Internationalization and Localization of XML: Introducing "ITS"

Christian Lieske

SAP AG

christian.lieske@sap.com

Sebastian Rhatz

Oxford University

Sebastian.Rahtz@oucs.ox.ac.uk

Felix Sasaki

W3C

fsasaki@w3.org

Introduction

This paper describes a new markup vocabulary called "Internationalization Tag Set" (ITS) *[lies06]*. ITS provides elements and attributes for the internationalization and localization of XML documents and schemas. ITS is formulated in a schema language independent manner and implemented in XML DTD, W3C XML Schema and RELAX NG. It can be integrated into new or existing vocabularies of a great variety, like DITA, DocBook, HTML or TEI.

The paper provides a general introduction to ITS and is organized as follows:

- First, we will explain background information about ITS, including the motivation for creating ITS and potential users and usage scenarios.

- The next section describes “basic concepts of ITS”. This encompasses the notion of “data categories” for internationalization and localization purposes, and the “selection” of nodes in XML documents to which they are related to. The data categories can be applied “globally” or “locally” in XML documents. They can be used for “adding” ITS information to nodes, or for “pointing” into documents to existing, ITS related information.

- The section “overview of ITS data categories” describes roughly the purpose of all categories which are defined in the current ITS working draft.

- Finally, the section “ITS specification development using ODD” describes how the ODD language has been applied for the creation of the ITS working draft, and the automatic generation of ITS markup declarations in the formats XML DTD, XML Schema and RELAX NG.

Background: about ITS

The “Internationalization Tag Set (ITS)” is a standard under development (current status is “Public Working Draft”) by the W3C ITS Working Group. ITS defines a standard which aims to enable worldwide use and effective localization of content of schemas and XML instances (both existing ones and new ones).

On the one hand, the standard is defined conceptually through the notion of data categories. On the other hand, it is defined through implementations of these data categories as a set of elements and attributes. The standard provides examples of how ITS can be used with existing popular markup schemes such as XHTML, DocBook, DITA, and TEI (the ITS Working Group will address this in a separate document/note on “Modularizations for ITS”). Furthermore, it provides implementations of ITS in three schema languages: XML DTD, XML Schema and RELAX NG.

Requirements for ITS are formulated in *[sav05]*. Currently, not all of these requirements are addressed. The ITS Working Group will cover some of the requirements in a separate document on techniques for the internationalization and localization of schemas and XML instances documents.

The Working Group welcomes any feedback on the “Internationalization Tag Set (ITS)” and on the accompanying documents.

Motivation for ITS

Content or software that is authored in one language (the 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, XML based document formats 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 demarcate 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 *[ishi06]*.

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 (such as the eXtensible User Interface Language XUL) creates challenges and opportunities in the domain of XML internationalization and localization.

The following example sketches one of the issues that currently hinders efficient XML-related localization: the lack of a standard, declarative mechanism which identifies which parts of an XML document need to be translated (the text in bold face shows the parts that need to be translated). Tools often cannot automatically do this identification.

In the example below, there are no clear mechanisms for making explicit which `<string>` element needs to be translated. (Translatable content is set on bold face.)

```
<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>
```

Out of Scope of ITS

ITS does not exhaustively cover all mechanisms and data formats which might be needed for configuring localization workflows or tools to process a specific format. These mechanisms and data formats, sometimes called “Localization Properties”, however, possibly may be implemented by the framework put forth in this standard. (“XML localization properties” is a generic term to name the mechanisms and data formats that allows 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.)

Users and Usages of ITS

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

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

- Schema developers who start a schema from scratch：

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

- Schema developers who work with an existing schema:

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

In some cases, an existing schema may already contain markup equivalent to that recommended in ITS. Then it is not necessary to add duplicate markup since ITS provides mechanisms for relating ITS markup to markup in the host vocabulary which serves a similar purpose. Before using these mechanisms, however, the developers should check that the behaviour associated with the markup in their own schema is fully compatible with the expectations described in the ITS specification.

- Vendors of content-related tools:

This type of users encompasses providers of tools for authoring, translation or other flavours 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 ITS will make the job of vendors easier by standardising the format and processing expectations of certain relevant markup items, and allowing them to more effectively identify how content should be handled.

- Content producers and architects:

This type of users comprises authors, translators and other types of content authors. They may use ITS to mark up specific bits of content. For content authors who need specific support for their local script, ITS provides the data categories “directionality” (for scripts like Hebrew or Arabic) and “ruby” (mainly, but not exclusively for Japanese or Chinese).

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

An example: A content producer may use an attribute on a particular element to express that the content of an element should not be translated.

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

A content author or an information architect may use markup at the top of an XML instance to identify a particular type of element or context in which the content should not be translated.

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

Basic Concepts of ITS

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).

ITS distinguishes two approaches to selection: local, and with global rules. The local approach puts ITS markup in the relevant element of the XML document (e.g. the author element in DocBook). The rule-based, global approach puts the ITS markup in elements defined by ITS itself (namely the <rules> element).

The mechanisms defined for ITS selection resemble those defined in CSS [bos98]. The local approach can be compared to the style attribute in CSS, and the approach with global rules is similar to the <style> element in CSS. In contrast to CSS, ITS uses XPath for identifying nodes.

With global selection, ITS markup appears in elements defined by ITS itself (for example the ITS `<translateRule>` element embedded within an ITS `<rules>` element. With local selection, ITS markup appears in elements of the host vocabulary, e.g. an ITS `translate` attribute at an XHTML <p> element.

Local Selection

Local selection may be used for example by content authors. They may use the ITS `translate` attribute to indicate what text should be translated and what text 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.

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

```
</article>
```

To allow for this usage of ITS, the schema developer will need to add the `translate` attribute to the schema as a common attribute or on all the relevant element definitions.

Global Selection

Global selection may for example by used by information architects. They may use the `<rules>` element for identifying non-translatable content in a general fashion.

The ITS `<rules>` element is part of the global, rule-based selection mechanism provided by ITS. It works as follows: A document can contain a `<rules>` element, which contains one or more data category specific ITS elements (for example `<translateRule>`). Each of these specific elements contains a `selector` attribute. As its name suggests, this attribute selects (or designates) the XML node or nodes to which a corresponding ITS information pertains. The values of the ITS selector `attribute` is an XPath absolute location path. Information about namespace bindings in these path expressions is contained in the ITS element <ns> which is a child of `<rules>`.

```
<topic id="myTopic" xml:lang="en-us"
 xmlns="myvocabulary.com">
 <title>Using ITS</title>
 <prolog>
  <its:rules xmlns:its="http://www.w3.org/2005/11/its">
   <its:ns prefix="my" ns="myvocabulary.com"/>
   <its:translateRule its:selector="//my:term"
its:translate="no"/>
  </its:rules>
 </prolog>
 <body>
  <p>An <term>ITS namespace</term> definition
exists...</p>
 </body>
</topic>
```

The ITS `selector` attribute allows:

- ITS data categories to be applied in global rules (even outside of an XML document or schema)

- ITS data categories to pertain to sets of XML nodes (for example all <p> elements in an XML document)
- ITS data categories to pertain to attributes
- ITS data categories to map to existing markup (for example to map the terminology data category to the term element in DITA)

The power of ITS selection mechanisms comes at a price: rules related to overriding/precedence, and inheritance have to be established. (see the section “Inheritance, Overriding/and Precedence”.)

For specification of the translate information, the contents of the <rules> element would normally be designed by an information architect familiar with the host vocabulary/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)

Adding Information and Pointing to Existing Information

As for global rules, depending on the data category and its usage, there may be attributes for adding information to the selected nodes, or for pointing to existing information in the document. For example, the data category “localization information” can be used for adding information to selected nodes, or for pointing to existing information in the document. For the former purpose, a <locInfo> element can be used. For the latter purpose, a locInfoPointer attribute can be used.

```
<its:rules>
 <its:locInfoRule its:locInfoType="alert"
its:selector="/body/p[1]">
 <its:locInfo>This p element has to be handled
carefully"</its:locInfo>
 </its:locInfoRule>
<its:locInfoRule its:locInfoType="alert"
 its:locInfoPointer="@locn-alert" its:selector="//*"/>
 <its:locInfoRule its:locInfoType="description"
 its:locInfoPointer="//@locn-note" its:selector="//*"/>
</its:locInfoRule>
</its:rules>
```

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 which express a closed set of values, that is: “translatability”, “directionality” and “elements within text”.

The functionalities of adding information and pointing to existing information are mutually exclusive. That is: attributes for pointing and adding must not appear at the same data category specific “rules” element.

Inheritance, Overriding/and Precedence

As mentioned above, the flexibility of combining local and global selection is based on mechanisms for overriding/precedence.

The ITS `translate` attribute may for example appear twice: at a `<translateRule>` element, and at a specific <p> element. Since the ITS `selector` attribute in the `<translateRule>` element may select all <p> elements, a question arises: What is the value for the “translatability” data category of the <p> element which has local markup? ITS provides precedence and inheritance rules which answer questions like this. In the example, the value is `"no"` (that is, the content of the <p> element should not be translated).

```
<text>
 <head>
 <its:rules>
  <its:translateRule its:translate="yes"
its:selector="//p"/>
 <its:rules>
 </head>
 <body>
  <p its:translate="no"> ...
<dl><dt>...</dt><dd>...</dd></dl></p>
 </body>
</text>
```

Overview of ITS data categories

This section provides a brief overview of the data categories which are defined in the current Internationalization Tag Set working draft. The overview is incomplete, e.g. not all variations of adding or pointing to information are exemplified.

Table 1 describes the position of data categories (global and / or local), whether it is possible to add information or to point to existing information in an XML document, and the default selections of the data category.

Data category	**Position**	**Adding / Pointing**	**Default selection**
translatability	global or local	adding	text and child elements, no attributes
localization Information	global or local	adding or pointing	text and child elements, no attributes
terminology	global or local	adding or pointing	text and child elements, no attributes

directionality	global or local	adding	text and child elements, including attributes
Ruby	global or local	adding or pointing	text and child elements, no attributes
language Information	global	pointing	text and child elements, including attributes
elements within text	global	pointing	text and child elements, no attributes

Table 1. Overview: ITS data categories

Translatability

The data category translatability 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).

As for global rules, translatability is expressed with a `<translateRule>` element with a `translate` attribute.

```
<its:rules>

 <its:translateRule its:translate="yes"
its:selector="//p"/>

<!-- All p elements should be translated-->

</its:rules>
```

Locally, translatability is expressed with a `translate` attribute. The default selection is the textual content of the element, including child elements, but excluding attributes.

Localization Information

The data category “localization information” is used to communicate information to localizers about a particular item of content. Two types of informative notes are needed: An alert contains information that the translator must read before translating a piece of text. A description provides useful background information that the translator will refer to only if they wish.

Locally, localization information can be expressed with a `locInfo` attribute and a `locInfoType` attribute. The latter contains the values `"alert"` or `"description"`. As for global rules, adding localization information to selected nodes is realized with a `<locInfoRule>` element with a `locInfoType` attribute and a `<locInfo>` child element.

```
<its:rules>
 <its:locInfoRule its:locInfoType="alert"
its:selector="/body/p[1]">
  <its:locInfo>This p element has to be handled
carefully"</its:locInfo>
 </its:locInfoRule>
</its:rules>
```

The functionality of pointing to existing localization information is realized via a `<locInfoRule>` element with a `locInfoPointer` attribute. An example was given in the section “Adding Information and Pointing to Existing Information”.

Terminology

The terminology data category is used to mark terms, to increase consistency across different parts of e.g. technical documentations.

Terminology can be expressed locally or globally. In addition to identifying terms, a reference to external term data bases may be added. For this purpose, the following `<termRule>` element contains a `termRef` attribute which expresses the data base reference.

```
<its:rules>
 <its:termRule its:selector="/body/p[1]/span"
its:termRef="http://example.com/termdatabase/#x142539"/>
</its:rules>
```

Directionality

This data category expresses the directionality of a piece of text. Its values are `"ltr"`, `"rtl"`, `"lro"` or `"rlo"`. Markup for directionality is important for scripts like Hebrew or Arabic which are written from right to left. The ITS definition for directionality is compliant with the dir attribute in XHTML 2.0 *[axel05]*.

Identical to the XHTML dir attribute, directionality can be expressed locally via an ITS `dir` attribute. Globally, directionality is expressed via a `<dirRule>` element.

```
<its:rules>
 <its:dirRule its:dir="rtl"
its:selector="/body/p[1]/quote[xml:lang='he']"/>
<!-- Some Hebrew quotation -->
</its:rules>
```

Ruby

The data category ruby 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.

Locally in a document, Ruby is realized with a `<ruby>` element. It contains a `<rb>` element with the ruby base text, and a `<rt>` element with the ruby text.

```
<text>
 <head> ... </head>
 <body>
  <p>This is about the
    <its:ruby>
     <its:rb>W3C</its:rubyBase>
     <its:rt>World Wide Web Consortium</its:rubyText>
    </its:ruby>.
  </p>
 </body>
</text>
```

To add ruby text to attribute values, a `<rubyRule>` element with a `rubyText` attribute can be used.

```
<text ...>
 <head> ... </head>
```

```
  <its:rules>
   <its:rubyRule its:rubyText="World Wide Web Consortium"
    its:selector="/body/img[1]/@alt"/>
  </its:rules>
 <body>
  <img src="w3c_home.png" alt="W3C"/> ...
 </body>
</text>
```

Language Information

The element <langRule> is used to express that a given piece of content (selected by the attribute langPointer) is used to express language information as defined by RFC 3066bis *[phil05]*. The purpose of <langRule> is to point to markup which value is an RFC3066bis compliant language tag. Examples are the lang attribute in HTML and xml:lang. The following <langRule> element points to the HTML lang attribute (given the appropriate namespace binding).

```
<its:langRule its:selector="//p"
its:langPointer="@h:lang"/>
```

Elements Within Text

The data category “elements within text” expresses information about whether an element is part of its parent text unit. This information is important e.g. for automatic segmentation processes. The values of the data category are "yes" (the element and its immediate child text nodes are part of the text unit of its parent element) or "no" (the element is not within text or holds an independent text unit within a parent text unit). Elements not listed are considered to be not within text.

“elements within text” can be only expressed globally, using a <withinTextRule> element:

```
<its:rules>
 <its:withinTextRule its:withinText="yes"
its:selector="//b | //em | //i"/>
</its:rules>
```

ITS Specification development using ODD

The ITS specification has been developed using an XML vocabulary known as ODD (*One Document Does it all*), an application of the Text Encoding Initiative (TEI) Guidelines (TEI, *[bur05]*). This is effectively a literate programming language for production and documentation of any XML schema, with four important characteristics:

1. The element and attribute sets making up the schema are formally specified using a special XML vocabulary
2. The specification language also includes support for macros (like DTD entities, or schema patterns), a hierarchical class system for attributes and elements, and the creation of pre-defined groups of elements known as modules.
3. Content models for elements and attributes are written using an embedded RELAXNG XML notation, but tools are available to generate schemas in any of RELAXNG, DTD language, or W3C schema.
4. Documentation describing the supported elements, attributes, value lists etc is managed along with their specification, together with use cases, examples, and other supporting material.

Some W3C working groups use the traditional model of writing DTDs or Schemas by hand, with associated documentation in a variant of HTML. This has a number of drawbacks. Maintaining consistency in documentation and specification of a large or complex schema, particularly one which is multiply authored, is a non-trivial problem, analogous to the problem of maintaining large software development projects. The ODD language was found to have several advantages. It allows for the automated production of outputs in different schema languages, as noted above. Formalising the process of documentation ensures and enforces good practice, and allows for automatic production of completely documented schemas and consistently detailed support documentation. The process is familiar, of course, to adherents of Don Knuth's *Literate Programming* ideas.

The ODD language

The ODD language is defined as one of the 22 modules which make up the TEI

Guidelines (TEI P5). These Guidelines are themselves written using this language, and generated as an output from that source. The module concerned adds a number of specialist elements to the existing range of TEI markup, which supports a broad range of documentation-style elements comparable to those provided by other XML schems such as DocBook or XHTML. The most important additions for these purposes are as follows:

1. `<schemaSpec>`, which sets up the definition of a new schema
2. `<elementSpec>`, which defines a new element
3. `<classSpec>`, which defines a new class
4. `<attList>`, which defines a set of attributes for an element
5. `<attDef>`, which defines an attribute

Each definition of a new primary object (elements and attribute) has associated description and examples. A complete example of a definition is as follows:

```
<elementSpec xmlns="http://www.tei-c.org/ns/1.0"
   module="namesdates"
   ident="district">
<desc>contains the name of any kind of subdivision of a
settlement,  such as a parish, ward, or other
administrative or geographic unit.</desc>
  <classes>
    <memberOf key="model.placeNamePart"/>
    <memberOf key="att.naming"/>
    <memberOf key="att.typed"/>
  </classes>
  <content>
    <rng:zeroOrMore
xmlns:rng="http://relaxng.org/ns/structure/1.0">
     <rng:choice>
      <rng:text/>
      <rng:ref name="model.gLike" />
      <rng:ref name="model.phrase"/>
      <rng:ref name="model.global"/>
      </rng:choice>
    </rng:zeroOrMore>
  </content>
  <exemplum>
    <egXML xmlns="http://www.tei-c.org/ns/Examples">
      <placeName>
        <district type="ward">Jericho</district>
        <settlement>Oxford</settlement>
      </placeName>
    </egXML>
  </exemplum>
  <exemplum>
    <egXML xmlns="http://www.tei-c.org/ns/Examples">
      <placeName>
        <district type="area">South Side</district>
        <settlement>Chicago</settlement>
      </placeName>
    </egXML>
  </exemplum>
</elementSpec>
```

The important things to note here are:

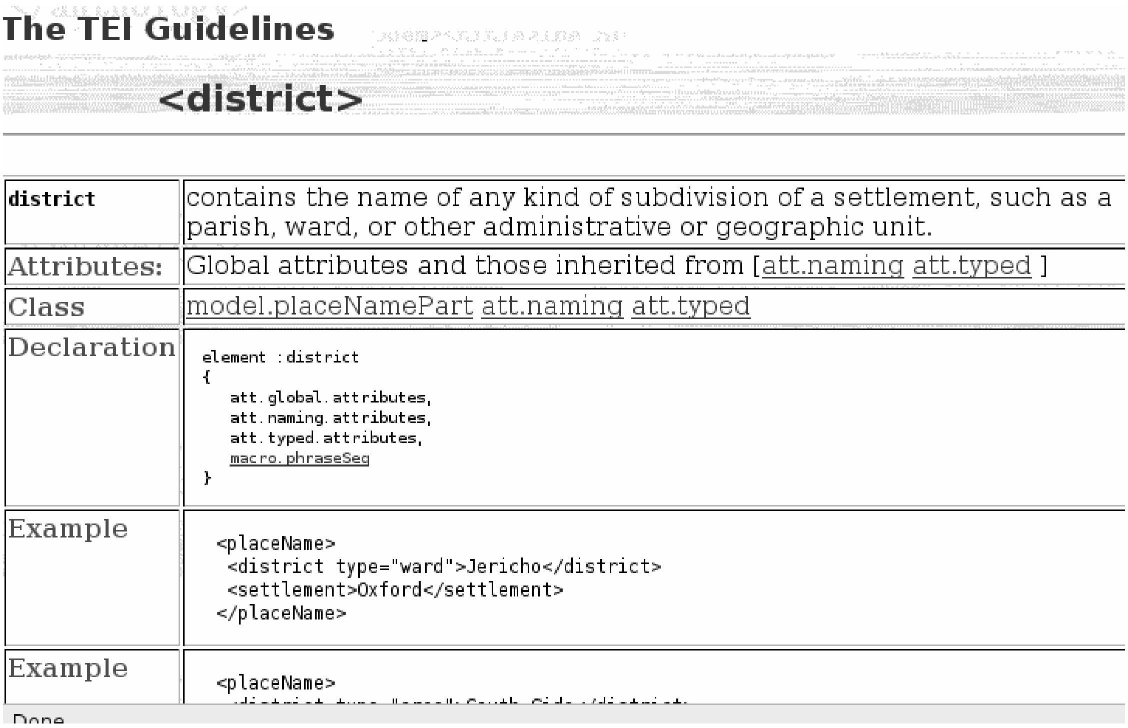
1. The `<memberOf>` elements establishing which class this element belongs to. Classes either establish the groups to which an element belongs, or a link to a set of attributes which will be added to this element.
2. The content model expressed in RELAXNG, which references other elements only by the names of classes to which they belong.
3. The worked examples, embedded in their own namespace.

This specification may be processed to produce a DTD, a RELAXNG schema, an XSD schema, or documentation in various forms (including internationalized and localized, where the appropriate translation work has been undertaken), some of which are shown below:

```
<!--doc:contains the name of any kind of subdivision of a
settlement,  such as a parish, ward, or other
```

```
administrative or geographic unit. -->
<!ELEMENT %n.district;
  #PCDATA | %model.gLike; | %model.phrase;
| %model.global;)*>
<!ATTLIST %n.district;
 %att.global.attributes;
 %att.naming.attributes;
 %att.typed.attributes; >
```

```
district =
  ## contains the name of any kind of subdivision of a
  ## settlement,  such as a parish, ward, or other
  ## administrative or geographic unit.
  element district { district.content,
district.attributes }
district.content = (text | model.gLike | model.phrase |
model.global)*
district.attributes =
  att.global.attributes,
  att.naming.attributes,
  att.typed.attributes,
  empty
model.placeNamePart |= district
```



The TEI Guidelines

<district>

district	contains the name of any kind of subdivision of a settlement, such as a parish, ward, or other administrative or geographic unit.
Attributes:	Global attributes and those inherited from [att.naming att.typed]
Class	model.placeNamePart att.naming att.typed

Declaration

```
element :district
{
   att.global.attributes,
   att.naming.attributes,
   att.typed.attributes,
   macro.phraseSeg
}
```

Example

```
<placeName>
 <district type="ward">Jericho</district>
 <settlement>Oxford</settlement>
</placeName>
```

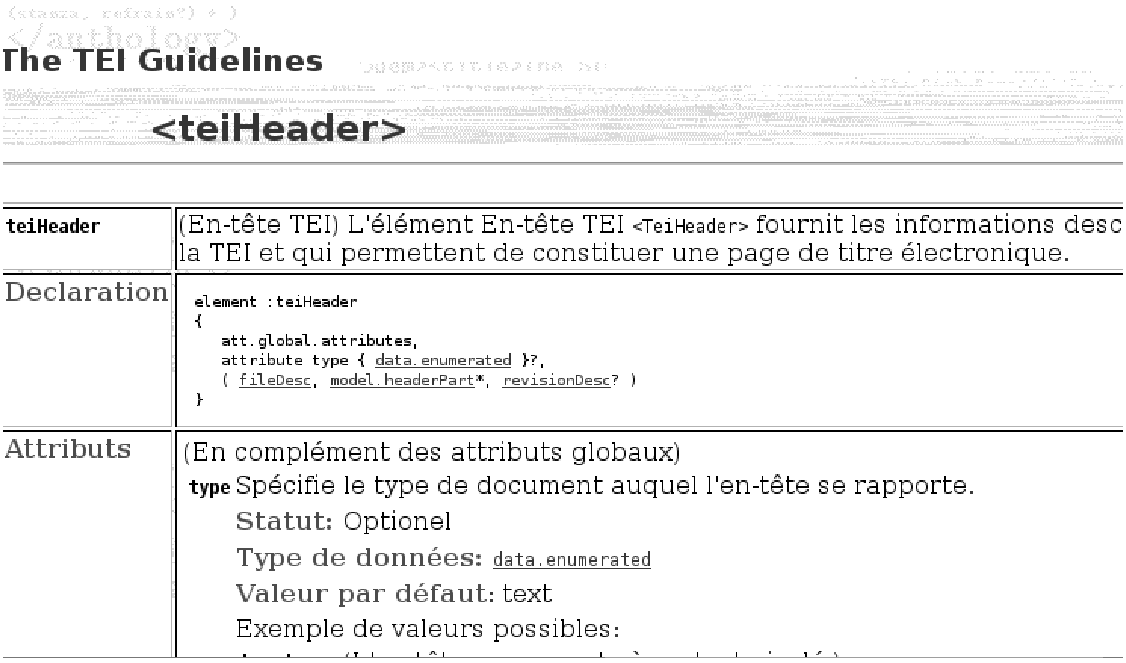
Example

```
<placeName>
 <district type="area">South Side</district>
```

Done

its01.jpg

Figure 1 Normal documentation layout



The TEI Guidelines

<teiHeader>

teiHeader	(En-tête TEI) L'élément En-tête TEI <TeiHeader> fournit les informations desc la TEI et qui permettent de constituer une page de titre électronique.

Declaration

```
element :teiHeader
{
   att.global.attributes,
   attribute type { data.enumerated }?,
   ( fileDesc, model.headerPart*, revisionDesc? )
}
```

Attributs

(En complément des attributs globaux)

type Spécifie le type de document auquel l'en-tête se rapporte.

Statut: Optionel

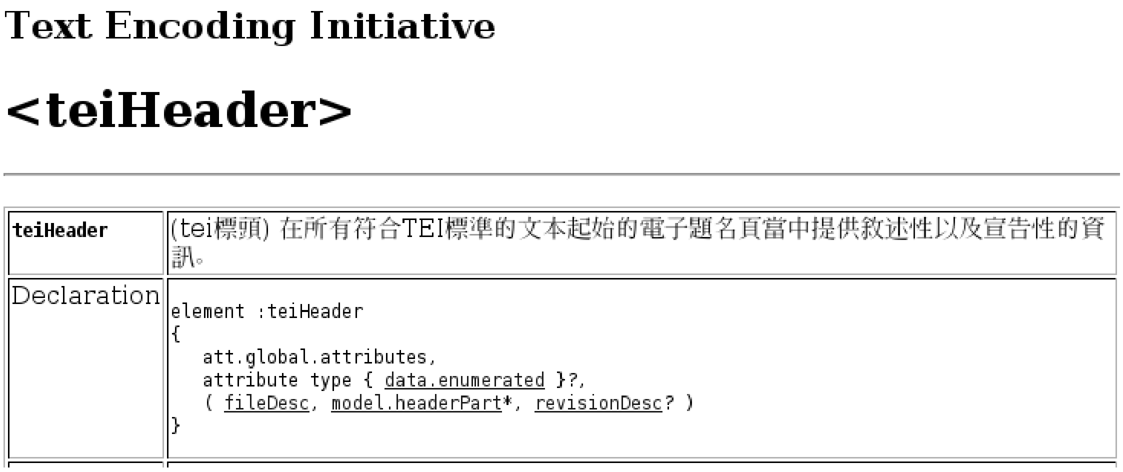
Type de données: data.enumerated

Valeur par défaut: text

Exemple de valeurs possibles:

its02.jpg

Figure 2 Documentation translated to French



Text Encoding Initiative

<teiHeader>

teiHeader	(tei標頭) 在所有符合TEI標準的文本起始的電子題名頁當中提供敘述性以及宣告性的資訊。

Declaration	element :teiHeader
	{
	att.global.attributes,
	attribute type { data.enumerated }?,
	(fileDesc, model.headerPart*, revisionDesc?)
	}

its03.jpg

Figure 3 Documentation translated to Chinese

Interestingly, ODD also has a placeholder (the `<equiv>` element) for associating objects with external ontologies, such as the CIDOC CRM (*[cidoc]*) for cultural information. Thus for the example above, we might link `<district>` to CIDOC's E48.

A detailed description of the ODD language is available at http://www.tei-c.org/release/doc/tei-p5-doc/html/TD.html

ODD in ITS

For the purposes of writing a W3C specification, a customization of the TEI was created which included the module for ODD markup and removed many elements only of interest in describing existing texts. This allows for a streamlined authoring environment, in which the editor is only prompted for elements which are relevant. To make matters more complicated, however, it was decided to retain the metadata header using the W3C *xmlspec* markup, so a further modification replaces the `<teiHeader>` with a `<header xmlns="http://example.com/xmlspec">` (using the techniques described in *[rahtz04]*).

The main body of the ITS specification is written using normal TEI markup, and the element definition fragments are standard:

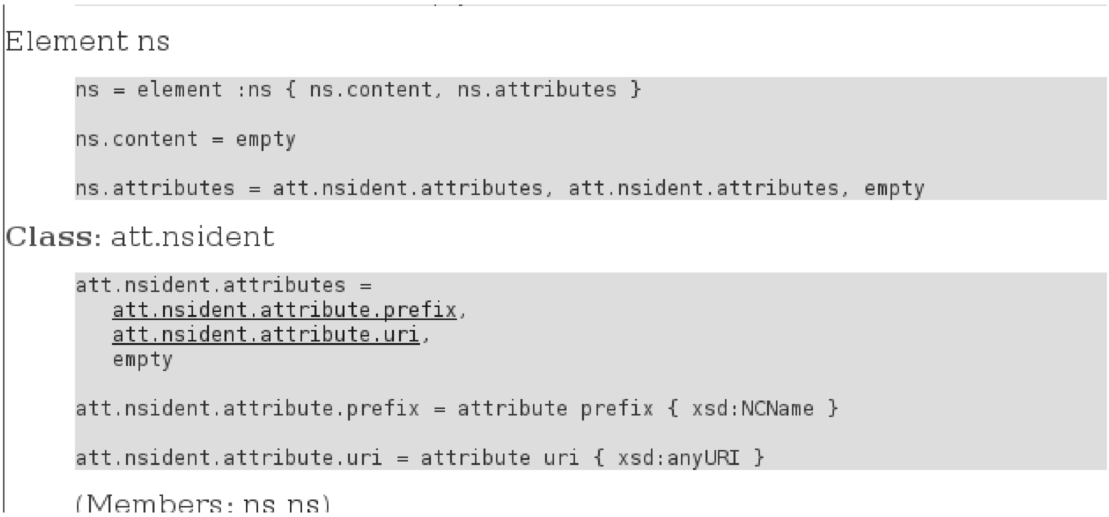
```
<elementSpec ident="rules"
 ns="http://www.w3.org/2005/11/its">
 <classes>
```

```
  <memberOf key="att.xlink"/>
 </classes>
 <content> ...
  <rng:choice>
   <rng:ref name="translateRule"/>
   <rng:ref name="locInfoRule"/>
   <rng:ref name="termRule"/>
   <rng:ref name="dirRule"/> ...
   </rng:choice>
  </rng:oneOrMore>
 </content> ...
</elementSpec>
```

Transforming ODD

XSLT transforms are provided by the TEI to extract documentation in HTML, XSL FO or LaTeX forms, and to generate RELAXNG documents and DTD. From the RELAXNG documents, James Clark's trang is used to create XML Schema documents. For the ITS specification, this allows a useful set of non-normative fragments to be generated for inclusion into host schemas.

For documentation purposes, the W3C ITS working group preference is for a BNF-like syntax. This is contrasted in the following alternative renditions of the formal documentation for <ns>.



Element ns

```
ns = element :ns { ns.content, ns.attributes }

ns.content = empty

ns.attributes = att.nsident.attributes, att.nsident.attributes, empty
```

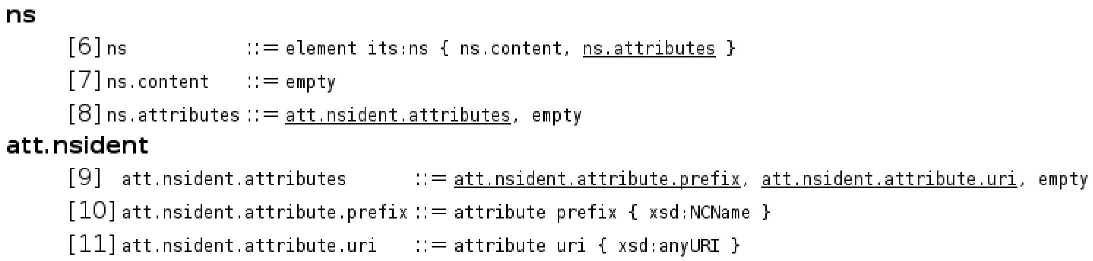
Class: att.nsident

```
att.nsident.attributes =
   att.nsident.attribute.prefix,
   att.nsident.attribute.uri,
   empty

att.nsident.attribute.prefix = attribute prefix { xsd:NCName }

att.nsident.attribute.uri = attribute uri { xsd:anyURI }
```

(Members: ns ns)

its04.jpg

Figure 4 Conventional ODD representation of element definition



```
ns
    [6] ns              ::= element its:ns { ns.content, ns.attributes }
    [7] ns.content      ::= empty
    [8] ns.attributes   ::= att.nsident.attributes, empty
att.nsident
    [9]  att.nsident.attributes       ::= att.nsident.attribute.prefix, att.nsident.attribute.uri, empty
    [10] att.nsident.attribute.prefix ::= attribute prefix { xsd:NCName }
    [11] att.nsident.attribute.uri    ::= attribute uri { xsd:anyURI }
```

its05.jpg

Figure 5 Representation of element definition preferred by W3C ITS working group

The W3C display is achieved by a variant of the transformation used to turn RELAXNG XML format into RELAXNG compact format. This is based on the XSLT transform of David Rosenborg (Pantor Engineering), and is used extensively in the TEI to show the more readable syntax for element content models. It turns the source

```
<elementSpec ident="ns">
  <classes>
    <memberOf key="att.nsident"/>
  </classes>
  <content>
    <rng:empty/>
  </content>
</elementSpec>
```

into W3C xmlspec markup of

```
<prod id="ns">
  <lhs>ns</lhs>
  <rhs> element :ns { ns.content,
     <nt def="ns.attributes">ns.attributes</nt> }
  </rhs>
</prod>
```

Bibliography

[axel05] Jonny Axelsson, Mark Birbeck and others. XHTML™ 2.0. W3C Working Draft 27 May 2005. Available at http://www.w3.org/TR/2005/WD-xhtml2-20050527.

[bos98] Bert Bos, Håkon Wium Lie, Chris Lilley, Ian Jacobs. Cascading Style Sheets, level 2, CSS2 Specification. W3C Recommendation 12-May-1998. Available at http://www.w3.org/TR/1998/REC-CSS2-19980512.

[bur06] Lou Burnard and Syd Bauman (eds). Text Encoding Initiative Guidelines development version (P5) . TEI Consortium, Charlottesville, Virginia, USA, Text Encoding Initiative.

[cidoc] The CIDOC Conceptual Reference Model . Draft International Standard ISO/DIS 21127.

[ishi06] Richard Ishida, Susan K. Miller. Localization vs. Internationalization. Article, W3C Internationalization Activity. Available at http://www.w3.org/International/questions/qa-i18n.

[lies06] Christian Lieske, Felix Sasaki (eds.). Internationalization Tag Set (ITS). W3C Working Draft 14 April 2006. Available at http://www.w3.org/TR/2006/WD-its-20060414/.

[phl05] Addison Phillips and Mark Davis. Tags for Identifying Languages. IETF Internet-Draft, October 2005. Available at http://www.ietf.org/internet-drafts/draft-ietf-ltru-registry-14.txt.

[rahtz04] Sebastian Rahtz, Norman Walsh and Lou Burnard. A unified model for text markup: TEI, Docbook, and beyond, paper presented at XML Europe 2004, Amsterdam, May 2004.

[sav05] Yves Savourel (ed.) . Internationalization and Localization Markup Requirements. W3C Working Draft 22 November 2005. Available at http://www.w3.org/TR/2005/WD-itsreq-20051122/.