruby-legacy-1.xml]

LOCAL: In a document, the Ruby data category is realized with a **ruby** element. It contains the following:

- An **rb** element that contains the ruby base text and allows for local ITS markup.
- An **rp** element that contains the ruby parenthesis. It is used in case of simple markup to specify characters that can denote the beginning and end of ruby text when user agents do not have other ways to present ruby text distinctively from the base text.
- An **rt** element that contains the ruby text and allows for local ITS markup. It has an optional **rbspan** attribute. The **rbspan** attribute allows an **rt** element to span multiple **rb** elements.
- An **rbc** element that contains the ruby base container.
- An **rtc** element that contains the ruby text container.

All these elements share the attributes of the **span** element.

Example 37: The Ruby data category expressed locally

```
<text
  xmlns:its="http://www.w3.org/2005/11/its"
  its:version="1.0">
 <head> ... </head>
 <body>
  <p>この本は <its:ruby>
    <its:rb>慶応義塾大学</its:rb>
    <its:rp>（</its:rp>
    <its:rt>けいおうぎじゅくだいがく</its:rt>
    <its:rp>）</its:rp>
   </its:ruby>の歴史を説明するものです。</p>
 </body>
</text>
```

[Source file: EX-ruby-implementation-1.xml]

Note:

The structure of the content model for the **ruby** element is identical with the structure of ruby markup as defined in [Ruby-TR]. An implementation of the Ruby data category is encouraged, but not mandated follow the conformance criteria for ruby defined in that specification.

The structure of ruby defined in section 5.4 of [OpenDocument] is also compliant with ruby defined in this specification.

6.6.3 Markup Declarations for Ruby

rubyRule

[53] `rubyRule` ::= `element its:rubyRule { rubyRule.content, rubyRule.attributes }`

[54] `rubyRule.content` ::= `rubyText?`

[55] `rubyRule.attributes` ::= `att.selector.attributes, attribute rubyPointer { string }?, attribute rtPointer { string }?, attribute rpPointer { string }?, attribute rbcPointer { string }?, attribute rtcPointer { string }?, attribute rbspanPointer { string }?`

rubyText

[56] `rubyText` ::= `element its:rubyText { rubyText.content, rubyText.attributes }`

[57] `rubyText.content` ::= `text`

[58] `rubyText.attributes` ::= `att.local.no-ns.attributes, attribute rbspan { string }?`

ruby

[59] `ruby` ::= `element its:ruby { ruby.content, ruby.attributes }`

[60] `ruby.content` ::= `( rb, ( rt | ( rp, rt, rp ) ) ) | ( rbc, rtc, rtc? )`

[61] `ruby.attributes` ::= `att.local.no-ns.attributes`

rb

[62] `rb` ::= `element its:rb { rb.content, rb.attributes }`

[63] `rb.content` ::= `( text | span )*`

[64] `rb.attributes` ::= `att.local.no-ns.attributes`

rt

[65] `rt` ::= `element its:rt { rt.content, rt.attributes }`

[66] `rt.content` ::= `( text | span )*`

```
[67] rt.attributes ::= att.local.no-ns.attributes, attribute rbspan { string }?
```

rbc

```
[68] rbc            ::= element its:rbc { rbc.content, rbc.attributes }
[69] rbc.content    ::= rb+
[70] rbc.attributes ::= att.local.no-ns.attributes
```

rtc

```
[71] rtc            ::= element its:rtc { rtc.content, rtc.attributes }
[72] rtc.content    ::= rt+
[73] rtc.attributes ::= att.local.no-ns.attributes
```

rp

```
[74] rp            ::= element its:rp { rp.content, rp.attributes }
[75] rp.content    ::= text
[76] rp.attributes ::= att.local.no-ns.attributes
```

6.7 Language Information

6.7.1 Definition

The element **langRule** is used to express the language of a given piece of content. The **langPointer** attribute points to the markup which expresses the language of the text selected by the selector attribute. This markup MUST use values that conform to [BCP47]. The recommended way to specify language identification is to use `xml:lang`. The **langRule** element is intended only as a fall-back mechanism for documents where language is identified with another construct.

Example 38: Pointing to language information via langRule

The following **langRule** element expresses that the content of all `p` elements (including attribute values and textual content of child elements) are in the language indicated by `mylangattribute`, which is attached to the `p` elements, and expresses language using values conformant to [BCP47].

```
<its:rules
  xmlns:its="http://www.w3.org/2005/11/its"  version="1.0">
 <its:langRule selector="//p" langPointer="@mylangattribute"/>
</its:rules>
```

[Source file: EX-lang-definition-1.xml]

Note:

The Language Information data category only provides for rules to be expressed at a global level. Locally users are able to use `xml:lang` (which is defined by XML) or an attribute specific to the format in question (as in Example 38).

`xml:lang` is the preferable means of language identification. To ease the usage of `xml:lang`, a declaration for this attribute is part of the non-normative XML DTD and XML Schema document for ITS markup declarations. There is no declaration of `xml:lang` in the non-normative RELAX NG document for ITS, since in RELAX NG it is not necessary to declare attributes from the XML namespace.

Applying the Language Information data category to `xml:lang` attributes using global rules is not necessary, since `xml:lang` is the standard way to specify language information in XML. `xml:lang` is defined in terms of RFC 3066 or its successor ([BCP47] is the "Best Common Practice" for language identification and encompasses [RFC 3066] and its successors.)

6.7.2 Implementation

The Language Information data category can be expressed only with global rules. The information applies to the textual content of the element, *including* child elements and attributes. There is no default.

GLOBAL: The **langRule** element contains the following:

- A required **selector** attribute. It contains an XPath expression which selects the nodes to which this rule applies.
- A required **langPointer** attribute that contains a relative XPath expression pointing to a node that contains language information.

6.7.3 Markup Declarations for Language Information

langRule

```
[77] langRule            ::= element its:langRule { langRule.content, langRule.attributes }
[78] langRule.content    ::= empty
[79] langRule.attributes ::= att.selector.attributes, attribute langPointer { string }
```

6.8 Elements Within Text

6.8.1 Definition

The Elements Within Text data category reveals if and how an element affects the way text content behaves from a linguistic viewpoint. This information is for example relevant to provide basic text segmentation hints for tools such as translation memory systems. The values associated with this data category are:

- "yes" : The element and its content are part of the flow of its parent element. For example the element `strong` in [XHTML 1.0]:

  ```
  <strong>Appaloosa horses</strong> have spotted coats.
  ```

- "nested" : The element is part of the flow of its parent element, its content is an independent flow. For example the element `fn` in [DITA 1.0]:

```
Palouse horses<fn>A Palouse horse is the same as an Appaloosa.</fn> have spotted coats.
```

- "no" : The element splits the text flow of its parent element and its content is an independent text flow. For example the element `p` when inside the element `li` in DITA or XHTML:

```
<li>Palouse horses: <p>They have spotted coats.</p> <p>They have been bred by the Nez Perce.</p> </li>
```

6.8.2 Implementation

The Elements Within Text data category can be expressed only with global rules. There is no inheritance. The default is that elements are not within text.

GLOBAL: The **withinTextRule** element contains the following:

- A required **selector** attribute. It contains an XPath expression which selects the nodes to which this rule applies.
- A required **withinText** attribute with the value "yes", "no" or "nested".

Example 39: Specifying elements within text with a withinTextRule element

```
<its:rules
  xmlns:its="http://www.w3.org/2005/11/its"  version="1.0">
 <its:withinTextRule withinText="yes" selector="//b | //em | //i"/>
</its:rules>
```

[Source file: EX-within-text-implementation-1.xml]

6.8.3 Markup Declarations for Elements Within Text

withinTextRule

```
[80] withinTextRule            ::= element its:withinTextRule { withinTextRule.content, withinTextRule.attributes }
[81] withinTextRule.content    ::= empty
[82] withinTextRule.attributes ::= att.selector.attributes, attribute withinText { "yes" | "no" | "nested" }
```

A References

BCP47
Addison Phillips, Mark Davis. *Tags for Identifying Languages*, September 2006. Available at http://www.rfc-editor.org/rfc/bcp/bcp47.txt.

QAFRAMEWORK
Karl Dubost, Lynne Rosental, Dominique Hazaël-Massieux, Lofton Henderson. *QA Framework: Specification Guidelines*. W3C Recommendation 17 August 2005. Available at http://www.w3.org/TR/2005/REC-qaframe-spec-20050817/. The latest version of QAFRAMEWORK is available at http://www.w3.org/TR/qaframe-spec/.

RELAX NG
Information technology -- Document Schema Definition Language (DSDL) -- Part 2: *Regular-grammar-based validation -- RELAX NG*. International Organization for Standardization (ISO) ISO/IEC 19757-2:2003.

RFC 2119
S. Bradner. Key Words for use in RFCs to Indicate Requirement Levels. IETF RFC 2119, March 1997. Available at http://www.ietf.org/rfc/rfc2119.txt.

RFC 3987
Martin Dürst, Michel Suignard. *Internationalized Resource Identifiers (IRIs)*. RFC 3987, January 2005. See http://www.ietf.org/rfc/rfc3987.txt.

XLink 1.0
Steve DeRose, Eve Maler, David Orchard. *XML Linking Language 1.0*. W3C Recommendation 27 June 2001. Available at http://www.w3.org/TR/2001/REC-xlink-20010627/. The latest version of XLink 1.0 is available at http://www.w3.org/TR/xlink/.

XML 1.0
Tim Bray, Jean Paoli, C.M. Sperberg-McQueen, et al., editors. *Extensible Markup Language (XML) 1.0 (Fourth Edition)*, W3C Recommendation 16 August 2006. Available at http://www.w3.org/TR/2006/REC-xml-20060816/. The latest version of XML 1.0 is available at http://www.w3.org/TR/REC-xml/.

XML Infoset
John Cowan, Richard Tobin. *XML Information Set (Second Edition)*. W3C Recommendation 4 February 2004. Available at http://www.w3.org/TR/2004/REC-xml-infoset-20040204/. The latest version of XML Infoset is available at http://www.w3.org/TR/xml-infoset/.

XML Names
Tim Bray, Dave Hollander, Andrew Layman, Richard Tobin. *Namespaces in XML (Second Edition)*. W3C Recommendation 16 August 2006. Available at http://www.w3.org/TR/2006/REC-xml-names-20060816/. The latest version of XML Names is available at http://www.w3.org/TR/REC-xml-names/.

XML Schema
Henry S. Thompson, David Beech, Murray Maloney, Noah Mendelsohn. *XML Schema Part 1: Structures Second Edition*. W3C Recommendation 28 October 2004. Available at http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/. The latest version of XML Schema is available at http://www.w3.org/TR/xmlschema-1/.

XPath 1.0
James Clark. *XML Path Language (XPath) Version 1.0*. W3C Recommendation 16 November 1999. Available at http://www.w3.org/TR/1999/REC-xpath-19991116. The latest version of XPath 1.0 is available at http://www.w3.org/TR/xpath.

B References (Non-Normative)

Bidi Article
Richard Ishida. *What you need to know about the bidi algorithm and inline markup*. Article of the W3C Internationalization Activity, June 2005.

CSS 2.1
Bert Bos, Tantek Çelik, Ian Hickson Håkon Wium Lie. *Cascading Style Sheets, level 2 revision 1 CSS 2.1 Specification*. W3C Working Draft 11 April 2006. Available at http://www.w3.org/TR/2006/WD-CSS21-20060411/. The latest version of CSS2 is available at http://www.w3.org/TR/CSS21/.

DITA 1.0
Michael Priestley, JoAnn Hackos, et. al., editors. *OASIS Darwin Information Typing Architecture (DITA) Language Specification v1.0*. OASIS Standard 9 May 2005. Available at http://www.oasis-open.org/committees/download.php/15316/dita10.zip.

DocBook
Norman Walsh and Leonard Muellner. *DocBook: The Definitive Guide*. Available at http://www.docbook.org/.

l10n i18n
Richard Ishida, Susan Miller. Localization vs. Internationalization. Article of the W3C Internationalization Activity, January 2006.

ISO 12200
(International Organization for Standardization). *Machine-readable terminology interchange format (MARTIF) -- Negotiated interchange*. [Geneva]: International Organization for Standardization, 1999.

ITS REQ
Yves Savourel. *Internationalization and Localization Markup Requirements*. W3C Working Draft 18 May 2006. Available at http://www.w3.org/TR/2006/WD-itsreq-20060518/. The latest version of ITS REQ is available at http://www.w3.org/TR/itsreq/.

Localizable DTDs
Richard Ishida, Yves Savourel *Requirements for Localizable DTD Design*. Working Draft 7 July 2003. Available at http://people.w3.org/rishida/localizable-dtds/.

NVDL
Information technology -- Document Schema Definition Languages (DSDL) -- Part 4: *Namespace-based Validation Dispatching Language (NVDL)*. International Organization for Standardization (ISO) ISO/IEC 19757-4:2003.

OpenDocument
Michael Brauer et al. *OASIS Open Document Format for Office Applications (OpenDocument).*. Oasis Standard 1 May 2005. Available at http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office. The latest version of OpenDocument is available at http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office.

RFC 3066
Harald Alvestrand. *Tags for the Identification of Languages*. RFC 3066, January 2001. Available at http://www.ietf.org/rfc/rfc3066.txt.

Ruby-TR
Marcin Sawicki (until 10 October, 1999), Michel Suignard, Masayasu Ishikawa (石川　雅康), Martin Dürst, Tex Texin, *Ruby Annotation*. W3C Recommendation 31 May 2001. Available at http://www.w3.org/TR/2001/REC-ruby-20010531/ . The latest version of Ruby Annotation is available at http://www.w3.org/TR/ruby/.

Schematron
Information technology -- Document Schema Definition Languages (DSDL) -- Part 3: *Rule-based validation -- Schematron*. International Organization for Standardization (ISO) ISO/IEC 19757-3:2003.

TEI
Lou Burnard and Syd Bauman (eds). *Text Encoding Initiative Guidelines development version (P5)*. TEI Consortium, Charlottesville, Virginia, USA, Text Encoding Initiative.

XHTML 1.0
Steven Pemberton et al. *XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition)*. W3C Recommendation 26 January 2000, revised 1 August 2002. Available at http://www.w3.org/TR/2002/REC-xhtml1-20020801/. The latest version of XHTML 1.0 is available at http://www.w3.org/TR/xhtml1/.

XML i18n BP
Yves Savourel, Diane Stoick. *Best Practices for XML Internationalization*. Available at http://www.w3.org/TR/2006/WD-xml-i18n-bp-20060518/. The latest version of xml-i18n-bp is available at http://www.w3.org/TR/xml-i18n-bp/.

XMLSPEC
The XML Spec Schema and Stylesheets. Available at http://www.w3.org/2002/xmlspec/.

XSLT 1.0
James Clark. *XSL Transformations (XSLT) Version 1.0*. W3C Recommendation 16 November 1999. Available at http://www.w3.org/TR/1999/REC-xslt-19991116. The latest version of XSLT 1.0 is available at http://www.w3.org/TR/xslt.

XUL
exTensible User Interface Language. Available at http://www.xulplanet.com/.

C Summary of ITS Markup (Non-Normative)

This section is informative.

The following list summarizes elements relating to global rules and their attributes:

- **<rules>** Container for global rules.
  - **href**
    Pointer to external rules files.
  - **type**
    Type of pointer to external rules files.
    Legal values are:
    - *simple*
  - **version**
    Version of the ITS schema.
- **<dirRule>** Rule about the Directionality data category.
  - **dir**
    The text direction for the selection.
    Legal values are:
    - *ltr*
    - *rtl*
    - *lro*
    - *rlo*
  - **selector**
    XPath expression identifying the nodes to be selected.
- **<langRule>** Rule about the Language Information data category.
  - **langPointer**
    Relative XPath expression pointing to a node that contains language information.
  - **selector**

XPath expression identifying the nodes to be selected.

- **<locNote>** Contains a localization note.
    - **translate**
      The Translate data category information to be attached to the current node.
    - **locNote**
      Localization note.
    - **locNoteType**
      The type of localization note.
    - **locNoteRef**
      URI referring to the location of the localization note.
    - **termInfoRef**
      Pointer to a resource containing information about the term.
    - **term**
      Indicates a term locally.
    - **dir**
      The text direction for the context.
- **<locNoteRule>** Rule about the Localization Note data category.
    - **locNotePointer**>
      Relative XPath expression pointing to a node that holds the localization note.
    - **locNoteType**
      The type of localization note.
      Legal values are:
        - *alert*
        - *description*
    - **locNoteRef**
      URI referring to the location of the localization note.
    - **locNoteRefPointer**
      Relative XPath expression pointing to a node that holds the URI referring to the location of the localization note.
    - **selector**
      XPath expression identifying the nodes to be selected.
- **<termRule>** Rule about the Terminology data category.
    - **term**
      Indicates whether the selection is a term or not.
      Legal values are:
        - *yes*
        - *no*
    - **termInfoRef**
      URI referring to the resource providing information about the term.
    - **termInfoRefPointer**
      Relative XPath expression pointing to a node containing a URI referring to the resource providing information about the term.
    - **termInfoPointer**
      Relative XPath expression pointing to a node containing information about the term.
    - **selector**
      XPath expression identifying the nodes to be selected.
- **<translateRule>** Rule about the Translate data category.
    - **translate**
      The Translate data category information to be applied to selected nodes.
      Legal values are:
        - *yes*
        - *no*
    - **selector**
      XPath expression identifying the nodes to be selected.
- **<withinTextRule>** Rule about the Elements Within Text data category.
    - **withinText**
      States whether current context is regarded as "within text".
      Legal values are:
        - *yes*
          no
          nested
    - **selector**
      XPath expression identifying the nodes to be selected.
- **<rubyRule>** Rule about the Ruby data category.
    - **rubyPointer**
      Relative XPath expression pointing to a node that corresponds to a `ruby` element
    - **rtPointer**
      Relative XPath expression pointing to a node that corresponds to a `rt` element
    - **rpPointer**
      Relative XPath expression pointing to a node that corresponds to a `rp` element

    - **rbcPointer**
      Relative XPath expression pointing to a node that corresponds to a `rbc` element
    - **rtcPointer**
      Relative XPath expression pointing to a node that corresponds to a `rtc` element
    - **rbspanPointer**
      Relative XPath expression pointing to a node that corresponds to a rbspan attribute.
    - **selector**
      XPath expression identifying the nodes to be selected.

The following list summarizes elements that are available for local use:

- **<span>** Inline element to contain ITS information.
- **<rb>** Ruby base text.
- **<rbc>** Container for rb elements in the case of complex ruby markup.
- **<rp>** Used in the case of simple ruby markup to specify characters that can denote the beginning and end of ruby text when user agents do not have other ways to present ruby text distinctively from the base text.
- **<rt>** Ruby text.
- **<rtc>** Container for rt elements in the case of complex ruby markup.
- **<ruby>** Ruby markup.

The following list summarizes attributes that are available for local use, with the local elements mentioned above, or with other elements in a host schema:

- **translate**
  The Translate data category information to be attached to the current node.

- **locNote**
  Localization note.
- **locNoteType**
  The type of localization note.
- **locNoteRef**
  URI referring to the location of the localization note.

- **termInfoRef**
  Pointer to a resource containing information about the term.
- **term**
  Indicates a term locally.

- **dir**
  The text direction for the context.

D Schemas for ITS (Non-Normative)

This section is informative.

The following schemas define ITS elements and attributes and could be used as building blocks when you want to integrate ITS markup into your own XML vocabulary. You can see examples of such integration in Best Practices for XML Internationalization. The schemas are not intended to be used alone for validation of documents with ITS markup.

The following schemas are provided:

- DTD for ITS
- XML Schema document for ITS
- RELAX NG compact syntax document for ITS
- RELAX NG XML syntax document for ITS

E Checking ITS Markup Constraints With Schematron (Non-Normative)

This section is informative.

Several constraints of ITS markup cannot be validated with ITS schemas. The following [Schematron] document allows for validating some of these constraints.

Example 40: Testing constraints in ITS markup

```
<sch:schema
  xmlns:sch="http://www.ascc.net/xml/schematron" >
<!-- Schematron document to test constraints for global and local ITS markup.
 For ITS markup definitions, see http://www.w3.org/TR/its/ . -->

 <sch:ns prefix="its" uri="http://www.w3.org/2005/11/its"/>
 <sch:pattern
   name="Check ITS Global Rules and Local Constraints, and Version Constraints">
  <sch:rule context="*">
<!-- Tests for locNoteRule -->

   <sch:report
```

```
     test="self::its:locNoteRule and child::its:locNote and @its:locNotePointer">
   locNoteRule error: A locNoteRule element must not have both a locNote child element
    and a locNotePointer attribute.</sch:report>
   <sch:report
     test="self::its:locNoteRule and @its:locNoteRef and @its:locNoteRefPointer">
   locNoteRule error: A locNoteRule element must not have both a locNoteRef attribute
    and a locNoteRefPointer attribute.</sch:report>
   <sch:report
     test="self::its:locNoteRule and child::its:locNote and @its:locNoteRef">
   locNoteRule error: A locNoteRule element must not have both a locNote child element
   and a locNoteRef attribute.</sch:report>
<!-- Test for termRule -->

   <sch:report
     test="self::its:termRule and @its:termInfoRef and @its:termInfoRefPointer">
   termRule error: A termRule element must not have both a termInfoRef attribute and a
   termInfoRefPointer attribute.</sch:report>
   <sch:report
     test="self::its:termRule and @its:termInfo and @its:termInfoPointer">
   termRule error: A termRule element must not have both a termInfo attribute and a
   termInfoPointer attribute.</sch:report>
   <sch:report
     test="self::its:termRule and @its:termInfoRef and @its:termInfoPointer">
   termRule error: A termRule element must not have both a termInfoRef attribute and a
   termInfoPointer attribute.</sch:report>
<!-- Test for rubyRule -->

   <sch:report
     test="self::its:rubyRule and child::its:rubyText and @its:rtPointer">
   rubyRule error: A rubyRule element must not have both a rubyText child element and
   a rtPointer attribute.</sch:report>
<!-- Test for locNote (local) -->

   <sch:report test="@its:locNote and @its:locNoteRef">
   Local ITS usage error: The locNote attribute and the locNoteRef attribute
   must not be used together.</sch:report>
<!-- Test for term (local) -->

   <sch:report
     test="@its:termInfoRef and not(its:term) and not(self::its:termRule)">
   Local ITS usage error: A termInfoRef attribute must not appear locally without
   a term attribute.</sch:report>
<!-- Version attribute test -->

   <sch:report test="/*/@its:version != @its:version">
   The version attribute at the root element and at the rules element
   must not specify different versions of ITS.</sch:report>
  </sch:rule>
 </sch:pattern>
</sch:schema>
```

[Source file: its-constraints-check-schematron.xml]

F Checking ITS Markup with NVDL (Non-Normative)

This section is informative.

The following [NVDL] document allows validation of ITS markup which has been added to a host vocabulary. Only ITS elements and attributes are checked. Elements and attributes of host language are ignored during validation against this NVDL document/schema.

Example 41: NVDL schema for ITS

```
<rules xmlns="http://purl.oclc.org/dsdl/nvdl/ns/structure/1.0">
 <namespace ns="http://www.w3.org/2005/11/its">
  <validate schema="its-elements.rng"/>
 </namespace>
 <namespace ns="http://www.w3.org/2005/11/its" match="attributes">
  <validate schema="its-attributes.rng"/>
 </namespace>
 <anyNamespace>
  <allow/>
 </anyNamespace>
</rules>
```

[Source file: its.nvdl]

The NVDL schema depends on the following two schemas:

- RELAX NG schema for ITS elements
- RELAX NG schema for ITS attributes

G Revision Log (Non-Normative)

The following log records major changes that have been made to this document between the publication in November 2005 and the publication in February 2006.

1. A section about basic concepts of ITS has been created.
2. Terminology has been modified: the terms for position of ITS information in situ versus dislocated have been replaced by selection in an instance document versus global, rule-based selection.
3. The definition of the Directionality data category has been changed, to be compliant to various other specifications. See the comment

on bidirectionality for further information.

4. Terminology within the text of this document and within the markup declarations has been modified: scope of ITS information has been replaced with selection of ITS information.
5. The schemaRules element has been removed. For ITS information as schema annotation, where is now only a`schemaRule` element.
6. All ITS attributes are now defined as qualified attributes. This leads to changes in the generated ITS schemas, for example the generation of parameter entities for prefixes in the XML DTD. This allows for easy changing of prefixes in element or attribute names.
7. The possibility of selector attributes in instance documents (in the previous draft this was called scope in an instance document) has been removed. Local selection in an instance document now relies only on default selections of data categories. Due to this change, the definition of precedence between selections and conformance criteria have been simplified, and the issue on namespace requirements and selector values could be resolved.
8. Definitions of default selections of data categories have been modified.
9. An `ns` element has been added to the **rules** element to allow for specifying namespace bindings.
10. The implementation of the Ruby data category has been modified, to reflect the removal of selector attributes in instance documents.
11. A section on mapping of ITS data categories to existing markup has been created.
12. Examples of integrating ITS markup into a TEI schema and into XML Spec have been created.
13. A **span** element has been created.
14. The examples have been modified to reflect changes mentioned above.
15. For clarity, various sections have been reworded and re-structured, and the visualization of ITS markup within the text of this document has been modified.
16. Tracking of issues is now handled via Bugzilla.
17. A revision log has been added.

The following log records major changes that have been made to this document between the publication in February 2006 and the publication in April 2006.

1. The `schemaRule` element, and the notion of schema annotation which was connected to it, have been abandoned.
2. The section on conformance has been rewritten and placed at the beginning of the document.
3. In global rules, the `documentRule` element has been replaced with elements which have data category specific names. This eases the creation and validation of global rules.
4. In global rules, instead of rule specific attributes for selection, there is now just one **selector** attribute.
5. In global rules, the `documentRules` element has been renamed to **rules**.
6. In global rules, in addition to the existing functionality of adding ITS information to selected nodes, a new functionality of pointing to information in an XML document has been created.
7. An XLink **href** attribute has been added to the **rules** element to allow for links to external rules. The precedence between selections has been modified to reflect this change.
8. Two new data categories Language Information and Elements Within Text have been defined.
9. The data category Ruby has been redefined, to be conforming to [Ruby-TR].
10. The declarations for ITS markup have been rewritten, to adopt the changes mentioned above.
11. The declarations for ITS markup (formally all assembled in a single section) have been separated and placed in the sections there they are described.
12. A modularization of ITS and XHTML 1.0 has been created.
13. The informative Section 1: Introduction and Section 2: Basic Concepts have been rewritten.
14. Examples and the modularizations of ITS with existing markup schemes have been changed to reflect the modifications mentioned above.

The following log records major changes that have been made to this document between the publication in April 2006 and the publication in May 2006.

1. The conformance section has been rewritten.
2. The terminology of mapping ITS data categories with existing markup has been changed to associating ITS Data Categories with existing markup.
3. The global rule elements have been rewritten to have attributes in the empty namespace.
4. The examples have been validated, partially modified and extended.
5. Documentation strings have been added to elements and attributes.
6. The Elements Within Text data category has been redefined.
7. Clarifications about ITS and inclusion mechanisms and selections of pointers to external (possibly non-textual) objects have been made.
8. The local attributes have been reorganized into various data category specific groups.
9. A versioning mechanism has been introduced
10. A summary of ITS markup has been created.
11. The automatically generated ITS schemas have been augmented with element and attribute documentations.
12. A description of ITS markup constraints not covered by the ITS schemas has been created.
13. The attributes termRef and termRefPointer have been renamed to **termInfoRef** and **termInfoRefPointer**.
14. A list of requirements which are formulated in [ITS REQ], but not covered in this document, has been created. See Section 1: Introduction.

The following log records major changes that have been made to this document between the publication in May 2006 and the publication in November 2006.

1. In response to issue 3290 (Introduction too "positive" and not enough informative) Section 1: Introduction has been modified.
2. In response to issue 3293 (Conformance/Compliance) Section 4: Conformance has been changed.
3. In response to issue 3318 (Clarify "within text" data category) the definition of the Elements Within Text data category has been clarified and illustrated with examples.
4. In response to issue 3321 (Relation of terminology data category to existing standards for terminology) the relation has been described in Section 6.4.1: Definition.

5. In response to issue 3323 (Version of XPath: Write "XPath 1.0 or its successor") the reference to [XPath 1.0] has been replaced by a reference to "[XPath 1.0] or its successor".
6. In response to issue 3330 (lro and rlo explanation) explanations have been modified in Section 6.5.1: Definition.
7. In response to issue 3456 (Attribute used for locInfo) a note has been added at the end of Section 6.3.2: Implementation.
8. In response to issue 3457 (RFC 3066bis) the references to "RFC 3066bis" been replaced by a reference to "RFC4646 or its successor".
9. In response to issue 3458 (Default translate value) default values have been made explicit in Section 6.2.2: Implementation.
10. In response to issue 3459 (Use of elements) ITS markup is now explicitly allowed inside other ITS markup.
11. In response to issue 3460 (Loc Info or Loc Note) all attributes and elements starting with `locInfo...` have been renamed `locNote...`.
12. In response to issue 3461 (Role of termInfo) Section 6.4.1: Definition has been clarified.
13. In response to issue 3462 (Pointing to terms) Section 6.4.2: Implementation has been clarified.
14. In response to issue 3463 (termInfoRef should allow for id strings) the **termInfoPointer** attribute has been added.
15. In response to issue 3464 (Allowing for XPath expressions to point to term definitions) the **termInfoPointer** attribute has been added.
16. In response to issue 3465 (Default is ltr) Section 6.5.2: Implementation has been clarified.
17. In response to issue 3466 (Existing ruby markup) Section 3.5: Usage of Internationalized Resource Identifiers in ITS has been added, the conformance clause 2-2 has been removed, the reference to [Ruby-TR] has been clarified and moved to Section 6.6.2: Implementation, and the definition of the Directionality data category has been clarified in Section 6.5.1: Definition.
18. In response to issue 3467 (rubyText is an attribute) the `rubyText` attribute has been changed to the **rubyText** element.
19. In response to issue 3468 (xml:lang missing) a note at the end of Section 6.7.1: Definition has been added.
20. In response to issue 3469 (Example 19 xml:lang) the role of `xml:lang` for the language information data category has been clarified in Section 6.7.1: Definition.
21. In response to issue 3473 (Language information data category overview) the role of the **langRule** element has been clarified in Section 6.7.1: Definition.
22. In response to issue 3479 (Example 33) Example 38 (which was Example 33) has been modified.
23. In response to issue 3480 (Language information data category overview) the role of `xml:lang` for the language information data category has been clarified in Section 6.7.1: Definition.
24. In response to issue 3481 (Translatability) the data category "Translatability" has been renamed to Translate.
25. In response to issue 3482 (Inheritance of translation information) the role of inheritance of translation information within global rules has been made more explicit in Section 6.2.2: Implementation.
26. In response to issue 3487 (Invert translate examples) Example 23 has been modified.
27. In response to issue 3488 (6.3 ed 1) Section 6.3.1: Definition has been modified.
28. In response to issue 3489 (Mention translation tools) a reference to translation tools has been added in Section 6.3.1: Definition.
29. In response to issue 3490 (Examples 22 and 23) Example 24 (which was Example 22) and Example 25 (which was example 23) have been modified.
30. In response to issue 3491 (Explanations for examples) the examples in Section 6.3.2: Implementation have been clarified.
31. In response to issue 3492 (Examples 24) Example 26 (which was Example 24) has been modified.
32. In response to issue 3493 (Marks terms and meanings) the purpose of the terminology data category has been clarified in Section 6.4.1: Definition.
33. In response to issue 3494 (termInfoRefPointer referent's data) the explanation of the termInfoRefPointer attribute has been clarified in Section 6.4.2: Implementation.
34. In response to issue 3495 (Hard to know what this is about) Section 6.8: Elements Within Text has been clarified.
35. In response to issue 3496 (Standardized wording) all subsections in Section 6: Description of Data Categories have been reworded to be consistent.
36. In response to issue 3497 (No implementation section) an implementation section (Section 6.7.2: Implementation) has been added for the language information data category.
37. In response to issue 3498 (Repetition) Section 6.5.1: Definition has been modified.
38. In response to issue 3499 (Is dir mandatory?) it has been made explicit that the **dir** attribute is mandatory at the **dirRule** element (see Section 6.5.2: Implementation).
39. In response to issue 3500 (Avoid xml:lang='he') Section 5.2.2: Local Selection in an XML Document (especially Example 15) has been modified.
40. In response to issue 3501 (Some Hebrew quotation) Section 6.5.2: Implementation has been modified.
41. In response to issue 3502 (Example 30) Example 33 (which was Example 30) has been modified.
42. In response to issue 3503 (Refer to bidi article) a reference to [Bidi Article] in Section 6.5.1: Definition has been added.
43. In response to issue 3504 (Example 31 lacks rp) Example 37 (which was Example 31) has been modified.
44. In response to issue 3505 (Use Japanese in Example 31) Example 37 (which was Example 31) has been modified.
45. In response to issue 3506 (Make the application of legacy ruby clearer) Section 6.6.2: Implementation has been modified.
46. In response to issue 3507 (In the case of no selection) Section 6.6.2: Implementation has been modified.
47. In response to issue 3508 (Example 32 head) Example 36 (which was Example 32) has been modified.
48. In response to issue 3509 (Too many places to look) the section 5.1 Summary of ITS Markup has been changed to Appendix C: Summary of ITS Markup, and Section 6: Description of Data Categories has been reworded.
49. In response to issue 3510 (Spell out the attributes) Section 5.2.1: Global, Rule-based Selection has been modified to list relevant attributes.
50. In response to issue 3511 (Example 14 translate) Example 15 (which was Example 14) has been modified.
51. In response to issue 3512 (Example 14 dir) Example 15 (which was Example 14) has been modified.
52. In response to issue 3513 (ITS markup must be integrated) the optionality of ITS markup in the targeted XML file(s) has been made explicit in Section 5.5: Associating ITS Data Categories with Existing Markup.
53. In response to issue 3514 (Attributes missing?) the attributes that are available for local use have been spelled out in Appendix C: Summary of ITS Markup.
54. In response to issue 3515 (xml:lang = language info, please) the role of `xml:lang` has been clarified in Section 6.7.1: Definition.
55. In response to issue 3516 (Editorial comments on ITS) Section 1: Introduction, Section 2: Basic Concepts, Section 3: Notation and Terminology, and Section 6: Description of Data Categories have been modified.
56. In response to issue 3612 (Missing term="yes|no" in termRule element) the value "no" has been reinstated for the **term** attribute.

57. In response to issue 3640 (Need to handle inheritance for terminology) the description of the inheritance / default / overriding behavior of the data categories has been clarified in Section 6.1: Position, Defaults, Inheritance and Overriding of Data Categories.
58. In response to issue 3803 (Both selector and the rbPointer attributes are base text in rubyRule) the `rbPointer` has been removed.

The following log records major changes that have been made to this document between the publication in November 2006 and January 2007.

1. In response to issue 3319 (Namespace binding mechanism) removed the ns element.
2. In response to issue 4096 (Inheritance and overriding needs to be made clearer) changed wording.
3. In response to issue 4098 (Need to make more explicit: what is possible with XSLT patterns realizing ITS selectors?) added a note about the subset of XPath in XSLT to Section 2.1.2: Global Approach.
4. In response to issue 4099 (Format for implementation output needs to be clear) added note on the Internationalization Tag Set (ITS) Version 1.0 testsuite processing to Section 4.2: Conformance Type 2: The Processing Expectations for ITS Markup.
5. In response to issue 4152 (Need to make explicit that ITS attributes at its:span are not namespace qualified) created two groups of ITS attributes, namespace qualified and not namespace qualified, the latter used at ITS **span**, the former for use at elements not from the ITS namespace.
6. In response to issue 4289 (Various mainly editorial changes) made the editorial changes described in the issue.
7. In response to issue 4290 (Create appendix about NVDL) introduced Appendix F: Checking ITS Markup with NVDL.
8. In response to issue 4291 (Have version attribute at its:rules element in no namespace) put the ITS version attribute at the **rules** element in no namespace.
9. In response to issue 4293 (Editorial fix in sec. 6.6.2) removed bullet point.
10. In response to issue 4294 (Add xlink:type to its:rules element) added the XLink **type** attribute to the **rules** element.
11. In response to issue 4295 (Change of XPath expressions in Schematron example) updated Appendix E: Checking ITS Markup Constraints With Schematron.
12. In response to issue 4322 (Editors to go to the draft and check that there are no "CSS style elements" or the like) removed wording related to CSS style elements.

The following log records major changes that have been made to this document between the publication in February 2007 and March 2007.

1. In response to comments received during the AC review made the following changes: replaced the examples 3 (EX-ways-to-use-its-1.xml), 4 (EX-ways-to-use-its-2.xml), 5 (EX-ways-to-use-its-3.xml), 6 (EX-ways-to-use-its-4.xml) and 7 (EX-ways-to-use-its-5.xsd) with other examples; updated the bibliography.
2. Fixed various (spelling etc.) errors in examples.
3. General spell check.
4. Updated reference for language identification in Section 6.7: Language Information.

H Acknowledgements (Non-Normative)

This document has been developed with contributions by the ITS Working Group: Damien Donlon (Sun Microsystems, Inc.), Martin Dürst (W3C Invited Expert), Poonam Gupta (Centre for Development of Advanced Computing (CDAC)), Richard Ishida (W3C/ERCIM), Jirka Kosek (W3C Invited Expert), Christian Lieske (SAP AG), Sebastian Rahtz (W3C Invited Expert), Francois Richard (HP), Goutam Saha (Centre for Development of Advanced Computing (CDAC)), Felix Sasaki (W3C/Keio), Yves Savourel (ENLASO Corporation), Diane Stoick (The Boeing Company), Najib Tounsi (Ecole Mohammadia d'Ingenieurs Rabat (EMI)), Andrzej Zydron (W3C Invited Expert).

A special thanks goes to Sebastian Rahtz who introduced us to the ODD language, which was used to create this document, and who provided the stylesheets to generate schemas and the XHTML version out of an ODD document. The generation of XHTML from ODD takes an intermediate step through the xmlspec-i18n.dtd.