D2.4: LT-WEB METADATA STANDARD

David Filip (UL), Shaun McCance (external contributor), Dave Lewis (TCD), Christian Lieske (external contributor / SAP AG), Arle Lommel (DFKI), Jirka Kosek (UEP), Felix Sasaki (DFKI / W3C Fellow), Yves Savourel (ENLASO)

Distribution: Public

MultilingualWeb-LT (LT-Web)
Language Technology in the Web

FP7-ICT-2011-7

Project no: 287815

Document Information

Deliverable number:	2.4
Deliverable title:	LT-Web Metadata Standard
Dissemination level:	PU
Contractual date of delivery:	30th September 2013
Actual date of delivery:	29th October 2013
Author(s):	**David Filip, Shaun McCance, Dave Lewis, Christian Lieske, Arle Lommel, Jirka Kosek, Felix Sasaki, Yves Savourel**
Participants:	W3C MLW-LT Working Group
Internal Reviewer:	W3C MLW-LT Working Group
Workpackage:	WP2
Task Responsible:	Dave Lewis
Workpackage Leader:	Dave Lewis

Revision History

Revision	Date	Author	Organization	Description
1	29/10/2013	See authors list above	See authors list above	Final versions; for drafts, see D2.2 (LT-Web Metadata Draft Documents)

The deliverable has been published as an HTML document at

http://www.w3.org/TR/2013/REC-its20-20131029/

The main content of this PDF file has been generated from the HTML.

Internationalization Tag Set (ITS) Version 2.0

W3C Recommendation 29 October 2013

This version:
http://www.w3.org/TR/2013/REC-its20-20131029/
Latest version:
http://www.w3.org/TR/its20/
Previous version:
http://www.w3.org/TR/2013/PR-its20-20130924/

Editors:
David Filip, University of Limerick
Shaun McCance, Invited Expert
Dave Lewis, TCD
Christian Lieske, SAP AG
Arle Lommel, DFKI
Jirka Kosek, UEP
Felix Sasaki, DFKI / W3C Fellow
Yves Savourel, ENLASO

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 previous publication 2013-09-24.

Copyright © 2013 W3C® (MIT, ERCIM, Keio, Beihang), All Rights Reserved. W3C liability, trademark and document use rules apply.

Abstract

The technology described in this document “*Internationalization Tag Set (ITS) 2.0*“ enhances the foundation to integrate automated processing of human language into core Web technologies. ITS 2.0 bears many commonalities with its predecessor, ITS 1.0 but provides additional concepts that are designed to foster the automated creation and processing of multilingual Web content. ITS 2.0 focuses on HTML, XML-based formats in general, and can leverage processing based on the XML Localization Interchange File Format (XLIFF), as well as the Natural Language Processing Interchange Format (NIF).

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/.

The technology described in this document “*Internationalization Tag Set (ITS) 2.0*“ enhances the foundation to integrate automated processing of human language into core Web technologies. ITS 2.0 bears many commonalities with is predecessor, ITS 1.0 but provides additional concepts that are designed to foster the automated creation and processing of multilingual Web content. ITS 2.0 focuses on HTML, XML-based formats in general, and can leverage processing based on the XML Localization Interchange File Format (XLIFF), as well as the Natural Language Processing Interchange Format (NIF).

This document was published by the MultilingualWeb-LT Working Group as a W3C Recommendation (see W3C document maturity levels). The Working Group has completed and approved this specification's Test Suite and created an Implementation Report that shows that two or more independent implementations pass each test.

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.

The ITS 2.0 specification has a normative dependency on the HTML5 specification: it relies on the HTML5 Translate attribute. By publishing this Recommendation, W3C expects that the functionality specified in this ITS 2.0 Recommendation will not be affected by changes to HTML5 as that specification proceeds to Recommendation.

If you wish to make comments, please send them to public-i18n-its-ig@w3.org. The archives for this list are publicly available. See also issues discussed within the MultilingualWeb-LT Working Group and the list of changes since the previous publication.

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

1 Introduction
  1.1 Overview
  1.2 General motivation for going beyond ITS 1.0
  1.3 Usage Scenarios
  1.4 High-level differences between ITS 1.0 and ITS 2.0
  1.5 Extended implementation hints
2 Basic Concepts
  2.1 Data Categories
  2.2 Selection
    2.2.1 Local Approach
    2.2.2 Global Approach
  2.3 Overriding, Inheritance and Defaults
  2.4 Adding Information or Pointing to Existing Information
  2.5 Specific HTML support
    2.5.1 Global approach in HTML5
    2.5.2 Local approach
    2.5.3 HTML markup with ITS 2.0 counterparts
    2.5.4 Standoff markup in HTML5
    2.5.5 Version of HTML

- 2.6 Traceability
- 2.7 Mapping and conversion
  - 2.7.1 ITS and RDF/NIF
  - 2.7.2 ITS and XLIFF
- 2.8 ITS 2.0 Implementations and Conformance

3 Notation and Terminology
- 3.1 Notation
- 3.2 Data category
- 3.3 Selection
- 3.4 ITS Local Attributes
- 3.5 Rule Elements
- 3.6 Usage of Internationalized Resource Identifiers in ITS
- 3.7 The Term HTML
- 3.8 The Term CSS Selectors

4 Conformance
- 4.1 Conformance Type 1: ITS Markup Declarations
- 4.2 Conformance Type 2: The Processing Expectations for ITS Markup
- 4.3 Conformance Type 3: Processing Expectations for ITS Markup in HTML
- 4.4 Conformance Type 4: Markup conformance for HTML5+ITS documents

5 Processing of ITS information
- 5.1 Indicating the Version of ITS
- 5.2 Locations of Data Categories
  - 5.2.1 Global, Rule-based Selection
  - 5.2.2 Local Selection in an XML Document
- 5.3 Query Language of Selectors
  - 5.3.1 Choosing Query Language
  - 5.3.2 XPath 1.0
  - 5.3.3 CSS Selectors
  - 5.3.4 Additional query languages
  - 5.3.5 Variables in selectors
- 5.4 Link to External Rules
- 5.5 Precedence between Selections
- 5.6 Associating ITS Data Categories with Existing Markup
- 5.7 ITS Tools Annotation

6 Using ITS Markup in HTML
- 6.1 Mapping of Local Data Categories to HTML
- 6.2 Global rules
- 6.3 Standoff Markup in HTML
- 6.4 Precedence between Selections

7 Using ITS Markup in XHTML

8 Description of Data Categories
- 8.1 Position, Defaults, Inheritance, and Overriding of Data Categories
- 8.2 Translate
  - 8.2.1 Definition
  - 8.2.2 Implementation
- 8.3 Localization Note
  - 8.3.1 Definition
  - 8.3.2 Implementation
- 8.4 Terminology
  - 8.4.1 Definition
  - 8.4.2 Implementation
- 8.5 Directionality
  - 8.5.1 Definition
  - 8.5.2 Implementation

8.6 Language Information
8.6.1 Definition
8.6.2 Implementation
8.7 Elements Within Text
8.7.1 Definition
8.7.2 Implementation
8.8 Domain
8.8.1 Definition
8.8.2 Implementation
8.9 Text Analysis
8.9.1 Definition
8.9.2 Implementation
8.10 Locale Filter
8.10.1 Definition
8.10.2 Implementation
8.11 Provenance
8.11.1 Definition
8.11.2 Implementation
8.12 External Resource
8.12.1 Definition
8.12.2 Implementation
8.13 Target Pointer
8.13.1 Definition
8.13.2 Implementation
8.14 ID Value
8.14.1 Definition
8.14.2 Implementation
8.15 Preserve Space
8.15.1 Definition
8.15.2 Implementation
8.16 Localization Quality Issue
8.16.1 Definition
8.16.2 Implementation
8.17 Localization Quality Rating
8.17.1 Definition
8.17.2 Implementation
8.18 MT Confidence
8.18.1 Definition
8.18.2 Implementation
8.19 Allowed Characters
8.19.1 Definition
8.19.2 Implementation
8.20 Storage Size
8.20.1 Definition
8.20.2 Implementation

Appendices

A References
B Internationalization Tag Set (ITS) MIME Type
C Values for the Localization Quality Issue Type
D Schemas for ITS
E Informative References
F Conversion to NIF

G Conversion NIF2ITS
H Localization Quality Guidance
I List of ITS 2.0 Global Elements and Local Attributes
J Revision Log
K Acknowledgements

1 Introduction

This section is informative.

1.1 Overview

Content or software that is authored in one language (so-called source language) for one locale (e.g. the French-speaking part of Canada) is often made available in additional languages or adapted with regard to other cultural aspects. A prevailing paradigm for multilingual production in many cases encompasses three phases: internationalization, translation, and localization (see the W3C's Internationalization Q&A for more information related to these concepts).

From the viewpoints of feasibility, cost, and efficiency, it is important that the original material is suitable for downstream phases such as translation. This is achieved by appropriate design and development. The corresponding phase is referred to as internationalization. A proprietary XML vocabulary may be internationalized by defining special markup to specify directionality in mixed direction text.

During the translation phase, the meaning of a source language text is analyzed, and a target language text that is equivalent in meaning is determined. For example national or international laws may regulate linguistic dimensions like mandatory terminology or standard phrases in order to promote or ensure a translation's fidelity.

Although an agreed-upon definition of the localization phase is missing, this phase is usually seen as encompassing activities such as creating locale-specific content (e.g. adding a link for a country-specific reseller), or modifying functionality (e.g. to establish a fit with country-specific regulations for financial reporting). Sometimes, the insertion of special markup to support a local language or script is also subsumed under the localization phase. For example, people authoring in languages such as Arabic, Hebrew, Persian or Urdu need special markup to specify directionality in mixed direction text.

The technology described in this document – the *Internationalization Tag Set (ITS) 2.0* addresses some of the challenges and opportunities related to internationalization, translation, and localization. ITS 2.0 in particular contributes to concepts in the realm of metadata for internationalization, translation, and localization related to core Web technologies such as XML. ITS does for example assist in production scenarios, in which parts of an XML-based document are to be excluded from translation. ITS 2.0 bears many commonalities with its predecessor, ITS 1.0 but provides additional concepts that are designed to foster enhanced automated processing – e.g. based on language technology such as entity recognition – related to multilingual Web content.

Like ITS 1.0, ITS 2.0 both identifies concepts (such as “Translate”), and defines implementations of these concepts (termed “ITS data categories”) as a set of elements and attributes called the *Internationalization Tag Set (ITS)*. The definitions of ITS elements and attributes are provided in the form of RELAX NG [RELAX NG] (normative). Since one major step from ITS 1.0 to ITS 2.0 relates to coverage for HTML, ITS 2.0 also establishes a relationship between ITS markup and the various HTML flavors. Furthermore, ITS 2.0 suggests when and how to leverage processing based on the XML Localization Interchange File Format ([XLIFF 1.2] and [XLIFF 2.0]), as well as the Natural Language Processing Interchange Format [NIF].

For the purpose of an introductory illustration, here is a series of examples related to the question, how ITS can indicate that certain parts of a document are not intended for translation.

Example 1: Document in which some content has to be left untranslated

In this document it is difficult to distinguish between those `string` elements that are intended for translation and those that are not to be translated. Explicit metadata is needed to 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: examples/xml/EX-motivation-its-1.xml]

ITS proposes several mechanisms, which differ among others in terms of the usage scenario/user types for which the mechanism is most suitable.

Example 2: Document that uses two different ITS mechanisms to indicate that some parts have to be left untranslated.

ITS provides two mechanisms to explicitly associate metadata with one or more pieces of content (e.g. XML nodes): a global, rule-based approach as well as a local, attribute-based approached. Here, for instance, a `translateRule` first specifies that only every second element inside `keyvalue_pairs` is intended for translation; later, an ITS `translate` attribute specifies that one of these elements is not to be translated.

```
<resources xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <its:rules version="2.0">
    <its:translateRule selector="//arguments" translate="no"/>
    <its:translateRule selector="//keyvalue_pairs/string[(position() mod 2)=1]" t
  </its:rules>
  <section id="Homepage">
    <arguments>
      <string>page</string>
      <string>childlist</string>
    </arguments>
    <variables>
```

```
      <string its:translate="no">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 its:translate='no'>NavajoWhite</string>
      <string>title</string>
      <string>List of Available Policies</string>
    </keyvalue_pairs>
  </section>
</resources>
```

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

1.2 General motivation for going beyond ITS 1.0

The basics of ITS 1.0 are simple:

1. Provide metadata (e.g. “Do not translate”) to assist internationalization-related processes
2. Use XPath (so-called global approach) to associate metadata with specific XML nodes (e.g. all elements named `uitext`) or put the metadata straight onto the XML nodes themselves (so-called local approach)
3. Work with a well-defined set of metadata categories or values (e.g. only the values "yes" and "no" for certain data categories)
4. Take advantage of existing metadata (e.g. terms already marked up with HTML markup such as `dt`)

This conciseness made real-world deployment of ITS 1.0 easy. The deployments helped to identify additional metadata categories for internationalization-related processes. The ITS Interest Group for example compiled a list of additional data categories (see this related summary). Some of these were then defined in ITS 2.0: ID Value, local Elements Within Text, Preserve Space, and Locale Filter. Others are still discussed as requirements for possible future versions of ITS:

1. “Context” = What specific related information might be helpful?
2. “Automated Language” = Does this content lend itself to automatic processing?

The real-world deployments also helped to understand that for the Open Web Platform – the ITS 1.0 restriction to XML was an obstacle for quite a number of environments. What was missing was, for example, the following:

1. Applicability of ITS to formats such as HTML in general, and HTML5 in particular
2. Easy use of ITS in various Web-exposed (multilingual) Natural Language Processing contexts
3. Computer-supported linguistic quality assurance
4. Content Management and translation platforms
5. Cross-language scenarios
6. Content enrichment
7. Support for W3C provenance [PROV-DM], “information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness”

8. Provisions for extended deployment in Semantic Web/Linked Open Data scenarios

ITS 2.0 was created by an alliance of stakeholders who are involved in content for global use. Thus, ITS 2.0 was developed with input from/with a view towards the following:

- Providers of content management and machine translation solutions who want to easily integrate for efficient content updates in multilingual production chains
- Language technology providers who want to automatically enrich content (e.g. via term candidate generation, entity recognition or disambiguation) in order to facilitate human translation
- Open standards endeavours (e.g. related to [XLIFF 1.2], [XLIFF 2.0] and [NIF]) that are interested for example in information sharing, and lossless roundtrip of metadata in localization workflows

One example outcome of the resulting synergies is the ITS Tool Annotation mechanism. It addresses the provenance-related requirement by allowing ITS processors to leave a trace: ITS processors can basically say “It is me that generated this bit of information”. Another example are the [NIF] related details of ITS 2.0, which provide a non-normative approach to couple Natural Language Processing with concepts of the Semantic Web.

1.3 Usage Scenarios

The [ITS 1.0] introduction states: “ITS is a technology to easily create XML, which is internationalized and can be localized effectively”. In order to make this tangible, ITS 1.0 provided examples for users and usages. Implicitly, these examples carried the information that ITS covers two areas: one that is related to the static dimension of mono-lingual content, and one that is related to the dynamic dimension of multilingual production.

- Static mono-lingual (for example, the area of content authors): This part of the content has the directionality “right-to-left”.
- Dynamic multilingual: (for example, the area of machine translation systems): This part of the content has to be left untranslated.

Although ITS 1.0 made no assumptions about possible phases in a multilingual production process chain, it was slanted towards a simple three phase “write→internationalize→translate” model. Even a birds-eye-view at ITS 2.0 shows that ITS 2.0 explicitly targets a much more comprehensive model for multilingual content production. The model comprises support for multilingual content production phases such as:

- Internationalization
- Pre-production (e.g. related to marking terminology)
- Automated content enrichment (e.g. automatic hyperlinking for entities)
- Extraction/filtering of translation-relevant content
- Segmentation
- Leveraging (e.g. of existing translation-related assets such as translation memories)
- Machine Translation (e.g. geared towards a specific domain)
- Quality assessment or control of source language or target language content
- Generation of translation kits (e.g. packages based on XLIFF)
- Post-production
- Publishing

The document [MLW US IMPL] lists a large variety of usage scenarios for ITS 2.0. Most of them are composed from the aforementioned phases.

W3C Recommendation

In a similar vein, ITS 2.0 takes a much more comprehensive view on the actors that may participate in a multilingual content production process. ITS 1.0 annotations (e.g. local markup for the Terminology data category) most of the time were conceived as being closely tied to human actors such as content authors or information architects. ITS 2.0 raises non-human actors such as word processors/editors, content management systems, machine translation systems, term candidate generators, entity identifiers/disambiguators to the same level. This change among others is reflected by the ITS 2.0 Tool Annotation, which allows systems to record that they have processed a certain part of content.

1.4 High-level differences between ITS 1.0 and ITS 2.0

The differences between ITS 1.0 and ITS 2.0 can be summarized as follows.

Coverage of [HTML5]: ITS 1.0 can be applied to XML content. ITS 2.0 extends the coverage to [HTML5]. Explanatory details about ITS 2.0 and [HTML5] are given in Section 2.5: Specific HTML support.

Addition of data categories: ITS 2.0 provides additional data categories and modifies existing ones. A summary of all ITS 2.0 data categories is given in Section 2.1: Data Categories.

Modification of data categories:

- ITS 1.0 provided the Ruby data category. ITS 2.0 does not provide ruby because at the time of writing the ruby model in HTML5 was still under development. Once these discussions are settled, the Ruby data category possibly will be reintroduced, in a subsequent version of ITS.
- The Directionality data category reflects directionality markup in [HTML 4.01]. The reason is that enhancements are being discussed in the context of HTML5 that are expected to change the approach to marking up directionality, in particular to support content whose directionality needs to be isolated from that of surrounding content. However, these enhancements are not finalized yet. They will be reflected in a future revision of ITS.

Additional or modified mechanisms: The following mechanisms from ITS 1.0 have been modified or added to ITS 2.0:

- ITS 1.0 used only XPath as the mechanism for selecting nodes in global rules. ITS 2.0 allows for choosing the query language of selectors. The default is XPath 1.0. An ITS 2.0 processor is free to support other selection mechanisms, like CSS selectors or other versions of XPath.
- In global rules it is now possible to set variables for the selectors (XPath expression). The `param` element serves this purpose.
- ITS 2.0 has an ITS Tools Annotation mechanism to associate processor information with the use of individual data categories. See Section 2.6: Traceability for details.

Mappings: ITS 2.0 provides a non-normative algorithm to convert ITS 2.0 information into [NIF] and links to guidance about how to relate ITS 2.0 to XLIFF. See Section 2.7: Mapping and conversion for details.

Changes to the conformance section: The Section 4: Conformance tells implementers how to implement ITS. For ITS 2.0, the conformance statements related to Ruby have been removed. For [HTML5], a dedicated conformance section has been created. Finally, a conformance clause related to Non-ITS elements and attributes has been added.

1.5 Extended implementation hints

As a general guidance, implementations of ITS 2.0 are encouraged to use a normalizing transcoder. It converts from a legacy encoding to a Unicode encoding form and ensures that the result is in Unicode Normalization Form C. Further information on the topic of Unicode normalization is provided in [Charmod Norm].

2 Basic Concepts

This section is informative.

The purpose of this section is to provide basic knowledge about how ITS 2.0 works. Detailed knowledge (including formal definitions) is given in the subsequent sections.

2.1 Data Categories

A key concept of ITS is the abstract notion of data categories. Data categories define the information that can be conveyed via ITS. An example is the Translate data category. It conveys information about translatability of content.

Section 8: Description of Data Categories defines data categories. It also describes their implementation, i.e. ways to use them for example in an XML context. The motivation for separating data category definitions from their implementation is to enable different implementations with the following characteristics:

- For various types of content (XML in general or HTML).
- For a single piece of content, e.g. a `p` element. This is the so-called local approach.
- For several pieces of content in one document or even a set of documents. This is the so-called global approach.
- For a complete markup vocabulary. This is done by adding ITS markup declarations to the schema for the vocabulary.

ITS 2.0 provides the following data categories:

- Translate: expresses information about whether a selected piece of content is intended for translation or not.
- Localization Note: communicates notes to localizers about a particular item of content.
- Terminology: marks terms and optionally associates them with information, such as definitions or references to a term data base.
- Directionality: specifies the base writing direction of blocks, embeddings and overrides for the Unicode bidirectional algorithm.
- Language Information: expresses the language of a given piece of content.
- Elements Within Text: expresses how content of an element is related to the text flow (constitutes its own segment like paragraphs, is part of a segment like emphasis marker etc.).
- Domain: identifies the topic or subject of the annotated content for translation-related applications.
- Text Analysis: annotates content with lexical or conceptual information (e.g. for the purpose of contextual disambiguation).
- Locale Filter: specifies that a piece of content is only applicable to certain locales.

- Provenance: communicates the identity of agents that have been involved processing content.
- External Resource: indicates reference points in a resource outside the document that need to be considered during localization or translation. Examples of such resources are external images and audio or video files.
- Target Pointer: associates the markup node of a given source content (i.e. the content to be translated) and the markup node of its corresponding target content (i.e. the source content translated into a given target language). This is relevant for formats that hold the same content in different languages inside a single document.
- Id Value: identifies a value that can be used as unique identifier for a given part of the content.
- Preserve Space: indicates how whitespace is to be handled in content.
- Localization Quality Issue: describes the nature and severity of an error detected during a language-oriented quality assurance (QA) process.
- Localization Quality Rating: expresses an overall measurement of the localization quality of a document or an item in a document.
- MT Confidence: indicates the confidence that MT systems provide about their translation.
- Allowed Characters: specifies the characters that are permitted in a given piece of content.
- Storage Size: specifies the maximum storage size of a given piece of content.

Most of the existing ITS 1.0 data categories are included and new ones have been added. Modifications of existing ITS 1.0 data categories are summarized in Section 1.4: High-level differences between ITS 1.0 and ITS 2.0.

2.2 Selection

Information (e.g. “translate this”) captured by an ITS data category always pertains to one or more XML or HTML nodes, primarily element and attribute nodes. In a sense, the relevant node(s) get “selected”. Selection may be explicit or implicit. ITS distinguishes two mechanisms for explicit selection: (1) local and (2) global (via `rules`). Both local and global approaches can interact with each other, and with additional ITS dimensions such as inheritance and defaults.

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 global approach is similar to the `style` element in HTML/XHTML:

- The local approach puts ITS markup in the relevant element of the host vocabulary (e.g. the `author` element in DocBook)
- The global rule-based approach puts the ITS markup in elements defined by ITS itself (namely the `rules` element)

ITS usually uses XPath in rules for identifying nodes although CSS Selectors and other query languages can in addition be implemented by applications.

ITS 2.0 can be used with XML documents (e.g. a DocBook article), HTML documents, document schemas (e.g. an XML Schema document for a proprietary document format), or data models in RDF.

The following two examples provide more details about the distinction between the local and global approach, using the Translate data category as an example.

2.2.1 Local Approach

The document in Example 3 shows how a content author can use the ITS translate attribute to indicate that all content inside the author element is not intended for translation (i.e. has to be left untranslated). Translation tools that are aware of the meaning of the attribute can protect the relevant content from being translated (possibly still allowing translators to see the protected content as context information).

Example 3: ITS markup on elements in an XML document (local approach)

```
<article xmlns="http://docbook.org/ns/docbook"
         xmlns:its="http://www.w3.org/2005/11/its"
         its:version="2.0" version="5.0" xml:lang="en">
  <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>
  <para>This is a short article.</para>
</article>
```

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

For the local approach (and Example 3) to work for a whole markup vocabulary, a schema developer would need to add the translate attribute to the schema as a common attribute or on all the relevant element definitions. The example indicates that inheritance plays a part in identifying which content does have to be translated and which does not: Although only the author element is marked as “do not translate”, its descendants (personname, firstname, surname) are considered to be implicitly marked as well. Tools that process this content for translation need to implement the expected inheritance.

For XML content, the local approach cannot be applied to a particular attribute. If ITS needs to be applied to a particular attribute, the global approach has to be used. The local approach applies to content of the current element and all its inherited nodes as described in Section 8.1: Position, Defaults, Inheritance, and Overriding of Data Categories. For the Translate data category used in [HTML5], this is different, see the explanation of the HTML5 definition of Translate.

2.2.2 Global Approach

The document in Example 4 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, e.g., in a “head” section, or even outside of the document itself). The rules element contains one or more ITS children/rule elements (for example translateRule). Each of these children elements contains a selector attribute. As its name suggests, this attribute selects the node or nodes to which the corresponding ITS information pertains. The values of ITS selector attributes are XPath absolute location paths (or CSS

Selectors if queryLanguage is set to "css"). Via the param element variables can be provided and used in selectors.

Information for the handling of namespaces in XPath expressions is taken from namespace declarations [XML Names] in the current rule element.

Example 4: ITS global markup in an XML document (rule-based approach)

```
<myTopic xmlns="http://mynsuri.example.com" id="topic01" xml:lang="en-us">
  <prolog>
    <title>Using ITS</title>
    <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
      <its:translateRule selector="//n:term" translate="no" xmlns:n="http://mynsu
    </its:rules>
  </prolog>
  <body>
    <p>ITS defines <term>data category</term> as an abstract concept for a partic
      information for internationalization and localization of XML schemas and dc
  </body>
</myTopic>
```

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

For the global approach (and Example 4) to work, a schema developer may need to add a rules element and associated markup to the schema. In some cases, global rules may be sufficient and other ITS markup (such as an translate attribute on the elements and attributes) may not be needed in the schema. However, it is likely that authors may need the local approach from time to time to override the general rule.

For specification of the Translate data category information, the contents of the translateRule 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 localization/translation.

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 nodes (for example all p elements in an XML instance)
- Changes can be made in a single location, rather than by searching and modifying local 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"

2.3 Overriding, Inheritance and Defaults

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 5 shows how inheritance and overriding work for the Translate data category:

- The ITS default is that all elements are translatable.
- 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"`.
- In the body of the document the default applies, and `its:translate="no"` is used to set "faux pas" as non-translatable.

Example 5: 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="2.0">
      <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> t
        dissertation on the origin of modern novel without mentioning the <tl>Epi
        Gilgamesh</tl>...</p>
    </div>
  </body>
</text>
```

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

For XML content, data category specific defaults are provided. These are independent of the actual XML markup vocabulary. Example for the Translate data category: `translate="yes"` for elements, and `translate="no"` for attributes.

For [HTML5], several HTML5 elements and attributes map exactly to ITS 2.0 data categories. Hence that HTML markup is normatively interpreted as ITS 2.0 data category information (see Section 2.5.3: HTML markup with ITS 2.0 counterparts for more information).

2.4 Adding Information or Pointing to Existing Information

Data categories can add information or point to information for the selected nodes. For example, the Localization Note data category can add information to selected nodes (using a `locNote` element), or point to existing information elsewhere in the document (using a `locNotePointer` attribute).

The data category overview table, in Section 8.1: Position, Defaults, Inheritance, and Overriding of Data Categories, provides an overview of which data categories allow the addition of information and which allow to point to existing information.

Adding information and pointing to existing information are *mutually exclusive*; attributes for adding information and attributes for pointing to the same information are not allowed to appear at the same rule element.

2.5 Specific HTML support

For applying ITS 2.0 data categories to HTML, five aspects are of importance:

1. Global approach in HTML5
2. Local Approach
3. HTML markup with ITS 2.0 counterparts
4. Standoff markup in HTML5
5. Version of HTML

In the following sections these aspects are briefly discussed.

2.5.1 Global approach in HTML5

To account for the so-called global approach in HTML, this specification (see Section 6.2: Global rules) defines:

- A link type for referring to external files with global rules from a `link` element.
- An approach to have inline global rules in the HTML `script` element.

It is preferable to use external global rules linked via the `link` element rather than to have inline global rules in the HTML document. The advantage is in being able to reuse the same rules file for many documents and also inline rules require secondary parsing of the `script` element.

Example 6: Using ITS global rules in HTML

The `link` element points to the rules file `EX-translateRule-html5-1.xml` The `rel` attribute identifies the ITS specific link relation `its-rules`.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=utf-8>
    <title>Translate flag global rules example</title>
    <link href=EX-translateRule-html5-1.xml rel=its-rules>
  </head>
  <body>
    <p>This sentence should be translated, but code names like
      the <code>span</code> element should not be translated.
      Of course there are always exceptions: certain
      code values should be translated, e.g. to a value
      in your language like <code translate=yes>warning</code>.</p>
  </body>
</html>
```

[Source file: examples/html5/EX-translate-html5-global-1.html]

Example 7: ITS rules file linked from HTML

The rules file linked in Example 6.

```
<its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its"
           xmlns:h="http://www.w3.org/1999/xhtml">
  <its:translateRule translate="no" selector="//h:code"/>
</its:rules>
```

[Source file: examples/html5/EX-translateRule-html5-1.xml]

Example 8: Using ITS inline global rules in HTML

The `script` element contains the same rules as the external rules file `EX-translateRule-html5-1.xml` in the above example.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=utf-8>
    <title>Translate flag global rules example</title>
    <script type=application/its+xml id=ru1>
      <its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its"
           xmlns:h="http://www.w3.org/1999/xhtml">
      <its:translateRule translate="no" selector="//h:code"/>
    </its:rules>
    </script>
  </head>
  <body>
    <p>This sentence should be translated, but code names like
      the <code>span</code> element should not be translated.
      Of course there are always exceptions: certain
      code values should be translated, e.g. to a value in
      your language like <code translate=yes>warning</code>.</p>
  </body>
</html>
```

[Source file: examples/html5/EX-translate-html5-inline-global-1.html]

2.5.2 Local approach

In HTML, an ITS 2.0 local data category is realized with the prefix `its-`. The general mapping of the XML based ITS 2.0 attributes to their HTML counterparts is defined in Section 6.1: Mapping of Local Data Categories to HTML. An informative table in Appendix I: List of ITS 2.0 Global Elements and Local Attributes provides an overview of the mapping for all data categories.

2.5.3 HTML markup with ITS 2.0 counterparts

There are four ITS 2.0 data categories, which have counterparts in HTML markup. In these cases, native HTML markup provides some information in terms of ITS 2.0 data categories. For these data categories, ITS 2.0 defines the following:

- The Language Information data category has the HTML `lang` attribute as a counterpart. In XHTML the counterpart is the `xml:lang` attribute. These HTML attributes act as local markup for the Language Information data category in HTML and take precedence over language information conveyed via a global `langRule`.

W3C Recommendation

- The Id Value data category has the HTML or XHTML `id` attribute as counterpart. This HTML attribute acts as local markup for the Id Value data category in HTML and takes precedence over identifier information conveyed via a global `idValueRule`.
- The Elements within Text data category has a set of HTML elements (the so-called phrasing content) as counterpart. In the absence of an Elements within Text local attribute or global rules selecting the element in question, most of the phrasing content elements are interpreted as `withinText="yes"` by default. The phrasing content elements `iframe`, `noscript`, `script` and `textarea` are interpreted as `withinText="nested"`.
- The Translate data category has a direct counterpart in [HTML5], namely the [HTML5] `translate` attribute. ITS 2.0 does not define its own behavior for [HTML5] `translate`, but just refers to the HTML5 definition. That definition also applies to nodes selected via global rules. That is, a `translateRule` like `<its:translateRule selector="//h:img" translate="yes"/>` will set the `img` element and its translatable attributes like `alt` to "yes".

Example 9: The Language Information, Id Value, Elements within Text and Translate ITS 2.0 data categories expressed by native HTML markup.

The `lang` attribute of the `html` element conveys the Language Information value "en". The `id` attribute of the `p` element conveys the Id Value "p1". The elements `em` and `img` are interpreted to be `withinText="yes"`. The `p` element and its children are set to be non-translatable via an [HTML5] `translate` attribute. Via inheritance, the `alt` attribute, normally translatable by default, also is non-translatable.

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
      <title>HTML native markup expressing three ITS 2.0 data categories</title>
  </head>
  <body>
    <p id="p1" translate="no">This is a <em>motherboard</em> and image:
      <img src="http://example.com/myimg.png" alt="My image"/>.</p>
  </body>
</html>
```

[Source file: examples/html5/EX-its-and-existing-HTML5-markup.html]

There are also some HTML markup elements that have or can have similar, but not necessarily identical, roles and behaviors as certain ITS 2.0 data categories. For example, the HTML `dfn` element could be used to identify a term in the sense of the Terminology data category. However, this is not always the case and it depends on the intentions of the HTML content author. To accommodate this situation, users of ITS 2.0 are encouraged to specify the semantics of existing HTML markup in an ITS 2.0 context with a dedicated global rules file. For example, a rule can be used to define that the HTML `dfn` has the semantics of ITS `term="yes"`. For additional examples, see the XML I18N Best Practices document.

2.5.4 Standoff markup in HTML5

The Provenance and the Localization Quality Issue data categories allow for using so-called standoff markup, see the XML Example 58. In HTML such standoff markup is placed into a `script` element. If this is done, the constraints for Provenance standoff markup in HTML and Localization quality issue markup in HTML need to be taken into account. Examples of standoff markup in HTML for the two data categories are Example 61 and Example 76.

2.5.5 Version of HTML

ITS 2.0 does not define how to use ITS in HTML versions prior to version 5. Users are thus encouraged to migrate their content to [HTML5] or XHTML. While it is possible to use `its-*` attributes introduced for [HTML5] in older versions of HTML (such as 3.2 or 4.01) and pages using these attributes will work without any problems, `its-*` attributes will be marked as invalid by validators.

2.6 Traceability

The ITS Tools Annotation mechanism allows processor information to be associated with individual data categories in a document, independently from data category annotations themselves (e.g. the Entity Type related to Text Analysis). The mechanism associates identifiers for tools with data categories via the `annotatorsRef` attribute (or annotators-ref in [HTML5]) and is mandatory for the MT Confidence data category. For the Terminology and Text Analysis data categories the ITS Tools Annotation is mandatory if the data categories provide confidence information. Nevertheless, ITS Tools Annotation can be used for all data categories. Example 23 demonstrates the usage in the context of several data categories.

2.7 Mapping and conversion

2.7.1 ITS and RDF/NIF

ITS 2.0 provides a non-normative algorithm to convert XML or HTML documents (or their DOM representations) that contain ITS metadata to the RDF format based on [NIF]. NIF is an RDF/OWL-based format that aims at interoperability between Natural Language Processing (NLP) tools, language resources and annotations.

The conversion from ITS 2.0 to NIF results in RDF triples. These triples represent the textual content of the original document as RDF typed information. The ITS annotation is represented as properties of content-related triples and relies on an ITS RDF vocabulary.

The back conversion from NIF to ITS 2.0 is defined informatively as well. One motivation for the back conversion is a roundtrip workflow like: 1) conversion to NIF 2) in NIF representation detection of named entities using NLP tools 3) back conversion to HTML and generation of Text Analysis markup. The outcome are HTML documents with linked information, see Example 52.

2.7.2 ITS and XLIFF

The XML Localization Interchange File Format [XLIFF 1.2] is an OASIS standard that enables translatable source text and its translation to be passed between different tools within localization and translation workflows. [XLIFF 2.0] is the successor of [XLIFF 1.2] and under development. XLIFF has been widely implemented in various translation management systems, computer aided translation tools and in utilities for extracting translatable content from source documents and merging back the content in the target language.

The mapping between ITS and XLIFF therefore underpins several important ITS 2.0 usage scenarios [MLW US IMPL]. These usage scenarios involve:

- the extraction of ITS metadata from a source language file into XLIFF
- the addition of ITS metadata into an XLIFF file by translation tools
- the mapping of ITS metadata in an XLIFF file into ITS metadata in the resulting target language files.

ITS 2.0 has no normative dependency on XLIFF, however a non-normative definition of how to represent ITS 2.0 data categories in XLIFF 1.2 or XLIFF 2.0 is being defined within the Internationalization Tag Set Interest Group.

2.8 ITS 2.0 Implementations and Conformance

What does it mean to implement ITS 2.0? This specification provides several conformance clauses as the normative answer (see Section 4: Conformance). The clauses target different types of implementers:

- Conformance clauses in Section 4.1: Conformance Type 1: ITS Markup Declarations tell markup vocabulary developers how to add ITS 2.0 markup declarations to their schemas.
- Conformance clauses in Section 4.2: Conformance Type 2: The Processing Expectations for ITS Markup tell implementers how to process XML content according to ITS 2.0 data categories.
- Conformance clauses in Section 4.3: Conformance Type 3: Processing Expectations for ITS Markup in HTML tell implementers how to process [HTML5] content.
- Conformance clauses in Section 4.4: Conformance Type 4: Markup conformance for HTML5+ITS documents tell implementers how ITS 2.0 markup is integrated into [HTML5].

The conformance clauses in Section 4.2: Conformance Type 2: The Processing Expectations for ITS Markup and Section 4.3: Conformance Type 3: Processing Expectations for ITS Markup in HTML clarify how information needs to be made available for given pieces of markup when processing a dedicated ITS 2.0 data category. To allow for flexibility, an implementation can choose whether it wants to support only ITS 2.0 global or local information, or XML or HTML content. These choices are reflected in separate conformance clauses and also in the ITS 2.0 test suite.

ITS 2.0 processing expectations only define which information needs to be made available. They do not define how that information actually is to be used. This is due to the fact that there is a wide variety of usage scenarios for ITS 2.0, and a wide variety of tools for working with ITS 2.0 is possible. Each of these tools may have its own way of using ITS 2.0 data categories (see [MLW US IMPL] for more information).

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 XML implementations of ITS for the above URI is `its`. It is recommended that XML implementations of this specification use this prefix, unless there is existing dedicated markup in use for a given data category. In HTML there is no namespace prefix:

`its-` is used instead to indicate ITS 2.0 attributes in HTML documents. See Section 6.1: Mapping of Local Data Categories to HTML for details.

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

3.2 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 and HTML environment (e.g., using an element or attribute).

For each data category, ITS distinguishes between the following:

- the prose description, see Section 8: Description of Data Categories
- schema language-independent formalization, see the "implementation" subsections in Section 8: Description of Data Categories
- schema language-specific implementations, see Appendix D: Schemas for ITS

Example 10: A data category and its implementation

> The Translate data category conveys information as to whether a piece of content is intended for translation 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 Schema document or a RELAX NG document. A different implementation would be a `translateRule` element that allows for specifying global rules about the Translate data category.

3.3 Selection

[Definition: **selection** encompasses mechanisms to specify to what parts of an XML or HTML document an ITS data category and its values apply.] 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 that is related to selected nodes.

> **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.

W3C Recommendation

Example 11: Selecting the text of a pointer to an external object

```
<text>
  <its:rules version="2.0"
             xmlns:its="http://www.w3.org/2005/11/its">
    <its:translateRule translate="yes" selector="//p/img/@src"/>
  </its:rules>
  ...
  <p xmlns:its="http://www.w3.org/2005/11/its">As you can see in
    <img src="instructions.jpg"/>, the truth is not always out there.</p>
</text>
```

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

3.4 ITS Local Attributes

[Definition: **ITS Local Attributes** are all attributes defined in Section 8: Description of Data Categories as a local markup.]

3.5 Rule Elements

[Definition: **Rule Elements** are all elements defined in Section 8: Description of Data Categories as elements for global rules.]

3.6 Usage of Internationalized Resource Identifiers in ITS

All attributes that have the type `anyURI` in the normative RELAX NG schema in Appendix D: Schemas for ITS MUST allow the usage of Internationalized Resource Identifiers (IRIs, [RFC 3987] or its successor) to ease the adoption of ITS in international application scenarios.

3.7 The Term HTML

This specification uses the term `HTML` to refer to HTML5 or its successor in HTML syntax [HTML5].

3.8 The Term CSS Selectors

This specification uses the term `CSS Selectors` in the sense of `Selectors` as specified in [Selectors Level 3] to prevent confusion with the generic use of the word "selector".

4 Conformance

This section is normative.

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

This specification defines four types of conformance: conformance of 1) ITS markup declarations, conformance of 2) processing expectations for ITS Markup, conformance of 3) processing expectations for ITS Markup in HTML, and 4) markup conformance for HTML5+ITS documents. The conformance type 4 is defined for using ITS markup in HTML5 documents, HTML5+ITS, which serves as an *applicable specification* in the sense specified in the Extensibility section of [HTML5].

W3C Recommendation

These conformance types and classes 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 a schema language independent manner.

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
- *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 `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].

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 that 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 8: 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 the nodes captured by a data category for internationalization or localization. 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.

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 5.5: 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` element.
- *2-3:* If an application claims to process ITS markup implementing the conformance clauses 2-2 and 2-3, it MUST process that markup with XML documents.
- *2-4:* Non-ITS elements and attributes found in ITS elements MAY be ignored.

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

Note:

The above conformance clauses are directly reflected in the ITS 2.0 test suite. All tests specify which data category is processed (clause 2-1); they are relevant for (clause 2-1-1) global or local selection, or both; they require the processing of defaults and precedence of selections (clauses 2-1-2 and 2-1-3); for each data category there are tests with linked rules (2-2); and all types of tests are given for XML (clause 2-3). Implementers are encouraged to organize their documentation in a similar way, so that users of ITS 2.0 easily can understand the processing capabilities available.

4.3 Conformance Type 3: Processing Expectations for ITS Markup in HTML

Description: Processors need to compute the ITS information that pertains to a node in an HTML 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.

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 8: Description of Data Categories have defaults / inheritance / overriding characteristics, and allow for using ITS markup in various positions (local, external global and inline global).

Who uses this conformance type: Applications that need to process the nodes captured by a data category for internationalization or localization. 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.

Conformance clauses:

W3C Recommendation

- *3-1:* A processor MUST implement at least *one* data category. For each implemented data category, the following MUST be taken into account:
    - *3-1-1:* processing of at least one selection mechanism (global or local).
    - *3-1-2:* the default selections for the data category.
    - *3-1-3:* the precedence definitions for selections defined in Section 6.4: Precedence between Selections, for the type of selections it processes.
- *3-2:* If an application claims to process ITS markup for the global selection mechanism, it MUST process a `href` attribute found on a `link` element that has a `rel` attribute with the value `its-rules`.
- *3-3:* If an application claims to process ITS markup implementing the conformance clauses 3-1 and 3-2, it MUST process that markup within HTML documents.

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

4.4 Conformance Type 4: Markup conformance for HTML5+ITS documents

Conforming HTML5+ITS documents are those that comply with all the conformance criteria for documents as defined in [HTML5] with the following exception:

- Conformance clause *4-1:* Global attributes that can be used on all HTML elements are extended by attributes for local data categories as defined in Section 6.1: Mapping of Local Data Categories to HTML.

5 Processing of ITS information

This section is normative.

> **Note:**
>
> Additional definitions about processing of HTML are given in Section 6: Using ITS Markup in HTML.

5.1 Indicating the Version of ITS

The version of the ITS schema defined in this specification is "2.0". 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 on the element where the ITS markup is used, or on one of its ancestors.

If there is no `rules` element and there are elements with standoff ITS markup in an XML document, an ITS `version` attribute MUST be provided on element with standoff ITS markup or a prefixed ITS `version` attribute (e.g., `its:version`) MUST be provided on one of its ancestors.

There MUST NOT be two different versions of ITS in the same document.

External, linked rules can have different versions than internal rules.

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 absolute selector as defined in Section 5.3: Query Language of Selectors.
- Locally in a document: the selection is realized using ITS local attributes, which are attached to an element node, or the span 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 8.1: 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. The rules element 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.

The data category overview table, in Section 8.1: Position, Defaults, Inheritance, and Overriding of Data Categories, provides an overview of what data categories allow to point to existing information or to add information.

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

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 5.5: Precedence between Selections.

5.2.2 Local Selection in an XML Document

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

The data category determines what is being selected. The necessary data category specific defaults are described in Section 8.1: Position, Defaults, Inheritance, and Overriding of Data Categories.

Example 12: 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, is not intended for translation. The attribute `its:translate="yes"` in the `title` element means that the content of this element, is to 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="2.0" 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">التدويل نشاط، W3
        <quote>Internationalization Activity, W3C</quote>.</par>
  </body>
</text>
```

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

Note:

The `dir` and `translate` attributes are not listed in the ITS attributes to be used in HTML. The reason is that these two attributes are available in HTML natively, so there is no need to provide them as `its-` attributes. The definition of the two attributes in HTML is compatible, that is it provides the same values and interpretation, as the definition for the two data categories Translate and Directionality.

5.3 Query Language of Selectors

5.3.1 Choosing Query Language

Rule elements have attributes that contain absolute and relative selectors. Interpretation of these selectors depends on the actual query language. The query language is set by `queryLanguage` attribute on `rules` element. If `queryLangauge` is not specified XPath 1.0 is used as a default query language.

5.3.2 XPath 1.0

XPath 1.0 is identified by `xpath` value in `queryLanguage` attribute.

5.3.2.1 Absolute selector

The absolute selector MUST be an XPath expression that starts with "`/`". That is, it MUST be an AbsoluteLocationPath or union of AbsoluteLocationPaths as described in XPath 1.0. This ensures that the selection is not relative to a specific location. The resulting nodes MUST be either element or attribute nodes.

Context for evaluation of the XPath expression is as follows:

- Context node is set to Root Node.
- Both context position and context size are 1.
- All variables defined by `param` elements are bind.
- All functions defined in the XPath Core Function Library are available. It is an error for an expression to include a call to any other function.
- The set of namespace declarations are those in scope on the element that has the attribute in which the expression occurs. This includes the implicit declaration of the prefix `xml` required by the XML Namespaces Recommendation; the default namespace (as declared by `xmlns`) is not part of this set.

Example 13: XPath expressions with namespaces

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

```
<!-- Definitions for TEI -->
<its:rules version="2.0"  xmlns:its="http://www.w3.org/2005/11/its">
 <its:termRule selector="//tei:term" term="yes" xmlns:tei="http://www.tei-c.org/
</its:rules>
```

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

Example 14: XPath expressions without namespaces

The `term` element from DocBook V4.5 is in no namespace.

```
<!-- Definitions for DocBook -->
<its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
  <its:termRule selector="//term" term="yes"/>
</its:rules>
```

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

5.3.2.2 Relative selector

The relative selector MUST use a RelativeLocationPath or an AbsoluteLocationPath as described in XPath 1.0. The XPath expression is evaluated relative to the nodes selected by the selector attribute.

The following attributes point to existing information: `allowedCharactersPointer`, `taClassRefPointer`, `taIdentPointer`, `taIdentRefPointer`, `taSourcePointer`, `domainPointer`, `externalResourceRefPointer`, `langPointer`, `locNotePointer`, `locNoteRefPointer`, `locQualityIssuesRefPointer`, `provenanceRecordsRefPointer`, `storageEncodingPointer`, `storageSizePointer`, `targetPointer`, `termInfoPointer`, `termInfoRefPointer`.

Context for evaluation of the XPath expression is the same as for an absolute selector with the following changes:

- Nodes selected by the expression in the `selector` attribute form the current node list.
- Context node comes from the current node list.

- The context position comes from the position of the current node in the current node list; the first position is 1.
- The context size comes from the size of the current node list.

5.3.3 CSS Selectors

Note:

The term `CSS Selectors` is used throughout the specification in the sense of `Selectors` as specified in [Selectors Level 3] to prevent confusion with the generic use of the word "selector". See The term CSS Selector.

Note:

The working group will not provide a CSS Selectors-based implementation; nevertheless there are several existing libraries that can translate CSS Selectors to XPath so that XPath selectors-based implementations can be used.

Note:

CSS selectors have no ability to point to attributes.

CSS Selectors are identified by the value `css` in the `queryLanguage` attribute.

5.3.3.1 Absolute selector

An absolute selector MUST be interpreted as a selector as defined in [Selectors Level 3]. Both simple selectors and groups of selectors can be used.

5.3.3.2 Relative selector

A relative selector MUST be interpreted as a selector as defined in [Selectors Level 3]. A selector is not evaluated against the complete document tree but only against subtrees rooted at nodes selected by the selector in the `selector` attribute.

5.3.4 Additional query languages

ITS processors MAY support additional query languages. For each additional query language the processor MUST define:

- the identifier of the query language used in `queryLanguage`;
- rules for evaluating an absolute selector to a collection of nodes;
- rules for evaluating a relative selector to a collection of nodes.

Because future versions of this specification are likely to define additional query languages, the following query language identifiers are reserved: `xpath`, `css`, `xpath2`, `xpath3`, `xquery`, `xquery3`, `xslt2`, `xslt3`.

5.3.5 Variables in selectors

A `param` element (or several ones) can be placed as the first child element(s) of the `rules` element to define the default values of variables used in the various selectors used in the rules.

An implementation MUST support the param element for all query languages it supports and at the same time define how variables are bound for evaluation of the selector expression. Implementations SHOULD also provide means for changing the default values of the param elements. Such means are implementation-specific.

The param element has a required name attribute. The value of the name attribute is a QName, see [XML Names]. The content of the element is a string used as default value for the corresponding variable.

Example 15: Using the param element to define the default value of a variable in a selector attribute.

The param element defines the default value for the $LCID variable. In this case, only the msg element with the attribute lcid set to "0x049" is seen as translatable.

```
<doc its:version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
  <its:rules version="2.0">
    <its:param name="LCID">0x0409</its:param>
    <its:translateRule selector="/doc" translate="no"/>
    <its:translateRule selector="//msg[@lcid=$LCID]" translate="yes"/>
  </its:rules>
  <msg lcid="0x0409" num="1">Create a folder</msg>
  <msg lcid="0x0411" num="1">フォルダーを作成する</msg>
  <msg lcid="0x0407" num="1">Erstellen Sie einen Ordner</msg>
  <msg lcid="0x040c" num="1">Créer un dossier</msg>
</doc>
```

[Source file: examples/xml/EX-param-in-global-rules-1.xml]

Note:

In XSLT-based applications, it may make sense to map ITS parameters directly to XSLT parameters. To avoid naming conflicts one can use a prefix with the parameter name's value to distinguish between the ITS parameters and the XSLT parameters.

5.4 Link to External Rules

One way to associate a document with a set of external ITS rules is to use the optional XLink [XLink 1.1] href attribute in the rules element. The referenced document MUST be a valid XML document containing at most one rules element. That rules element can be the root element or be located 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>
  <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="2.0" xmlns:its="http://www.w3.org/2005/11/its">
    <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: examples/xml/EX-link-external-rules-1.xml]

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

```
<myDoc>
  <header>
    <its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its"
      xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="EX-link-external-rul
      <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: examples/xml/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>
  <header>
    <its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
      <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: examples/xml/EX-link-external-rules-3.xml]

Example 19: External rules file with the `rules` element as the root element

As with Example 16, these rules can be applied to Example 17. The only difference is that in Example 19, the `rules` element is the root element of the external file.

```
<its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
  <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>
```

[Source file: examples/xml/EX-link-external-rules-4.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 (see Example 17). The linking mechanism is recursive in a depth-first approach, and subsequently after the processing the rules MUST be read top-down (see Example 18).

5.5 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. Selection via explicit (i.e., not inherited) local ITS markup 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 rule inside the rules element
   2. Any rule linked via the XLink `href` attribute

   Note:

   ITS does not define precedence related to rules defined or linked based on non-ITS mechanisms (such as processing instructions for linking rules).
3. Selection via inherited values. This applies only to element nodes. The inheritance rules are laid out in a dedicated data category overview table: see the column "*Inheritance for element nodes*". Selection via inheritance takes precedence over default values, see below item.
4. Selections via defaults for data categories, see Section 8.1: Position, Defaults, Inheritance, and Overriding of Data Categories

In case of conflicts between global selections via multiple rules elements or conflicts between multiple `param` elements with the same name, the last rule or last `param` element has higher precedence.

Note:

The precedence order fulfills the same purpose as the built-in template rules of [XSLT 1.0]. Override semantics are always complete, that is all information provided via lower precedence is overridden by the higher precedence. E.g. defaults are overridden by inherited values and these are overridden by nodes selected via global rules, which are in turn overridden by local markup.

Example 20: Conflicts between selections of ITS information resolved using the precedence order

The two elements title and author of this document are intended as separate content when inside a prolog element, but in other contexts as part of the content of their parent element. In order to make this distinction two withinTextRule elements are used:

The first rule specifies that title and author in general are to 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 is to be treated separately. This is normally the default, but the rule is needed to override the first rule.

```
<text>
  <prolog>
    <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
      <its:withinTextRule withinText="yes" selector="//title|//author"/>
      <its:withinTextRule withinText="no" selector="//prolog/title|//prolog/autho
    </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</a
      the best introductions to the vast topic of designing user interfaces.</p>
  </body>
</text>
```

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

5.6 Associating ITS Data Categories with Existing Markup

Some markup schemes provide markup that 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. For example, the [DITA 1.0] format can use its translate attribute to apply to “transcluded” content, going beyond the ITS 2.0 local selection mechanism, but not contradicting it.

Example 21: Association of the ITS data categories Translate and Terminology with DITA 1.0 markup

In this example, there is an existing translate attribute in DITA, and it is associated with the ITS semantics using the its:rules section. Similarly, the DITA dt and term elements are associated with the ITS Terminology data category.

```
<topic id="myTopic">
  <title>The ITS Topic</title>
  <prolog>
    <its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
      <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 p
            information related to internationalization and localization of XML sch
            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: examples/xml/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 21
    - 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, in the case of a command-line tool by providing the paths of both the XML document to process and its corresponding external rules file.

5.7 ITS Tools Annotation

In some cases, it may be important for instances of data categories to be associated with information about the processor that generated them. For example, the score of the MT Confidence data category (provided via the `mtConfidence` attribute) is meaningful only when the consumer of the information also knows which MT engine produced it, because the score provides the relative confidence of translations from the same MT engine but does not provide a score that can be reliably compared between MT engines. The same is true for confidence provided for the Text Analysis data category, providing confidence information via the `taConfidence` attribute, or the Terminology data category, providing confidence information via the `termConfidence` attribute.

ITS 2.0 provides a mechanism to associate such processor information with the use of individual data categories in a document, independently from data category annotations themselves.

The attribute `annotatorsRef` provides a way to associate all the annotations of a given data category within the element with information about the processor that generated those data category annotations.

Note:

- Three cases of providing tool information can be expected:
    1. information about tools used for creating or modifying the textual content;

2. information about tools that do 1), but also create ITS annotations, see Appendix I: List of ITS 2.0 Global Elements and Local Attributes;
3. information about tools that don’t modify or create content, but just create ITS annotations.

   annotatorsRef is only meant to be used when actual ITS annotation is involved, that is for 2) and 3). To express tool information related only to the creation or modification of textual content and independent of ITS data categories, that is case 1), the tool or toolRef attribute provided by the Provenance data category is to be used.

- An example of case 2) is an MT engine that modifies content and creates ITS MT Confidence annotations. Here the situation may occur that several tools are involved in creating MT Confidence annotations: the MT engine and the tool inserting the markup. The annotatorsRef attribute is to identify the tool most useful in further processes, in this case the MT engine.

The value of annotatorsRef is a space-separated list of references where each reference is composed of two parts: a data category identifier and an IRI. These two parts are separated by a | VERTICAL LINE (U+007C) character:

- The data category identifier MUST be one of the identifiers specified in the data category overview table.
- Within one annotatorsRef value, a data category identifier MUST NOT appear more than one time.
- The IRI indicates information about the processor used to generate the data category annotation. No single means is specified for how this IRI has to be used to indicate processor information. Possible mechanisms are: to encode information directly in the IRI, e.g., as parameters; to reference an external resource that provides such information, e.g. an XML file or an RDF declaration; or to reference another part of the document that provides such information.

In HTML documents, the mechanism is implemented with the its-annotators-ref attribute.

The attribute applies to the content of the element where it is declared (including its children elements) and to the attributes of that element.

On any given node, the information provided by this mechanism is a space-separated list of the accumulated references found in the annotatorsRef attributes declared in the enclosing elements and sorted by data category identifiers. For each data category, the IRI part is the one of the inner-most declaration.

Example 22: Accumulation and Overriding of the annotatorsRef Values

In this example, the text shows the computed tools reference information for the given node. Note that the references are ordered alphabetically and that the IRI values are always the ones of the inner-most declaration.

```
<doc its:version="2.0" xmlns:its="http://www.w3.org/2005/11/its"
  its:annotatorsRef="mt-confidence|MT1"
  >doc node: "mt-confidence|MT1"
  <group its:annotatorsRef="terminology|ABC"
    >group node: "mt-confidence|MT1 terminology|ABC"
    <p its:annotatorsRef="text-analysis|Tool3"
      >This p node: "text-analysis|Tool3 mt-confidence|MT1 terminology|ABC"</p>
    <p its:annotatorsRef="mt-confidence|MT123"
      >This p node: "mt-confidence|MT123 terminology|ABC"</p>
 </group>
  <!-- To make this example usable in real life, we would have
    annotations of the three data categories - text-analysis, mt-confidence and t
```

```
     - in the document -->
   <p its:annotatorsRef="text-analysis|XYZ"
    >This p node: "text-analysis|XYZ mt-confidence|MT1"</p>
</doc>
```

[Source file: examples/xml/EX-its-tool-annotation-1.xml]

Example 23: Example of ITS Tools Annotation

The annotatorsRef attribute is used in this XML document to indicate that information about the processor that generated the mtConfidence values for the first two p elements are found in element with id="T1" in the external document tools.xml, while that information for the third p element is found in the element with id="T2" in the same document. In addition, annotatorsRef is used to identify a Web resource with information about the QA tool used to generate the Localization Quality Issue annotation in the document.

```
<doc its:version="2.0"
    its:annotatorsRef=
    "mt-confidence|file:///tools.xml#T1 localization-quality-issue
    |http://www.qalsp-ex.com/qatools/transcheckv1.3"
    xmlns:its="http://www.w3.org/2005/11/its">
    <p its:mtConfidence="0.78">Text translated with tool T1</p>
    <p its:mtConfidence="0.55" its:locQualityIssueType="typographical"
        its:locQualityIssueComment="Sentence without capitalization"
        its:locQualityIssueSeverity="50">text also translated with tool T1</p>
    <p its:mtConfidence="0.34"
      its:annotatorsRef="mt-confidence|file:///tools.xml#T2"> Text translated
        with tool T2</p>
</doc>
```

[Source file: examples/xml/EX-its-tool-annotation-2.xml]

Example 24: Example of ITS Tool Annotation

The its-annotators-ref attributes are used in this HTML document to indicate that the MT Confidence annotation on the first two span elements come from one MT (French to English) engine, while the annotation on the third comes from another (Italian to English) engine. Both its-annotators-ref attributes refer to a Web resource for information about the engine generating the MT Confidence annotation.

```
<!DOCTYPE html>
<html lang=en>
    <head>
        <meta charset=utf-8>
            <title>Sentences about capital cities
                machine translated into English with mtConfidence defined
                locally.</title>
    </head>
    <body its-annotators-ref="mt-confidence|http://www.exmt-prov.com/2012/11/9/fr
        <p>
 <span its-mt-confidence=0.8982>Dublin is the capital of Ireland.</span>
 <span its-mt-confidence=0.8536>The capital of the Czech Republic is Prague.</spa
 <span its-mt-confidence=0.7009
  its-annotators-ref="mt-confidence|http://www.exmt-prov.com/2012/11/9/it-t-en">
  The capital Italia is Roma.</span>
        </p>
    </body>
</html>
```

[Source file: examples/html5/EX-its-tool-annotation-html5-1.html]

6 Using ITS Markup in HTML

This section is normative.

Note:

Please note that the term `HTML` refers to HTML5 or its successor in HTML syntax [HTML5].

6.1 Mapping of Local Data Categories to HTML

All data categories defined in Section 8: Description of Data Categories and having local implementation may be used in HTML with the exception of the Translate, Directionality and Language Information data categories.

Note:

The above-mentioned data categories are excluded because HTML has native markup for them.

In HTML data categories are implemented as attributes. The name of the HTML attribute is derived from the name of the attribute defined in the local implementation by using the following rules:

1. The attribute name is prefixed with `its-`
2. Each uppercase letter in the attribute name is replaced by `-` (U+002D) followed by a lowercase variant of the letter.

Example 48 demonstrates the Elements Within Text data category with the local XML attribute `withinText`. Example 49 demonstrates the counterpart in HTML, i.e., the local attribute `its-within-text`.

Values of attributes, which corresponds to data categories with a predefined set of values, MUST be matched ASCII-case-insensitively.

Note:

Case of attribute names is also irrelevant given the nature of HTML syntax. So in HTML the terminology data category can be stored as `its-term`, `ITS-TERM`, `its-Term` etc. All of those attributes are treated as equivalent and will be normalized upon DOM construction.

Values of attributes that correspond to data categories that use XML Schema double data type MUST be also valid floating-point numbers as defined in [HTML5].

6.2 Global rules

Various aspects for global rules in general, external global rules, or inline global rules need to be taken into account. An example of an HTML5 document using global rules is Example 6. The corresponding rules file is Example 7.

Note:

By default XPath 1.0 will be used for selection in global rules. If users prefer an easier selection mechanism, they can switch query language to CSS selectors by using the queryLanguage attribute, see Section 5.3.1: Choosing Query Language.

Note:

The HTML5 parsing algorithm automatically puts all HTML elements into the XHTML namespace (http://www.w3.org/1999/xhtml). Selectors used in global rules need to take this into account.

Linking to external global rules is specified in the href attribute of link elements, with the link relation its-rules.

Note:

Using XPath in global rules linked from HTML documents does not create an additional burden to implementers. Parsing HTML content produces a DOM tree that can be directly queried using XPath, functionality supported by all major browsers.

Inline global rules MUST be specified inside a script element that has a type attribute with the value application/its+xml. The script element itself SHOULD be a child of the head element. Comments MUST NOT be used inside global rules. Each script element MUST NOT contain more than one rules element.

Note:

It is preferred to use external global rules linked using the link element than to have global rules embedded in the document.

6.3 Standoff Markup in HTML

The constraints for Provenance standoff markup in HTML and Localization quality issues markup in HTML MUST be followed.

6.4 Precedence between Selections

The following precedence order is defined for selections of ITS information in various positions of HTML document (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 the mechanism of external global rules or inline global rules), to be processed in a document order, see Section 5.2.1: Global, Rule-based Selection for details.

   Note:

   ITS does not define precedence related to rules defined or linked based on non-ITS mechanisms (such as processing instructions for linking rules). Selection via inheritance takes precedence over default values (see below).
3. Selection via inherited values. This applies only to element nodes. The inheritance rules are laid out in a dedicated data category overview table (see the column "*Inheritance for element nodes). Selection via inheritance takes precedence over default values (see below).*"
4. Selections via defaults for data categories, see Section 8.1: Position, Defaults, Inheritance, and Overriding of Data Categories.

In case of conflicts between global selections via multiple rules elements or conflicts between multiple param elements with the same name, the last rule or last param element has higher precedence.

Example 6, previously discussed, demonstrates the precedence: the code element with the translate attribute set to yes has precedence over the global rule setting all code elements as untranslatable.

7 Using ITS Markup in XHTML

This section is normative.

XHTML documents aimed at public consumption by Web browsers, including HTML5 documents in XHTML syntax, SHOULD use the syntax described in Section 6: Using ITS Markup in HTML in order to adhere to DOM Consistency HTML Design Principle.

Example 25: Using ITS 2.0 markup in XHTML

This example illustrates the use of ITS 2.0 local markup in XHTML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
    <head>
        <title>XHTML and ITS2.0</title>
    </head>
    <body>
        <h1>XHTML and ITS2.0</h1>
        <p>Don't use
          <span its-loc-note="Internationalization Tag Set">ITS</span> prefixed
            attributes inside the content, like its:locNote.</p>
    </body>
</html>
```

[Source file: examples/html5/EX-xhtml-markup-1.html]

Note:

Please note that this section defines how to use ITS in XHTML content that is directly served to Web browsers. Such XHTML is very often sent with an incorrect media type and parsed as HTML rather than XML in Web browsers. In such case it is more robust and safer to use HTML-like syntax for ITS metadata.

However when XHTML is not used as a delivery but rather as an exchange or storage format all XML features can be used in XHTML and it is advised to use XML syntax for ITS metadata.

8 Description of Data Categories

This section is normative.

8.1 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 apply. It also provides data category identifiers used in Section 5.7: ITS Tools Annotation:

- *Default values* apply if both local and 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 nodes or their child notes are not translatable.
- For ITS data categories with inheritance, the information conveyed by the data category can be overridden. For example, a local `translate` attribute overrides the Translate information conveyed by a global `translateRule`.

Foreign elements can be used only inside `rules`. Foreign attributes can be used on any element defined in ITS.

Note:

An ITS application is free to decide what pieces of content it uses. For example:

- Terminology information is added to a `term` element. The information pertains only to the content of the element, since there is no inheritance for Terminology. Nevertheless an ITS application can make use of the complete element, e.g., including attribute nodes etc.
- Using ID Value, a unique identifier is provided for a `p` element. An application can make use of the complete `p` element, including child nodes and attributes nodes. The application is also free to make use just of the string value of `p`. Nevertheless the id provided via ID Value pertains only to the `p` element. It cannot be used to identify nested elements or attributes.
- Using target pointer, selected `source` elements have the ITS information that their translation is available in a `target` element; see Example 65. This information does not inherit to child elements of `target pointer`. E.g., the translation of a `span` element nested in `source` is not available in a specific `target` element. Nevertheless, an application is free to use the complete content of `source`, including `span`, and, e.g., present it to a translator.

Data category (`identifier`)	Local Usage	Global, rule-based selection	Global adding of information	Global pointing to existing information	Default Values	Inheritance for elements nodes	Ex
Translate (`translate`)	Yes	Yes	Yes	No	For XML: `translate="yes"` for elements, and `translate="no"` for attributes. For [HTML5]: see HTLM5 Translate Handling.	For XML: Textual content of element, *including* content of child elements, but *excluding* attributes.	loc glo

Data category (identifier)	Local Usage	Global, rule-based selection	Global adding of information	Global pointing to existing information	Default Values	Inheritance for elements nodes	Ex
						For [HTML5]: see HTLM5 Translate Handling.	
Localization Note (localization-note)	Yes	Yes	Yes	Yes	None	Textual content of element, *including* content of child elements, but *excluding* attributes	loc glo
Terminology (terminology)	Yes	Yes	Yes	Yes	term="no"	None	loc glo
Directionality (directionality)	Yes	Yes	Yes	No	dir="ltr"	Textual content of element, *including* attributes and child elements	loc glo
Language Information (language-information)	No	Yes	No	Yes	None	Textual content of element, *including* attributes and child elements	glo
Elements Within Text (elements-within-text)	Yes	Yes	Yes	No	For XML content: withinText="no". For [HTML5]: see HTLM5 Element Within Text Handling.	None	loc glo
Domain (domain)	No	Yes	Yes	Yes	None	Textual content of element, *including* attributes and child elements	glo
Text Analysis (text-analysis)	Yes	Yes	Yes	Yes	None	None	loc glo
Locale Filter (locale-filter)	Yes	Yes	Yes	No	localeFilterList="*", localeFilterType="include"	Textual content of element, *including* attributes and child elements	loc glo

Data category (identifier)	Local Usage	Global, rule-based selection	Global adding of information	Global pointing to existing information	Default Values	Inheritance for elements nodes	Ex
Provenance (provenance)	Yes	Yes	No	Yes	None	Textual content of element, *including* child elements and attributes	loc glo
External Resource (external-resource)	No	Yes	No	Yes	None	None	glo
Target Pointer (target-pointer)	No	Yes	No	Yes	None	None	glo
ID Value (id-value)	No	Yes	No	Yes	None	None	glo
Preserve Space (preserve-space)	Yes	Yes	Yes	No	default	Textual content of element, *including* attributes and child elements	loc glo
Localization Quality Issue (localization-quality-issue)	Yes	Yes	Yes	Yes	None	Textual content of element, *including* child elements, but excluding attributes	loc glo
Localization Quality Rating (localization-quality-rating)	Yes	No	No	No	None	Textual content of element, *including* child elements, but excluding attributes	loc
MT Confidence (mt-confidence)	Yes	Yes	Yes	No	None	Textual content of element, *including* child elements, but excluding attributes	loc glo
Allowed Characters (allowed-characters)	Yes	Yes	Yes	Yes	None	Textual content of element,	loc glo

Data category (identifier)	Local Usage	Global, rule-based selection	Global adding of information	Global pointing to existing information	Default Values	Inheritance for elements nodes	Ex
						including child elements, but excluding attributes	
[Storage Size](#) (storage-size)	Yes	Yes	Yes	Yes	None	None	loc glo

Example 26: Defaults, inheritance and overriding behavior of data categories

In this example, the content of all the `data` elements is translatable and none of the attributes are translatable, because the default for the [Translate](#) data category in elements is "yes" and in attributes is "no", and neither of their values are overridden at all. The first `translateRule` is overridden by the local `its:translate="no"` attribute. The content of `revision`, `profile`, `reviser` and `locNote` elements are not translatable. This is because the default is overridden by the same `its:translate="no"` that these elements inherit from the local ITS markup in the `prolog` element. The exception is the `field` element where the second `translateRule` takes precedence over the inherited value. The last `translateRule` indicates that the content of `type` is not translatable because the global rule takes precedence over the default value.

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

```
<Res xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <prolog its:translate="no">
    <revision>Sep-07-2006</revision>
    <profile>
      <reviser>John Doe</reviser>
      <field>Computing Engineering</field>
    </profile>
    <its:rules version="2.0">
      <its:translateRule selector="//prolog" translate="yes"/>
      <its:translateRule selector="/Res/prolog/profile/field" translate="yes"/>
      <its:translateRule selector="//msg/type" 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">
      <type>Error</type>
      <data>Host {0} cannot be found.</data>
    </msg>
    <msg id="HostDisconnected">
      <type>Error</type>
      <data>The connection with {0} has been lost.</data>
    </msg>
    <msg id="FileNotFound">
      <type>Error</type>
      <data its:locNote="{0} is a filename">{0} not found.</data>
    </msg>
  </body>
</Res>
```

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

Note:

The data categories differ with respect to defaults. This difference 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.

8.2 Translate

8.2.1 Definition

The Translate data category expresses information about whether the content of an element or attribute is intended for translation or not. The values of this data category are "yes" (translatable) or "no" (not translatable).

8.2.2 Implementation

The Translate data category can be expressed with global rules, or locally on an individual element. Handling of inheritance and interaction between elements and attributes is different for XML content versus [HTML5] content.

For XML: for elements, the data category information inherits to the textual content of the element, *including* child elements, but *excluding* attributes. The default is that elements are translatable and attributes are not.

For HTML: The interpretation of the `translate` attribute is given in HTML5. Nodes in an HTML document selected via a global rule are also interpreted following HTML5.

Note:

As of writing, the default in [HTML5] is that elements are translatable, and that translatable attributes inherit from the respective elements. There is a pre-defined list of translatable attributes, for example `alt` or `title`.

Since the [HTML5] definition also applies to nodes selected via global rules, a `translateRule` like `<its:translateRule selector=""//h:img" translate="yes"/>` will set the `img` element and its translatable attributes like `alt` to "yes".

GLOBAL: The `translateRule` element contains the following:

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

Example 27: The Translate data category expressed globally

The `translateRule` element specifies that the elements `code` is not to be translated.

```
<its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
  <its:translateRule translate="no" selector="//code"/>
</its:rules>
```

[Source file: examples/xml/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".

In [HTML5] the native [HTML5] `translate` attribute MUST be used to express the Translate data category.

Note:

For XML content, 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. If attributes need to be translatable, then this has to be declared globally. Note that this restriction does not apply to HTML5.

Example 28: The Translate data category expressed locally

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

```
<messages its:version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
  <msg num="123">Click Resume Button on Status Display or <panelmsg its:translate
      >CONTINUE</panelmsg> Button on printer panel</msg>
</messages>
```

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

Example 29: The Translate data category expressed locally in HTML

The local `translate="no"` attribute specifies that the content of `span` is not to be translated.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=utf-8>
    <title>Translate flag test: Default</title>
  </head>
  <body>
    <p>The <span translate=no>World Wide Web Consortium</span> is
      making the World Wide Web worldwide!</p>
  </body>
</html>
```

[Source file: examples/html5/EX-translate-html5-local-1.html]

8.3 Localization Note

8.3.1 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 in 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 has to 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.

8.3.2 Implementation

The Localization Note data category can be expressed with global rules, or locally on an individual element. For elements, the data category information inherits 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 absolute selector that 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 selector pointing to a node that holds the localization note.
  - A `locNoteRef` attribute that contains an IRI referring to the location of the localization note.
  - A `locNoteRefPointer` attribute that contains a relative selector pointing to a node that holds the IRI referring to the location of the localization note.

Example 30: 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 is in an external file, allowing it to provide notes without modifying the source document.

```
<myRes>
  <head>
    <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0" its:transl
      <its:locNoteRule locNoteType="alert" selector="//msg[@id='DisableInfo']">
        <its:locNote>The variable {0} has three possible values: 'printer', 'stac
          options'.</its:locNote>
      </its:locNoteRule>
    </its:rules>
  </head>
  <body>
    <msg id="DisableInfo">The {0} has been disabled.</msg>
  </body>
</myRes>
```

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

Example 31: The locNotePointer attribute

The locNotePointer attribute is a relative selector pointing to a node that holds the note.

```
<Res>
  <prolog>
    <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
      <its:translateRule selector="//msg/notes" translate="no"/>
      <its:locNoteRule locNoteType="description" selector="//msg/data" locNotePoi
    </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: examples/xml/EX-locNotePointer-attribute-1.xml]

Example 32: The locNoteRef attribute

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

```
<myRes>
  <head>
    <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
      <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: examples/xml/EX-locNoteRef-attribute-1.xml]

Example 33: The locNoteRefPointer attribute

The locNoteRefPointer attribute contains a relative selector pointing to a node that holds the IRI referring to the location of the note.

```
<dataFile>
  <prolog>
    <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
      <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: examples/xml/EX-locNoteRefPointer-attribute-1.xml]

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

- Exactly one of the following:
  - A locNote attribute that contains the note itself.
  - A locNoteRef attribute that contains an IRI 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 34: The Localization Note data category expressed locally

```
<msgList xmlns:its="http://www.w3.org/2005/11/its" xml:space="preserve" its:versi
 <data name="LISTFILTERS_VARIANT" its:locNote="Keep the leading space!" its:locNo
  <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
  <value>Translated from English content dated <span id="version-info">%1\$s</spa
 </data>
</msgList>
```

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

Example 35: The Localization Note data category expressed locally in HTML

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <title>LocNote test: Default</title>
  </head>
  <body>
    <p>This is a
    <span its-loc-note="Check with terminology engineer" its-loc-note-type=alert>
     motherboard</span>.</p>
  </body>
</html>
```

[Source file: examples/html5/EX-locNote-html5-local-1.html]

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.

8.4 Terminology

8.4.1 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 30042] 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.

8.4.2 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 absolute selector that selects the nodes to which this rule applies.
- A required `term` attribute with the value "yes" or "no".
- Zero or one of the following:
  - A `termInfoPointer` attribute that contains a relative selector pointing to a node that holds the terminology information.
  - A `termInfoRef` attribute that contains an IRI referring to the resource providing information about the term.
  - A `termInfoRefPointer` attribute that contains a relative selector pointing to a node that holds the IRI referring to the location of the terminology information.

Example 36: Usage of the termInfoPointer attribute

```
<text>
  <its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
    <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,
    implied author or some other addresser, and the fiction.</gloss></p>
</text>
```

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

Example 37: Usage of the termInfoRef attribute

```
<text>
  <its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
    <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: examples/xml/EX-terms-selector-2.xml]

Example 38: Usage of the termInfoRefPointer attribute

```
<text>
  <its:rules version="2.0"  xmlns:its="http://www.w3.org/2005/11/its">
    <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: examples/xml/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 an IRI referring to the resource providing information about the term.
- An optional termConfidence attribute with the value of a rational number in the interval 0 to 1 (inclusive). The value follows the XML Schema double data type with the constraining facets minInclusive set to 0 and maxInclusive set to 1. termConfidence represents the confidence of the agents producing the annotation that the annotated unit is a term or not. 1 represents the highest level of confidence. termConfidence does not provide confidence information related to termInfoRef.

Any node selected by the terminology data category with the `termConfidence` attribute specified MUST be contained in an element with the `annotatorsRef` (or in HTML `its-annotators-ref`) attribute specified for the Terminology data category. See Section 5.7: ITS Tools Annotation for more information.

Example 39: The Terminology data category expressed locally, including term information reference and confidence score.

```
<book its:version="2.0" xmlns:its="http://www.w3.org/2005/11/its"
  its:annotatorsRef="terminology|http://example.com/term-tool">
  <head>...</head>
  <body>
    ...
    <p>And he said: you need a new
      <quote its:term="yes"
        its:termInfoRef="http://www.directron.com/motherboards1.html"
        its:termConfidence="0.5">motherboard</quote></p>
    ...
 </body>
</book>
```

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

Example 40: The Terminology data category expressed locally in HTML

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <title>Terminology test: default</title>
  </head>
  <body>
    <p>We need a new <span its-term=yes>motherboard</span>
    </p>
  </body>
</html>
```

[Source file: examples/html5/EX-term-html5-local-1.html]

8.5 Directionality

Note:

At the time of writing, enhancements are being discussed in the context of HTML5 that are expected to change the approach to marking up Directionality, in particular to support content where directionality needs to be isolated from that of surrounding content. However, these enhancements are not finalized yet. This section therefore reflects directionality markup in [HTML 4.01]; enhancements in HTML5 will be reflected in a future revision.

8.5.1 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".

W3C Recommendation

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: "lro" (left-to-right override)

  CSS rule: `*[dir="lro"] { unicode-bidi: bidi-override; direction: ltr}`
- Data category value: "rlo" (right-to-left override)

  CSS rule: `*[dir="rlo"] { unicode-bidi: bidi-override; direction: rtl}`

More information about how to use this data category is provided by [Bidi Article].

8.5.2 Implementation

The Directionality data category can be expressed with global rules, or locally on an individual element. For elements, the data category information inherits 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 absolute selector that selects the nodes to which this rule applies.
- A required `dir` attribute with the value "ltr", "rtl", "lro" or "rlo".

Example 41: Document that 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 "Internationalization Activity, W3C",
      and the order of characters is <bdo direction='rtlText'> פעילות
הבינאום          , W3C</bdo>.</par>
  </body>
</text>
```

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

Example 42: 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, except that bdo elements with that attribute have right-to-left override content.

```
<its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
  <its:dirRule dir="rtl" selector="//*[@direction='rtlText']"/>
  <its:dirRule dir="rlo" selector="//bdo[@direction='rtlText']"/>
</its:rules>
```

[Source file: examples/xml/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".

Note:

[HTML 4.01] does not have the "lro" and "rlo" values for its dir attribute, so these values are not used for HTML documents. HTML uses an inline bdo element instead.

Example 43: The Directionality data category expressed locally

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

```
<text xml:lang="en" xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <body>
    <par>In Arabic, the title <quote xml:lang="ar" its:dir="rtl">نشاط التدويل،
        W3C</quote> means "Internationalization Activity, W3C".</par>
  </body>
</text>
```

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

Example 44: The Directionality data category expressed locally in HTML

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <title>Dir test: Default</title>
  </head>
  <body>
    <p>In Arabic, the title <q dir=rtl lang=ar>نشاط التدويل، W3C</q>
      means "Internationalization Activity, W3C".</p>
  </body>
</html>
```

[Source file: examples/html5/EX-dir-html5-local-1.html]

8.6 Language Information

8.6.1 Definition

The element langRule is used to express the language of a given piece of content. The langPointer attribute points to the markup that 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 in XML, and lang in HTML. The langRule element is intended only as a fall-back mechanism for documents where language is identified with another construct.

Example 45: 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 version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
  <its:langRule selector="//p" langPointer="@mylangattribute"/>
</its:rules>
```

[Source file: examples/xml/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 lang in HTML, or an attribute specific to the format in question (as in Example 45).

In XML 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 1.0].

In HTML lang is the mandated means of language identification.

8.6.2 Implementation

The Language Information data category can be expressed only with global rules. For elements, the data category information inherits 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 absolute selector that selects the nodes to which this rule applies.
- A required langPointer attribute that contains a relative selector pointing to a node that contains language information. If the attribute xml:lang is present or lang in HTML for the

selected node, the value of the `xml:lang` attribute or `lang` in HTML MUST take precedence over the `langPointer` value.

8.7 Elements Within Text

8.7.1 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>
  ```

8.7.2 Implementation

The Elements Within Text data category can be expressed with global rules, or locally on an individual element. There is no inheritance.

For XML: The default is that elements are not within text.

For HTML: The default is that elements are not within text, with the following exceptions:

- For the elements that are part of the HTML5 phrasing content the default is `withinText="yes"`, with the following exceptions:
  - For the elements `iframe`, `noscript`, `script` and `textarea` the default is `withinText="nested"`.

Example 46: Illustrates the defaults for the Elements Within Text data category in HTML.

In this document the different flows of text are the following (brackets indicating inline or nested elements):

```
- "Elements within Text defaults for HTML5"
- "The element p is not within text. But [the element em is]."
- "A button [Click Here] is also within text. But [] is nested."
- "The content of textarea"
- "Some additional text... [] []"
- "The script element is nested."
- "The noscript element is nested."

<!DOCTYPE html>
<html>
  <head>
```

```
    <meta charset=utf-8>
    <title>Elements within Text defaults for HTML5</title>
  </head>
  <body>
   <p>The element p is not within text. But <em>the element em is</em>.</p>
   <p>A button <button onclick="display()">Click Here</button> is also within tex
But <textarea>The content of textarea</textarea> is nested.</p>
Some additional text...
<script><!--
function display() {
alert("The script element is nested."); }
//-->
</script>
<noscript>The noscript element is nested.</noscript>
  </body>
</html>
```

[Source file: examples/html5/EX-within-text-defaults-html5-1.html]

GLOBAL: The `withinTextRule` element contains the following:

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

Example 47: Specifying elements within text with a `withinTextRule` element

```
<its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
  <its:withinTextRule withinText="yes" selector="//b | //em | //i"/>
</its:rules>
```

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

LOCAL: The following local markup is available for the Elements Within Text data category:

- A `withinText` attribute with the values "yes", "no" or "nested".

Example 48: The Elements Within Text data category expressed locally

```
<text xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <body>
    <par>Text with <bold its:withinText="yes">bold</bold>.</par>
  </body>
</text>
```

[Source file: examples/xml/EX-within-text-local-1.xml]

Example 49: The Elements Within Text data category expressed locally in HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=utf-8>
    <title>Within text test: Default</title>
  </head>
  <body>
    <p>Text with <span its-within-text='yes'>bold</span>.</p>
  </body>
</html>
```

[Source file: examples/html5/EX-within-text-local-html5-1.html]

8.8 Domain

8.8.1 Definition

The Domain data category is used to identify the topic or subject of content. Such information allows for more relevant linguistic choices during various processes.

Examples of usage include:

- Allowing machine translation systems to select the most appropriate engine and rules to translate the content.
- Providing a general indication of what terminology collection is most suitable for use by translators.

This data category addresses various challenges:

- Often domain-related information already exists in the document (e.g., keywords in the HTML `meta` element). The Domain data category provides a mechanism to point to this information.
- There are many flat or structured lists of domain related values, keywords, key phrases, classification codes, ontologies, etc. The Domain data category does not propose its own given list. Instead it provides a mapping mechanism to associate the values in the document with the values used by the consumer tool.

8.8.2 Implementation

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

The information provided by this data category is a comma-separated list of one or more values, which is obtained by applying the following algorithm:

```
STEP 1: Set the initial value of the resulting string as an empty string.
STEP 2: Get the list of nodes resulting of the evaluation of the domainPointer attribute.
STEP 3: For each node:
    STEP 3-1: If the node value contains a COMMA (U+002C):
        STEP 3-1-1: Split the node value into separate strings using the COMMA (U+002C) as separator.
        STEP 3-1-2: For each string:
            STEP 3-1-2-1: Trim the leading and trailing white spaces of the string.
            STEP 3-1-2-2: If the first character of the value is an APOSTROPHE (U+0027) or a QUOTATION MARK (U+0022): Remove it.
```

```
                STEP 3-1-2-3: If the last character of the value is an
                APOSTROPHE (U+0027) or a QUOTATION MARK (U+0022):
                Remove it.
                STEP 3-1-2-4: If the value is empty: Go to STEP 3-1-2.
                STEP 3-1-2-5: Check the domainMapping attribute to see if
                there is a mapping set for the string:
                    STEP 3-1-2-5-1. If a mapping is found: Add the
                    corresponding value to the result string.
                    STEP 3-1-2-5-2. Else (if no mapping is found): Add the
                    string to the result string.
        STEP 3-2: Else (if the node value does not contain a COMMA (U+002C)):
            STEP 3-2-1: Trim the leading and trailing white spaces of the string.
            STEP 3-2-2: If the first character of the value is an APOSTROPHE
            (U+0027) or a QUOTATION MARK (U+0022): Remove it.
            STEP 3-2-3: If the last character of the value is an APOSTROPHE
            (U+0027) or a QUOTATION MARK (U+0022): Remove it.
            STEP 3-2-4: If the value is empty: Go to STEP 3.
            STEP 3-2-5: Check if there is a mapping for the string:
                STEP 3-2-5-1: If a mapping is found: Add the corresponding
                value to the result string.
                STEP 3-2-5-2: Else (if no mapping is found): Add the string (in its
                original cases) to the result string.
    STEP 4: Remove duplicated values from the resulting string.
    STEP 5: Return the resulting string.
```

GLOBAL: The `domainRule` element contains the following:

- A required `selector` attribute. It contains an absolute selector that selects the nodes to which this rule applies.
- A required `domainPointer` attribute that contains a relative selector pointing to a node that contains the domain information.
- An optional `domainMapping` attribute that contains a comma separated list of mappings between values in the content and consumer tool specific values. The left part of the pair corresponds to the source content and is unique within the mapping and case-sensitive. The right part of the mapping belongs to the consumer tool. Several left parts can map to a single right part. The values in the left or the right part of the mapping may contain spaces; in that case they MUST be delimited by quotation marks, that is pairs of APOSTROPHE (U+0027) or QUOTATION MARK (U+0022).

Note:

Although the `domainMapping` attribute it is optional, its usage is recommended. Many commercial machine translation systems use their own domain definitions; the `domainMapping` attribute will foster interoperability between these definitions and metadata items like `keywords` or `dcterms.subject` in Web pages or other types of content.

Values used in the `domainMapping` attribute are arbitrary strings. In some consumer systems or existing content, the domain may be identified via an IRI like `http://example.com/domains/automotive`. The `domainMapping` allows for using IRIs too. For the mapping, they are regarded as ordinary string values.

Note:

Although the focus of ITS 2.0, and some of the usage scenarios addressed in ITS 2.0 High-level Usage Scenarios) is on “single engine” environments, ITS 2.0 (for example in the context of the Domain data category) can accommodate ""workflow/multi engine" scenarios.

Example:

- A scenario involves Machine Translation (MT) engines A and B. The domain labels used by engine A follow the naming scheme A_123, the one for engine B follow the naming scheme B_456.
- A `domainMapping` as follows is in place: domainMapping="'sports law' Legal, 'property law' Legal"
- Engine A maps 'Legal' to A_4711, Engine B maps 'Legal' to B_42.

Thus, ITS does not encode a process or workflow (like "Use MT engine A with domain A_4711, and use MT engine B with domain A_42"). Rather, it encodes information that can be used in workflows.

Example 50: The `domainRule` element

The `domainRule` element expresses that the content of the HTML `body` element is in the domain expressed by the HTML `meta` element with the `name` attribute, value `keywords`. The `domainPointer` attribute points to that `meta` element.

```
<its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0"
           xmlns:h="http://www.w3.org/1999/xhtml">
  <its:domainRule selector="/h:html/h:body"
                  domainPointer="/h:html/h:head/h:meta[@name='keywords']/@content
</its:rules>
```

[Source file: examples/xml/EX-domain-1.xml]

Example 51: The `domainRule` element

The `domainRule` element expresses that the content of the HTML `body` element is in the domain expressed by associated values. The `domainPointer` attribute points to the values in the source content. In this case it points to the `meta` elements with the `name` attribute set to "keywords" or to "dcterms.subject". These elements hold the values in their `content` attributes. The `domainMapping` attribute contains the comma-separated list of mappings. In the example, "automotive" is available in the source content, and "auto" is used within the consumer tool, e.g., a machine translation system.

```
<its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0"
           xmlns:h="http://www.w3.org/1999/xhtml">
  <its:domainRule selector="/h:html/h:body"
    domainPointer="/h:html/h:head/h:meta[@name='dcterms.subject' or @name='keywor
    domainMapping="automotive auto, medical medicine, 'criminal law' law, 'proper
</its:rules>
```

[Source file: examples/xml/EX-domain-2.xml]

Note:

In HTML, one possible way how to express domain information is a `meta` element with the `name` attribute set to "keywords" (see standard metadata names in HTML). Alternatively, following the process for other metadata names the extension value of "dcterms.subject" can be used. The usage of both "keywords" and "dcterms.subject" is shown in example Example 51.

In the area of machine translation (e.g., machine translation systems or systems harvesting content for machine translation training), there is no agreed upon set of value sets for domain. Nevertheless, it is recommended to use a small set of values both in source content and within consumer tools, to foster interoperability. If larger value sets are needed (e.g., detailed terms in the law or medical domain), mappings to the smaller value set needed for interoperability is to be provided. An example would be a `domainMapping` attribute for generalizing the law domain: `domainMapping="'criminal law' law, 'property law' law, 'contract law' law"`.

It is possible to have more than one domain associated with a piece of content. For example, if the consumer tool is a statistical machine translation engine, it could include corpora from all domains available in the source content in training the machine translation engine.

The consumer machine translation engine might choose to ignore the domain and take a one-size-fits-all approach, or may be selective in which domains to use, based on the range of content marked with domain. For example, if the content has hundreds of sentences marked with domain "automotive" and "medical", but only a couple of sentences marked with additional domains "criminal law" and "property law", the consumer tool may opt to include its domains "auto" and "medicine", but not "law", since the extra training resources do not justify the improvement in the output. Guidance about appropriate actions in such cases is beyond the scope of this specification.

8.9 Text Analysis

8.9.1 Definition

The Text Analysis data category is used to annotate content with lexical or conceptual information for the purpose of contextual disambiguation. This information can be provided by so-called text analysis software agents such as named entity recognizers, lexical concept disambiguators, etc., and is represented by either string valued or IRI references to possible resource descriptions. Example: A named entity recognizer provides the information that the string "Dublin" in a certain context denotes a town in Ireland.

While text analysis can be done by humans, this data category is targeted more at software agents.

The information can be used for several purposes, including, but not limited to:

- Informing a human agent such as a translator that a certain fragment of textual content (so-called “text analysis target”) may follow specific translation rules. Examples: proper names, brands, or officially regulated expressions.
- Informing a software agent such as a content management system about the conceptual type of a textual entity to enable special processing. Examples: places, personal names, product names, or geographic names, chemical compounds, and protein names that are situated in a specific index.

The data category provides three pieces of annotation: confidence, entity type or concept class, entity identifier or concept identifier as specified in the following table.

Information	Description	Value	Example	Comments
Text analysis confidence	The confidence of the agent (that produced	The XML Schema double data type with the constraining	0.5647346	The confidence value applies to two pieces of information (see the following rows in this table). This is opposed to termConfidence which is part of the Terminology data

Information	Description	Value	Example	Comments
	the annotation)in its own computation	facets minInclusive set to 0 and maxInclusive set to 1		category. termConfidence represents the confidence in just a single piece of information: the decision whether something is a term or not (term). termConfidence does not relate to the confidence about additional information about the term that can be encoded with `termInfoRef`.
Entity type / concept class	The type of entity, or concept class of the text analysis target	IRI	http://nerd.eurecom.fr/ontology#Location	
Entity / concept identifier	A unique identifier for the text analysis target	Mode 1: Identifier (string value) of the collection source + identifier of the concept in that collection	"Wordnet3.0" to identify the collection resource; "301467919" to identify a synset in Wordnet3.0	Mode 1 and mode 2 are mutually exclusive. They MUST NOT be used at the same time for the same text analysis target/node.
		Mode 2: Identifier (IRI) of the text analysis target	http://dbpedia.org/resource/Dublin	

Note:

The use case for Text Analysis is distinct from that for the Terminology data category. Text Analysis informs human agents or software agents in cases where either explicit terminology information is not (yet) available, or would not be appropriate, e.g. conceptual information for general vocabulary.

Text Analysis support is achieved by associating a fragment of text with an external resource that can be interpreted by a language review agent. The agent may for example use the web resource to disambiguate the meaning or lexical choice of the fragment, and thereby contributing to its correct translation. The web resource may as well provide information on appropriate synonyms and example usage. This is for example the case if the web resource is WordNet [WordNet]. In the case of a concept class, the external resource may provide a formalized conceptual definition arranged in a hierarchical framework of related concepts. In the case of a named entity, the external resource may provide a full-fledged description of the associated real world entity.

Extended example: The word 'City' in the fragment 'I am going to the City' may be enhanced by one of the following:

- one of WordNet's synsets that can be represented by 'city'
- an ontological concept of 'City' that could represent a subclass of 'Populated Place' as a concept
- the central area of a particular city – as interpreted as an entity instance (e.g., 'City of London')

Note:

A given document fragment can only be annotated once. When support for multiple annotations is necessary (e.g., when all three of the annotations in the extended example above need to be accommodated) NIF 2.0, TEI Stand-off Markup, or other so-called stand-off annotation mechanisms is better suitable.

Some external resources such as DBpedia also provide information for some ontological concepts and named entity definitions in multiple languages, and this facilitates translation even more because a possible link traversal would allow a direct access to foreign language labels for named entities.

8.9.2 Implementation

The Text Analysis data category can be expressed with global rules, or locally on an individual element. There is no inheritance.

Note:

This specification defines a normative way to represent text analysis information in XML and HTML locally. However, text analysis information can also be represented in other formats, e.g., JSON. The Internationalization Tag Set Interest Group maintains a description of such alternative serializations. Readers of this specification are encouraged to evaluate whether that description fulfills their needs and to provide comments in the ITS IG mailing list (public archive).

GLOBAL: The `textAnalysisRule` element contains the following:

- A required `selector` attribute that contains an absolute selector that selects the nodes to which this rule applies.
- At least one of the following:
    - A `taClassRefPointer` attribute that contains a relative selector pointing to a node that holds an IRI, which implements the entity type / concept class information.
    - Exactly one of the following:
        - When using identification mode 1: A `taSourcePointer` attribute that contains a relative selector to a node that holds the identifier of the collection source; and a `taIdentPointer` attribute that contains a relative selector to a node that holds the identifier of the concept in the collection.
        - When using identification mode 2: A `taIdentRefPointer` attribute that contains a relative selector pointing to a node that holds an IRI that holds the identifier of the text analysis target.

For an example, see Example 54.

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

- An optional `taConfidence` attribute that implements the text analysis confidence.
- At least one of the following:
    - A `taClassRef` attribute that holds an IRI, which implements the Entity type / concept class information.
    - Exactly one of the following:
        - When using identification mode 1: A `taSource` attribute that holds the identifier of the collection source, and a `taIdent` attribute that holds the identifier of the concept in the collection.

- When using identification mode 2: A taIdentRef attribute that holds the identifier of the text analysis target.

Any node selected by the Text Analysis data category with the taConfidence attribute specified MUST be contained in an element with the annotatorsRef (or in HTML its-annotators-ref) attribute specified for the Text Analysis data category. For more information, see Section 5.7: ITS Tools Annotation.

Example 52: Local mixed usage of taClassRef, and taIdentRef in HTML.

```
<!DOCTYPE html>
<html lang="en" its-annotators-ref="text-analysis|http://enrycher.ijs.si">
  <head>
    <meta charset="utf-8" />
    <title>Text analysis: Local Test</title>
  </head>
  <body>
    <p><span
            its-ta-confidence="0.7"
            its-ta-class-ref="http://nerd.eurecom.fr/ontology#Location"
            its-ta-ident-ref="http://dbpedia.org/resource/Dublin">Dublin</span>
      is the <span
            its-ta-source="Wordnet3.0"
            its-ta-ident="301467919"
            its-ta-confidence="0.5"
            >capital</span> of Ireland.</p>
  </body>
</html>
```

[Source file: examples/html5/EX-text-analysis-html5-local-1.html]

Note:

For expressing Entity type / concept class information, implementers are encouraged to use an existing repository of entity types such as the Named Entity Recognition and Disambiguation [NERD] ontology. Of course this requires that the repository satisfies the constraints imposed by the text analysis data category (e.g., use of IRIs).

Various target types can be expressed via Entity type / concept class: types of entities, types of lexical concepts, or ontology concepts. While a relationship between these types may exist, this specification does not prescribe a way of automatically inferring a one target type from another.

Note:

Text Analysis is primarily intended for textual content. Nevertheless, the data category can also be used in multimedia contexts. Example: objects on an image could be annotated with DBpedia IRIs.

When serializing the Text Analysis data category markup in HTML, one way to serialize the markup is RDFa Lite or Microdata. This serialization is due to the existing search and crawling infrastructure that is able to consume these formats. For other usage scenarios (e.g., adding text annotation to feed into a subsequent terminology process), using native ITS Text Analysis data category markup is preferred. In this way, the markup easily can be stripped out again later.

Example 53: Local mixed usage of taClassRefPointer, and taIdentRefPointer, in HTML+RDFa Lite.

See Example 54 for the companion document with the mapping data.

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
      <link href=EX-text-analysis-html5-rdfa.xml rel=its-rules>
        <title>Entity: Local Test</title>
  </head>
  <body>
    <p><span property="http://xmlns.com/foaf/0.1/name"
      about="http://dbpedia.org/resource/Dublin"
      typeof="http:/nerd.eurecom.fr/ontology#Location">Dublin</span> is
      the capital of Ireland.</p>
  </body>
</html>
```

[Source file: examples/html5/EX-text-analysis-html5-rdfa.html]

Example 54: Companion document, having the mapping data for Example 53.

```
<its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
  <its:textAnalysisRule selector="//*[@typeof and @about]"
    taClassRefPointer="@typeof" taIdentRefPointer="@about"/>
</its:rules>
```

[Source file: examples/html5/EX-text-analysis-html5-rdfa.xml]

8.10 Locale Filter

8.10.1 Definition

The Locale Filter data category specifies that a node is only applicable to certain locales.

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

- Including a legal notice only in locales for certain regions.
- Dropping editorial notes from all localized output.

The Locale Filter data category associates with each selected node a filter type and a list of extended language ranges conforming to [BCP47].

The list is comma-separated and can include the wildcard extended language range "*". The list can also be empty. Whitespace surrounding language ranges is ignored.

The type can take the values "include" or "exclude":

- A single wildcard "*" with a type "include" indicates that the selected content applies to all locales.
- A single wildcard "*" with a type "exclude" indicates that the selected content applies to no locale.
- An empty string with a type "include" indicates that the selected content applies to no locale.
- An empty string with a type "exclude" indicates that the selected content applies to all locales.

- Otherwise, with a type "include", the selected content applies to the locales for which the language tag has a match in the list when using the Extended Filtering algorithm defined in [BCP47].
- If, instead, the type is "exclude", the selected content applies to the locales for which the language tag does not have a match in the list when using the Extended Filtering algorithm defined in [BCP47].

8.10.2 Implementation

The Locale Filter data category can be expressed with global rules, or locally on an individual element. For elements, the data category information inherits to the textual content of the element, *including* child elements and attributes. The default is that the language range is "*" and the type is "include".

GLOBAL: The `localeFilterRule` element contains the following:

- A required `selector` attribute. It contains an absolute selector that selects the nodes to which this rule applies.
- A required `localeFilterList` attribute with a comma-separated list of extended language ranges, or an empty string value.
- An optional `localeFilterType` attribute with a value "include" or "exclude".

Example 55: The Locale Filter data category expressed globally

This document contain three `localeFilterRule` elements: The first one specifies that the elements `legalnotice` with a `role` set to "Canada" apply only to the Canadian locales. The second one specifies that the elements `legalnotice` with a `role` set to "nonCanada" apply to all locales that are not Canadian. And the third one specifies that none of the `remark` elements apply to any locale.

```
<book xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <info>
    <its:rules version="2.0">
      <its:localeFilterRule selector="//legalnotice[@role='Canada']"
        localeFilterList="*-CA"/>
      <its:localeFilterRule selector="//legalnotice[@role='nonCanada']"
        localeFilterList="*-CA" localeFilterType="exclude"/>
      <its:localeFilterRule selector="//remark"
        localeFilterList=""/>
    </its:rules>
    <legalnotice role="Canada">
      <para>This notice is only for Canadian locales.</para>
    </legalnotice>
    <legalnotice role="nonCanada">
      <para>This notice is for locales that are non-Canadian locales.</para>
    </legalnotice>
    <remark>Note: This section will be written later.</remark>
  </info>
</book>
```

[Source file: examples/xml/EX-locale-filter-selector-1.xml]

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

- A `localeFilterList` attribute with a comma-separated list of extended language ranges, or an empty string value.

- An optional localeFilterType attribute with a value "include" or "exclude".

Example 56: The Locale Filter data category expressed locally in HTML

In this example the Locale Filter data category is used to select different sections depending on whether the locale is a Canadian one or not.

```
<!DOCTYPE html>
<html>
 <head>
  <meta charset=utf-8>
  <title>Locale filter</title>
 </head>
 <body>
  <div its-locale-filter-list="*-ca">
   <p>Text for Canadian locales.</p>
  </div>
  <div its-locale-filter-list="*-ca" its-locale-filter-type="exclude">
   <p>Text for non-Canadian locales.</p>
  </div>
 </body>
</html>
```

[Source file: examples/html5/EX-locale-filter-local-html5-1.html]

Example 57: The Locale Filter data category expressed locally in XML

```
<book xmlns:its="http://www.w3.org/2005/11/its">
  <info>
    <legalnotice its:localeFilterList="en-CA, fr-CA">
      <para>This legal notice is only for English and French Canadian locales.</p
    </legalnotice>
  </info>
</book>
```

[Source file: examples/xml/EX-locale-filter-attribute-1.xml]

8.11 Provenance

8.11.1 Definition

The Provenance data category is used to communicate the identity of agents that have been involved in the translation of the content or the revision of the translated content. This allows translation and translation revision consumers, such as post-editors, translation quality reviewers, or localization workflow managers, to assess how the performance of these agents may impact the quality of the translation. Translation and translation revision agents can be identified as a person, a piece of software or an organization that has been involved in providing a translation that resulted in the selected content.

This data category offers three types of information. First, it allows identification of translation agents. Second, it allows identification of revision agents. Third, if provenance information is needed that includes temporal or sequence information about translation processes (e.g. multiple revision cycles) or requires agents that support a wider range of activities, the data category offers a mechanism to refer to external provenance information.

Note:

The specification does not define the format of external provenance information, but it is recommended that an open provenance or change-logging format be used, e.g. the W3C provenance data model [PROV-DM].

Translation or translation revision tools, such as machine translation engines or computer assisted translation tools, may offer an easy way to create this information. Translation tools can then present this information to post-editors or translation workflow managers. Web applications may to present such information to consumers of translated documents.

The data category defines seven pieces of information:

Information	Description	Value
Human provenance information	Identification of a human translation agent	A string or an IRI (only for the `Ref` attributes)
Organizational provenance information	Identification of an organization acting as a translation agent	A string or an IRI (only for the `Ref` attributes)
Tool-related provenance information	Identification of a software tool that was used in translating the selected content	A string or an IRI (only for the `Ref` attributes)
Human revision provenance information	Identification of a human translation revision agent	A string or an IRI (only for the `Ref` attributes)
Organizational revision provenance information	Identification of an organization acting as a translation revision agent	A string or an IRI (only for the `Ref` attributes)
Tool-related revision provenance information	Identification of a software tool that was used in revising the translation of the selected content	A string or an IRI (only for the `Ref` attributes)
Reference to external provenance information	A reference to external provenance information	A space (U+0020) separated list of IRIs

Note:

The tool related provenance and tool related revision provenance pieces of information are not meant to express information about tools used for creating ITS annotations themselves. For this purpose, ITS 2.0 provides a separate mechanism. See Section 5.7: ITS Tools Annotation for details, especially the note on annotatorsRef usage scenarios.

8.11.2 Implementation

The Provenance data category can be expressed with global rules, or locally on individual elements. For elements, the data category information inherits to the textual content of the element, *including* child elements and attributes.

GLOBAL: The `provRule` element contains the following:

- A required `selector` attribute. It contains an absolute selector that selects the nodes to which this rule applies.
- A `provenanceRecordsRefPointer` attribute that contains a relative selector pointing to a node containing a list of provenance records. These are related to the content selected via the `selector` attribute.

Example 58: The Provenance data category used globally with standoff provenance records.

This example expresses provenance information in a standoff manner using `provenanceRecords` elements. The `provRule` element specifies that for any element with a `ref` attribute that `ref` attribute holds a reference to an associated `provenanceRecords` element where the provenance information is listed. The `legalnotice` element has been revised two times. Hence, the related provenanceRecords element contains two provenanceRecord child elements.

```
<text xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <dc:creator>John Doe</dc:creator>
  <its:provenanceRecords xml:id="pr1">
    <its:provenanceRecord
      toolRef="http://www.example.onlinemtex.com/2012/7/25/wsdl/"
      org="acme-CAT-v2.3"
      revToolRef="http://www.mycat.com/v1.0/download"
      revOrg="acme-CAT-v2.3"
      provRef="http://www.examplelsp.com/excontent987/production/prov/e6354"/>
  </its:provenanceRecords>
  <its:provenanceRecords xml:id="pr2">
    <its:provenanceRecord
      person="John Doe"
      orgRef="http://www.legaltrans-ex.com"
      revPerson="Tommy Atkins"
      revOrgRef="http://www.example.myorg.com"
      provRef="http://www.example.myorg.com/job-12-7-15-X31/reviewed/prov/re85734
    <its:provenanceRecord
      revPerson="John Smith"
      revOrgRef="http://john-smith.qa.example.com"/>
  </its:provenanceRecords>
  <its:rules version="2.0">
    <its:provRule selector="//*[@ref]" provenanceRecordsRefPointer="@ref"/>
  </its:rules>
  <title>Translation Revision Provenance Agent: Global Test in XML</title>
  <body>
    <par ref="#pr1"> This paragraph was translated from the machine.</par>
    <legalnotice postediting-by="http://www.example.myorg.com" ref="#pr2">This te
      translated directly by a person.</legalnotice>
  </body>
</text>
```

[Source file: examples/xml/EX-provenance-global-1.xml]

LOCAL: Using the inline markup to represent the data category locally is limited to a single occurrence for a given content (e.g., one cannot have different toolRef attributes applied to the same span of text because the inner-most one would override the others). A local *standoff markup* is provided to allow such cases.

The following local markup is available for the Provenance data category:

- Either (inline markup): at least one of the following attributes:
  - A person or personRef attribute that implements the human provenance information.
  - An org or orgRef attribute that implements the organizational provenance information.
  - A tool or toolRef attribute that implements the tool-related provenance information.
  - A revPerson or revPersonRef attribute that implements the human revision provenance information.

    - A revOrg or revOrgRef attribute that implements the organizational revision provenance information.
    - A revTool or revToolRef attribute that implements the tool-related revision provenance information.
    - A provRef attribute that implements the reference to external provenance descriptions.
- Or (standoff markup):
    - A provenanceRecordsRef attribute. Its value is an IRI pointing to the provenanceRecords element containing the list of provenance records related to this content.
    - An element provenanceRecords, which contains:
        - One or more elements provenanceRecord, each of which contains at least one of the following attributes:
            - A person or personRef attribute that implements the human provenance information.
            - An org or orgRef attribute that implements the organizational provenance information.
            - A tool or toolRef attribute that implements the tool-related provenance information.
            - A revPerson or revPersonRef attribute that implements the human revision provenance information.
            - A revOrg or revOrgRef attribute that implements the organizational revision provenance information.
            - A revTool or revToolRef attribute that implements the tool-related revision provenance information.
            - A provRef attribute that implements the reference to external provenance descriptions.

Note:

Ideally the order of provenanceRecord elements within a provenanceRecords element reflects the order with which they were added to the document, with the most recently added one listed first.

When the attributes person, personRef, org, orgRef, tool, toolRef, revPerson, revPersonRef, revOrg, revOrgRef, revTool, revToolRef and provRef are used in a standoff manner, the information they carry pertains to the content of the element that refers to the standoff annotation, not to the content of the element provenanceRecord where they are declared.

In HTML the standoff markup MUST either be stored inside a script element in the same HTML document, or be linked from any provenanceRecordsRef to an external XML or HTML file with the standoff inside. If standoff is inside a script element that element MUST have a type attribute with the value application/its+xml. Its id attribute MUST be set to the same value as the xml:id attribute of the provenanceRecords element it contains.

Example 59: Annotating provenance information in XML with local inline markup

The provenance related attributes at the par and legalnotice elements are used to associate the provenance information directly with the content of these elements.

```
<text xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <title>Translation Revision Provenance Agent: Local Test in XML</title>
  <body>
    <par its:toolRef="http://www.onlinemtex.com/2012/7/25/wsdl/"
```

```
      its:org="acme-CAT-v2.3"
      its:revToolRef="http://www.mycat.com/v1.0/download"
      its:revOrg="acme-CAT-v2.3"
      its:provRef="http://www.example.lsp1.com/prov/e6354 http://www.example.lsp2
      >This paragraph was translated from the machine.</par>
    <legalnotice its:person="John Doe"
      its:orgRef="http://www.legaltrans-ex.com/"
      its:provRef="http://www.examplelsp.com/excontent987/legal/prov/e6354"
      its:revPerson="Tommy Atkins"
      its:revOrgRef="http://www.example.myorg.com"
      >This text was translated directly by a person.</legalnotice>
  </body>
</text>
```

[Source file: examples/xml/EX-provenance-local-1.xml]

Example 60: Annotating provenance information in HTML with local inline markup

In this example several spans of content are associated with provenance information.

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <title>Provenance Agent: Local Test in HTML5</title>
  </head>
  <body>
    <p its-tool-ref="http://www.onlinemtex.com/2012/7/25/wsdl/"
      its-org="acme-CAT-v2.3"
      its-prov-ref="http://www.examplelsp.com/excontent987/production/prov/e6354"
      its-rev-org="acme-CAT-v2.3"
      >This paragraph was translated from the machine.</p>
    <p class="legal-notice"
     its-person="John Doe"
     its-org-ref="http://www.legaltrans-ex.com/"
     its-prov-ref="http://www.examplelsp.com/excontent987/legal/prov/e6354"
     its-rev-person="Tommy Atkins" its-rev-org-ref="http://www.example.myorg.com"
     >This text was translated directly by a person.</p>
  </body>
</html>
```

[Source file: examples/html5/EX-provenance-html5-local-1.html]

Example 61: Annotating provenance information in HTML with local standoff markup

The following example shows a document using local standoff markup to encode provenance information. The `p` elements delimit the content to markup. They hold `its-provenance-records-ref` attributes that point to the standoff information inside the `script` elements.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=utf-8>
    <title>Test</title>
    <script id=pr1 type=application/its+xml>
      <its:provenanceRecords xml:id="pr1" xmlns:its="http://www.w3.org/2005/11/it
```

```
        <its:provenanceRecord
         toolRef="http://www.onlinemtex.com/2012/7/25/wsdl/"
         org="acme-CAT-v2.3"
         provRef="http://www.examplelsp.com/excontent987/production/prov/e6354"
         revToolRef="http://www.mycat.com/v1.0/download"
         revOrg="acme-CAT-v2.3" />
      </its:provenanceRecords>
    </script>
    <script id=pr2 type=application/its+xml>
      <its:provenanceRecords xml:id="pr2" xmlns:its="http://www.w3.org/2005/11/it
        <its:provenanceRecord
         person="John Doe"
         orgRef="http://www.legaltrans-ex.com/"
         provRef="http://www.examplelsp.com/excontent987/legal/prov/e6354"
         revPerson="Tommy Atkins"
         revOrgRef="http://www.example.myorg.com" />
        <its:provenanceRecord
         revPerson="John Smith"
         revOrgRef="http://john-smith.qa.example.com" />
        </its:provenanceRecords>
      </script>
  </head>
  <body>
    <p its-provenance-records-ref="#pr1">This paragraph was translated from the m
    <p its-provenance-records-ref="#pr2">This text was translated directly by a p
  </body>
</html>
```

[Source file: examples/html5/EX-provenance-html5-local-2.html]

8.12 External Resource

8.12.1 Definition

The External Resource data category indicates that a node represents or references potentially translatable data in a resource outside the document. Examples of such resources are external images and audio or video files.

8.12.2 Implementation

The External Resource data category can be expressed only with global rules. There is no inheritance. There is no default.

GLOBAL: The `externalResourceRefRule` element contains the following:

- A required `selector` attribute. It contains an absolute selector that selects the nodes to which this rule applies.
- A required `externalResourceRefPointer` attribute that contains a relative selector pointing to a node that provides the IRI of the external resource.

Example 62: The `externalResourceRefRule` element

The `externalResourceRefRule` element expresses that the `imagedata`, `audiodata` and `videodata` elements contain references to external resources. These references are expressed via a fileref attribute. The externalResourceRefPointer attribute points to that attribute.

```
<doc xmlns:its="http://www.w3.org/2005/11/its"
  xmlns:db="http://docbook.org/ns/docbook">
  <its:rules version="2.0">
    <its:externalResourceRefRule
      selector="//db:imagedata | //db:audiodata |  //db:videodata"
      externalResourceRefPointer="@fileref"/>
  </its:rules>
  <db:mediaobject>
    <db:videoobject>
      <db:videodata fileref="movie.avi"/>
    </db:videoobject>
    <db:imageobject>
      <db:imagedata fileref="movie-frame.gif"/>
    </db:imageobject>
    <db:textobject>
      <db:para>This video illustrates the proper way to assemble an inverting
        time distortion device. </db:para>
      <db:warning>
        <db:para> It is imperative that the primary and secondary temporal
          couplings not be mounted in the wrong order. Temporal catastrophe is
          the likely result. The future you destroy may be your own. </db:para>
      </db:warning>
    </db:textobject>
  </db:mediaobject>
</doc>
```

[Source file: examples/xml/EX-externalresource-1.xml]

Example 63: Two externalResourceRefRule elements used for external resources associated with HTML video elements

The two externalResourceRefRule elements select the src and the poster attributes at HTML video elements. These attributes identify different external resources, and at the same time contain the references to these resources. For this reason, the externalResourceRefPointer attributes point to the value of src and poster respectively. The underlying HTML document is given in Example 64.

```
<its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its"
  xmlns:html="http://www.w3.org/1999/xhtml">
  <its:externalResourceRefRule selector="//html:video/@src"
    externalResourceRefPointer="."/>
  <its:externalResourceRefRule selector="//html:video/@poster"
    externalResourceRefPointer="."/>
</its:rules>
```

[Source file: examples/xml/EX-externalresource-2.xml]

Example 64: An HTML document that can be used for Example 63.

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <title>Video element example</title>
  </head>
  <body>
    <video
      height=360
      poster=http://www.example.com/video-image.png
      src=http://www.example.com/video/v2.mp
      width=640>
      <p>If your browser doesn't support
        the <code>video</code> element, you can
        <a href=http://www.example.com/video/v2.mp>download the video</a> instea
    </video>
  </body>
</html>
```

[Source file: examples/html5/EX-externalresource-html5-1.html]

8.13 Target Pointer

8.13.1 Definition

Some formats, such as those designed for localization or for multilingual resources, hold the same content in different languages inside a single document. The Target Pointer data category is used to associate the node of a given source content (i.e., the content to be translated) and the node of its corresponding target content (i.e., the source content translated into a given target language).

This specification makes no provision regarding the presence of the target nodes or their content: A target node may or may not exist and it may or may not have content.

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

- Extract the source content to translate and put back the translation at its proper location.
- Compare source and target content for quality verification.
- Reuse existing translations when localizing the new version of an existing document.
- Access aligned bi-lingual content to build memories, or to train machine translation engines.

Note:

In general, it is recommended to avoid developing formats where the same content is stored in different languages in the same document, except for very specific use cases. See the best practices “Working with multilingual documents” from [XML i18n BP] for further guidance.

8.13.2 Implementation

The Target Pointer data category can be expressed only with global rules. There is no inheritance. There is no default.

GLOBAL: The `targetPointerRule` element contains the following:

- A required `selector` attribute. It contains an absolute selector that selects the nodes to which this rule applies.
- A required `targetPointer` attribute. It contains a relative selector that points to the node for the target content corresponding to the selected source node.

Note:

The source node and the target node may be of different types, but the target node has to contain the same content as the source node (e.g., an attribute node cannot be the target node of a source node that is an element with children).

Example 65: Defining the target location of a source content with the `targetPointerRule` element

```
<file>
  <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
    <its:translateRule selector="/file" translate="no"/>
    <its:translateRule selector="//source" translate="yes"/>
    <its:targetPointerRule selector="//source" targetPointer="../target"/>
  </its:rules>
  <entry id="one">
    <source>Remember last folder</source>
    <target/>
  </entry>
  <entry id="two">
    <source>Custom file filter:</source>
    <target/>
  </entry>
</file>
```

[Source file: examples/xml/EX-target-pointer-global-1.xml]

8.14 ID Value

8.14.1 Definition

The ID Value data category indicates a value that can be used as unique identifier for a given part of the content.

The recommended way to specify a unique identifier is to use `xml:id` [XML ID] or `id` in HTML (See the best practice “Defining markup for unique identifiers” from [XML i18n BP]). The `idValueRule` element is intended only as a fall-back mechanism for documents in which unique identifiers are available with another construct.

Providing a unique identifier that is maintained in the original document can be useful for several purposes, for example:

- Allow automated alignment between different versions of the source document, or between source and translated documents.
- Improve the confidence in leveraged translation for exact matches.
- Provide backtracking information between displayed text and source material when testing or debugging.

Note:

- The ID Value data category only provides for rules to be expressed at a global level. Locally, users are able to use `xml:id` (which is defined by XML) or `id` in HTML, or an attribute specific to the format in question (as in Example 68).
- Applying the ID Value data category to `xml:id` (in XML) or `id` (in HTML) attributes in global rules is not necessary, since these attributes are the recommended way to specify an identifier.

8.14.2 Implementation

The ID Value data category can be expressed only with global rules. There is no inheritance. There is no default.

GLOBAL: The `idValueRule` element contains the following:

- A required `selector` attribute. It contains an absolute selector that selects the nodes to which this rule applies.
- A required `idValue` attribute. It contains any XPath expression; the context for the evaluation of the XPath expression is the same as for relative selectors. The evaluation of the XPath expression constructs a string corresponding to the identifier of the node to which this rule applies. The identifier MUST be unique at least within the document. If the attribute `xml:id` is present or `id` in HTML for the selected node, the value of the `xml:id` attribute or `id` in HTML MUST take precedence over the `idValue` value.

Example 66: Pointing to an ID Value with the `idValueRule` element

The `idValueRule` element indicates that the unique identifier for each `<text>` element is the value of the attribute `name` of its parent element.

```
<?xml version="1.0"?>
<resources>
  <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
    <its:translateRule translate="no" selector="/resources"/>
    <its:translateRule translate="yes" selector="//text"/>
    <its:idValueRule selector="//text" idValue="../@name"/>
  </its:rules>
  <entry name="btn.OK">
    <text>OK</text>
    <pos>1, 1</pos>
    <trig>sendOK</trig>
  </entry>
  <entry name="btn.CANCEL">
    <text>Cancel</text>
    <pos>2, 1</pos>
    <trig>cancelAll</trig>
  </entry>
</resources>
```

[Source file: examples/xml/EX-idvalue-element-1.xml]

Example 67: Constructing ID values using the `idValueRule` element.

The `idValue` attribute allows to build composite values based on different attributes, elements, or even hard-coded text. Any of the String functions offered by XPath can be used. In the document below, the two elements `<text>` and `<desc>` are translatable, but they have only one corresponding identifier, the `name` attribute in their parent element.

To make sure the identifier is unique for both the content of `<text>` and the content of `<desc>`, the XPath expression `concat(../@name, '_t')` gives the identifier "settingsMissing_t" for the content of `<text>` and the expression `concat(../@name, '_d')` gives the identifier "settingsMissing_d" for the content of `<desc>`.

```
<?xml version="1.0"?>
<doc>
```

```
  <its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
    <its:idValueRule selector="//text" idValue="concat(../@name, '_t')"/>
    <its:idValueRule selector="//desc" idValue="concat(../@name, '_d')"/>
  </its:rules>
  <msg name="settingsMissing">
    <text>Can't find settings file.</text>
    <desc>The module cannot find the default settings file. You need to
      re-initialize the system.</desc>
  </msg>
</doc>
```

[Source file: examples/xml/EX-idvalue-element-2.xml]

Example 68: Using `xml:id` and `idValueRule`

When an `xml:id` attribute is present for a node selected by an `idValueRule` element, the value of `xml:id` takes precedence over the value defined by the `idValueRule` element. In the example below, the unique ID to use is “btnAgain” for the first `<res>` element, and “retryTip” for the second `<res>` element.

```
<?xml version="1.0"?>
<file>
  <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
    <its:idValueRule selector="//res" idValue="@name"/>
  </its:rules>
  <res name="retryBtn" xml:id="btnAgain">Try Again</res>
  <res name="retryTip">click this to re-run the process with the current
    settings.</res>
</file>
```

[Source file: examples/xml/EX-idvalue-attribute-1.xml]

8.15 Preserve Space

8.15.1 Definition

The Preserve Space data category indicates how whitespace is to be handled in content. The possible values for this data category are "default" and "preserve" and carry the same meaning as the corresponding values of the xml:space attribute. The default value is "default". The Preserve Space data category does not apply to HTML documents in HTML syntax.

8.15.2 Implementation

The Preserve Space data category can be expressed with global rules, or locally using the `xml:space` attribute. For elements, the data category information inherits to the textual content of the element, *including* child elements and attributes.

> **Note:**
>
> The Preserve Space data category is not applicable to HTML documents in HTML syntax because `xml:space` (and by extension Preserve Space) has no effect in documents parsed as text/html. However, the data category can be used in HTML *in XHTML syntax*.

GLOBAL: The `preserveSpaceRule` element contains the following:

- A required selector attribute. It contains an absolute selector that selects the nodes to which this rule applies.
- A required space attribute with the value "default" or "preserve".

Example 69: The Preserve Space data category expressed globally

The preserveSpaceRule element specifies that whitespace in all verse elements are to be treated literally.

```
<book>
 <info>
  <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
   <its:preserveSpaceRule selector="//verse" space="preserve"/>
  </its:rules>
 </info>
 <verse>
’Twas brillig, and the slithy toves
 Did gyre and gimble in the wabe;
All mimsy were the borogoves,
 And the mome raths outgrabe.
 </verse>
</book>
```

[Source file: examples/xml/EX-preservespace-global-1.xml]

LOCAL: The xml:space attribute, as defined in section 2.10 of [XML 1.0], maps exactly to the Preserve Space data category.

Example 70: The Preserve Space data category expressed locally

The standard xml:space attribute specifies that the whitespace in the verse element are to be treated literally.

```
<book>
 <verse xml:space="preserve">
'Twas brillig, and the slithy toves
 Did gyre and gimble in the wabe;
All mimsy were the borogoves,
 And the mome raths outgrabe.
</verse>
</book>
```

[Source file: examples/xml/EX-preservespace-local-1.xml]

8.16 Localization Quality Issue

8.16.1 Definition

The Localization Quality Issue data category is used to express information related to localization quality assessment tasks. Such tasks can be conducted on the translation of some source content (such as a text or an image) into a target language or on the source content itself where its quality may impact on the localization process.

Note:

Automated or manual quality assessment is one area of quality management for translation and localization. An example of existing quality assessment is in-country review (e.g., as part of a language acceptance test for software). An important part of quality assessment is the list of issue types that are being used. Very often, simple issue categories like "correct/incorrect" or "like/dislike" are inadequate; instead, more specific ones such as "terminology" or "grammar" are more helpful in identifying concrete reasons for quality problems and for obtaining a more objective picture of quality levels.

Non-normative terminology related to localization quality as used in this section is provided in Appendix H: Localization Quality Guidance.

This data category can be used in a number of ways, including the following example scenarios:

- A human reviewer working with a web-based tool adds quality markup manually in a text editor, including comments and suggestions, to localized content as part of the review process. A subsequent process examines this markup to ensure that changes were made.
- A fully automatic quality checking tool flags a number of potential quality issues in an XML or HTML file and marks them up using ITS 2.0 markup. A human reviewer then uses another tool to examine this markup and decide whether the file needs to receive more extensive review or be passed on for further processing without a further manual review stage.
- A quality assessment process identifies a number of issues and adds the ITS markup to a rendered HTML preview of an XML file along with CSS styling that highlights these issues. The resulting HTML file is then sent back to the translator to assist his or her revision efforts.

Note:

What issues should be considered in quality assessment tasks depends on the nature of the project and tools used. Further guidance is beyond the scope of this specification, but implementers may wish to consult the references cited in Appendix H: Localization Quality Guidance.

The data category defines five pieces of information:

Information	Description	Value	Notes
Type	A classifier that groups similar issues into categories (for example to differentiate spelling errors from grammar errors).	One of the values defined in list of type values.	ITS 2.0-compliant tools that use these types MUST map their internal values to these types. If the type of the issue is set to `uncategorized`, a comment MUST be specified as well.
Comment	A human-readable description of a specific instance of a quality issue.	Text	Comments can be used to explain an issue or provide guidance in addressing an issue. For example, a note about a Terminology issue might specify what term should be used.
Severity	A classifier for the seriousness of an issue. The seriousness depends on the Quality Model that is being applied. The Quality Model should be made	A rational number in the interval 0 to 100 (inclusive). The value follows the XML Schema double data type with the constraining facets minInclusive set to 0 and maxInclusive set to 100.	It is up to tools to map the values allowed by ITS 2.0 to their own system’s scale. If needed, the original value can be passed along using a custom namespace for XML, or a `data-` attribute for HTML.

Information	Description	Value	Notes
	explicit via the Profile Reference.	The higher values represent greater severity.	
Profile Reference	A reference to a description of the quality assessment model (or a specific profile (customization/ instantiation) of a model, where relevant) used for the issue.	An IRI pointing to the reference document.	The use of resolvable IRIs is strongly recommended as it provides a way for human evaluators to learn more about the quality issues in use.
Enabled	A flag indicating whether the issue is enabled or not.	A value `yes` or `no`, with the default value being `yes`.	This flag is used to activate or deactivate issues. There is no prescribed behavior associated with activated or deactivated issues. One example of usage is a tool that allows the user to deactivate false positives so they are not displayed again each time the document is re-checked.

8.16.2 Implementation

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

GLOBAL: The `locQualityIssueRule` element contains the following:

- A required `selector` attribute. It contains an absolute selector that selects the nodes to which this rule applies.
- Either (in parallel to local inline markup)
  - At least one of the following attributes:
    - A `locQualityIssueType` attribute that implements the type information.
    - A `locQualityIssueComment` attribute that implements the comment information.
  - An optional `locQualityIssueSeverity` attribute that implements the severity information.
  - An optional `locQualityIssueProfileRef` attribute that implements the profile reference information.
  - An optional `locQualityIssueEnabled` attribute that implements the enabled information.
- Or (standoff markup) exactly one of the following:
  - A `locQualityIssuesRef` attribute. Its value is an IRI pointing to the `locQualityIssues` element containing the list of issues related to this content.
  - A `locQualityIssuesRefPointer` attribute that contains a relative selector pointing to a node with the exact same semantics as `locQualityIssuesRef`.

Note:

The attribute locQualityIssuesRefPointer does not apply to HTML as local markup is provided for direct annotation in HTML.

Example 71: Annotating an issue in XML with locQualityIssueRule element

The locQualityIssueRule element associates the issue information with the value of the text attribute.

```
<?xml version="1.0"?>
<doc>
  <header>
    <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
      <its:locQualityIssueRule selector="//image[@id='i1']/@text"
        locQualityIssueType="typographical"
        locQualityIssueComment="Sentence without capitalization"
        locQualityIssueSeverity="50"/>
    </its:rules>
  </header>
  <para>Click the button <image id="i1" src="button.png"
                          text="start button"/>.</para>
</doc>
```

[Source file: examples/xml/EX-locQualityIssue-global-1.xml]

Example 72: Annotating an issue in XML with local standoff markup and a global rule

The following example shows a document using local standoff markup to encode several issues. But because, in this case, the mrk element does not allow attributes from another namespace we cannot use locQualityIssuesRef directly. Instead, a global rule is used to map the function of locQualityIssuesRef to a non-ITS construct, here the ref attribute of any mrk elements that have their attribute type set to "x-itslq".

```
<?xml version="1.0"?>
<doc xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <file>
    <header>
      <its:rules version="2.0">
        <its:locQualityIssueRule selector="//mrk[@type='x-itslq']"
          locQualityIssuesRefPointer="@ref"/>
      </its:rules>
    </header>
    <unit id="1">
      <segment>
        <source>This is the content</source>
        <target><mrk type="x-itslq" ref="#lq1">c'es</mrk> le contenu</target>
      </segment>
      <its:locQualityIssues xml:id="lq1">
        <its:locQualityIssue locQualityIssueType="misspelling"
          locQualityIssueComment="'c'es' is unknown. Could be 'c'est'"
          locQualityIssueSeverity="50"/>
        <its:locQualityIssue locQualityIssueType="typographical"
          locQualityIssueComment="Sentence without capitalization"
          locQualityIssueSeverity="30"/>
      </its:locQualityIssues>
    </unit>
  </file>
</doc>
```

W3C Recommendation

[Source file: examples/xml/EX-locQualityIssue-global-2.xml]

LOCAL: Using inline markup to represent the data category locally is limited to a single occurrence for a given content (e.g. one cannot have different `locQualityIssueType` attributes applied to the same span of text because the inner-most one would override the others). A local *standoff markup* is provided to allow such cases.

The following local markup is available for the Localization Quality Issue data category:

- Either (inline markup):
    - At least one of the following attributes:
        - A `locQualityIssueType` attribute that implements the type information.
        - A `locQualityIssueComment` attribute that implements the comment information.
    - An optional `locQualityIssueSeverity` attribute that implements the severity information.
    - An optional `locQualityIssueProfileRef` attribute that implements the profile reference information.
    - An optional `locQualityIssueEnabled` attribute that implements the enabled information.
- Or (standoff markup):
    - A `locQualityIssuesRef` attribute. Its value is an IRI pointing to the `locQualityIssues` element containing the list of issues related to this content.
    - An element `locQualityIssues` with a `xml:id` attribute set to the identifier specified in the `locQualityIssuesRef` attribute. The `locQualityIssues` element contains:
        - One or more elements `locQualityIssue`, each of which contains:
            - At least one of the following attributes:
                - A `locQualityIssueType` attribute that implements the type information.
                - A `locQualityIssueComment` attribute that implements the comment information.
            - An optional `locQualityIssueSeverity` attribute that implements the severity information.
            - An optional `locQualityIssueProfileRef` attribute that implements the profile reference information.
            - An optional `locQualityIssueEnabled` attribute that implements the enabled information.

    Note:

    Ideally the order of `locQualityIssue` elements within a `locQualityIssues` element reflects the order with which they were added to the document, with the most recently added one listed first.

    When the attributes `locQualityIssueType`, `locQualityIssueComment`, `locQualityIssueSeverity`, `locQualityIssueProfileRef` and `locQualityIssueEnabled` are used in a standoff manner, the information they carry pertains to the content of the element that refers to the standoff annotation, not to the content of the element `locQualityIssue` where they are declared.

    In HTML the standoff markup MUST either be stored inside a `script` element in the same HTML document, or can be linked from any `locQualityIssuesRef` to an external XML or

HTML file with the standoff inside. If standoff is inside a `script` element, that element MUST have a `type` attribute with the value `application/its+xml`. Its `id` attribute MUST be set to the same value as the `xml:id` attribute of the `locQualityIssues` element it contains.

Example 73: Annotating an issue in XML with local inline markup

The attributes `locQualityIssueType`, `locQualityIssueComment` and `locQualityIssueSeverity` are used to associate the issue information directly with a selected span of content.

```
<?xml version="1.0"?>
<doc xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <para><span its:locQualityIssueType="typographical"
      its:locQualityIssueComment="Sentence without capitalization"
      its:locQualityIssueSeverity="50">this</span> is an example</para>
</doc>
```

[Source file: examples/xml/EX-locQualityIssue-local-1.xml]

Example 74: Annotating an issue in HTML with local inline markup

In this example several spans of content are associated with a quality issue.

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <title>Telharmonium 1897</title>
    <style type=text/css>
         [its-loc-quality-issue-type]{
           background-color:yellow;
           margin:2px;
         }
         [its-loc-quality-issue-severity = "100"]{
           border: 2px solid red;
         }
        </style>
  </head>
  <body>
    <h1>Telharmonium (1897)</h1>
    <p>
      <span
        data-mytool-qacode=named_entity_not_found
        its-loc-quality-issue-comment="Should be Thomas Cahill."
        its-loc-quality-issue-profile-ref=http://example.org/qaMovel/v1
        its-loc-quality-issue-severity=100
        its-loc-quality-issue-type=inconsistent-entities>Christian Bale</span>
      (1867–1934) conceived of an instrument that could transmit its sound
      from a power plant for hundreds of miles to listeners over telegraph wiring
      Beginning in 1889 the sound quality of regular telephone concerts was very
      on account of the buzzing generated by carbon-granule microphones. As a res
      Cahill decided to set a new standard in perfection of sound <span
        its-loc-quality-issue-comment="should be 'quality'"
        its-loc-quality-issue-profile-ref=grammar
        its-loc-quality-issue-severity=50
        its-loc-quality-issue-type=misspelling>qulaity</span> with his instrument
      a standard that would not only satisfy listeners but that
      would overcome all the flaws of traditional instruments.</p>
  </body>
</html>
```

[Source file: examples/html5/EX-locQualityIssue-html5-local-1.html]

Example 75: Annotating an issue in XML with local standoff markup

The following example shows a document using local standoff markup to encode several issues. The `mrk` element delimits the content to markup and holds a `locQualityIssuesRef` attribute that points to the `locQualityIssues` element where the issues are listed.

```
<?xml version="1.0"?>
<xliff version="1.2" xmlns="urn:oasis:names:tc:xliff:document:1.2"
  xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <file original="example.doc" source-language="en" datatype="plaintext">
    <body>
      <trans-unit id="1">
        <source xml:lang="en">This is the content</source>
        <target xml:lang="fr"><mrk mtype="x-itslq"
            its:locQualityIssuesRef="#lq1">c'es</mrk> le contenu</target>
        <its:locQualityIssues xml:id="lq1">
          <its:locQualityIssue locQualityIssueType="misspelling"
            locQualityIssueComment="'c'es' is unknown. Could be 'c'est'"
            locQualityIssueSeverity="50"/>
          <its:locQualityIssue locQualityIssueType="typographical"
            locQualityIssueComment="Sentence without capitalization"
            locQualityIssueSeverity="30"/>
        </its:locQualityIssues>
      </trans-unit>
    </body>
  </file>
</xliff>
```

[Source file: examples/xml/EX-locQualityIssue-local-2.xml]

Example 76: Annotating an issue in HTML with local standoff markup

The following example shows a document using local standoff markup to encode several issues. The `span` element delimits the content to markup and holds a `loc-quality-issues-ref` attribute that points to a special `span` element where the issues are listed within a set of other special `span` elements.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=utf-8>
    <title>Test</title>
    <script src=qaissues.js type=text/javascript></script>
    <script type=application/its+xml id=lq1>
      <its:locQualityIssues xml:id="lq1" xmlns:its="http://www.w3.org/2005/11/its
        <its:locQualityIssue
          locQualityIssueType="misspelling"
          locQualityIssueComment="'c'es' is unknown. Could be 'c'est'"
          locQualityIssueSeverity="50"/>
        <its:locQualityIssue
          locQualityIssueType="typographical"
          locQualityIssueComment="Sentence without capitalization"
          locQualityIssueSeverity="30"/>
      </its:locQualityIssues>
    </script>
    <style type=text/css>.qaissue { background-color: yellow; } </style>
  </head>
  <body onload=addqaissueattrs()>
    <p>
      <span its-loc-quality-issues-ref=#lq1>c'es</span> le contenu</p>
  </body>
</html>
```

[Source file: examples/html5/EX-locQualityIssue-html5-local-2.html]

8.17 Localization Quality Rating

8.17.1 Definition

The Localization Quality Rating data category is used to express an overall measurement of the localization quality of a document or an item in a document.

This data category allows to specify a quality score or a voting result for a given item or document, as well as to indicate what constitutes a passing score or vote. It also allows pointing to a profile describing the quality assessment model used for the scoring or the voting.

8.17.2 Implementation

The Localization Quality Rating data category is only expressed locally on individual elements. The data category information inherits to the textual content of the element, *including* child elements, but *excluding* attributes.

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

- Exactly one of the following:
  - A `locQualityRatingScore` attribute. Its value is a rational number in the interval 0 to 100 (inclusive). The value follows the XML Schema double data type with the constraining facets minInclusive set to 0 and maxInclusive set to 100. The higher values represent better quality.
  - A `locQualityRatingVote` attribute. Its value is a signed integer with higher values indicating a better vote.
- If `locQualityRatingScore` is used:
  - an optional `locQualityRatingScoreThreshold` attribute indicating the lowest score that constitutes a passing score in the profile used. Its value is a rational number in the interval 0 to 100 (inclusive). The value follows the XML Schema double data type with the constraining facets minInclusive set to 0 and maxInclusive set to 100.
- If `locQualityRatingVote` is used:
  - an optional `locQualityRatingVoteThreshold` attribute indicating the lowest value that constitutes a passing vote in the profile used. Its value is a signed integer.

W3C Recommendation

- An optional locQualityRatingProfileRef attribute. Its value is an IRI pointing to the reference document describing the quality assessment model used for the scoring.

Example 77: The Localization Quality Rating data category expressed locally in XML

The locQualityRatingScore, locQualityRatingThreshold and locQualityRatingProfileRef are used to score the quality of the document.

```
<?xml version="1.0"?>
<doc xml:lang='nl'
  xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0"
  its:locQualityRatingScore="100"
  its:locQualityRatingScoreThreshold="95"
  its:locQualityRatingProfileRef="http://example.org/qaModel/v13">
  <title>De lotgevallen van Tom Sawyer</title>
  <para>Hij kwam vrij laat te huis, en toen hij voorzichtig het raam insprong,
   viel hij in eene hinderlaag, in de persoon van zijne tante, bij wie, toen zij
   den staat zag, waarin zijne kleederen verkeerden, het besluit om zijn vrijen
   Zaterdag in een gevangenschap met dwangarbeid te veranderen, onherroepelijk
   vaststond.</para>
</doc>
```

[Source file: examples/xml/EX-locQualityRating-local-1.xml]

Example 78: The Localization Quality Rating data category expressed locally in HTML

The its-loc-quality-rating-score, its-loc-quality-rating-score-threshold and its-loc-quality-rating-profile-ref are used to score the quality of the document.

```
<!DOCTYPE html>
<html lang=fr
  its-loc-quality-rating-profile-ref=http://example.org/qaModel/v13
  its-loc-quality-rating-score=90
  its-loc-quality-rating-score-threshold=80
  >
 <head>
  <meta charset=utf-8>
  <title>Rikki-tikki-tavi</title>
 </head>
 <body>
  <p>C'est l'histoire de la grande guerre que Rikki-Tikki-Tavi a combattu tout se
   à travers les salles de bain du grand bungalow au cantonnement Segowlee. Darze
   le tailbird, l'a aidé, et Chuchundra, le rat musqué, qui ne sort jamais jusqu'
   milieu du plancher, mais se glisse toujours contre la paroi, lui donnait des
   conseils, mais Rikki-Tikki-Tavi fait le véritable combat.</p>
 </body>
</html>
```

[Source file: examples/html5/EX-locQualityRating-html5-local.html]

8.18 MT Confidence

8.18.1 Definition

The MT Confidence data category is used to communicate the confidence score from a machine translation engine for the accuracy of a translation it has provided. It is not intended to provide a score that is comparable between machine translation engines and platforms. This data category does NOT aim to establish any sort of correlation between the confidence score and either human evaluation of MT usefulness, or post-editing cognitive effort. For harmonization's sake, MT Confidence is provided as a rational number in the interval 0 to 1 (inclusive).

> **Note:**
>
> Implementers are expected to interpret the floating-point number and present it to human and other consumers in a convenient form, such as percentage (0-100%) with up to 2 decimal digits, font or background color coding, etc.

> **Note:**
>
> The value provided by the MT Confidence data category can be 1) the quality score of the translation as produced by an MT engine, or 2) a quality estimation score that uses both MT-system-internal features and additional external features. For this reason it is important that MT Confidence provides additional information about the MT engine (via the `annotatorsRef` attribute, or in HTML the `its-annotators-ref` attribute). Otherwise the score on its own is hard to interpret and to reuse. In the case of 2), MT Confidence potentially conveys information about any additional tools that were used in deriving the score.

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

- Automated prioritising of raw machine translated text for further processing based on empirically set thresholds.
- Providing readers, translators, post-editors, reviewers, and proof-readers of machine translated text with self-reported relative accuracy prediction.

MT confidence scores can be displayed e.g., on websites machine translated on the fly, by simple web-based translation editors or by Computer Aided Translation (CAT) tools.

8.18.2 Implementation

The MT Confidence category can be expressed with global rules or locally on individual elements. For elements, the data category information is inherited by the textual content of the element, *including* child elements, but *excluding* attributes.

Any node selected by the MT Confidence data category MUST be contained in an element with the `annotatorsRef` (or in HTML, `its-annotators-ref`) attribute specified for the MT Confidence data category. For more information, see Section 5.7: ITS Tools Annotation.

GLOBAL: The `mtConfidenceRule` element contains the following:

- A required `selector` attribute. It contains an absolute selector that selects the nodes to which this rule applies.
- A required `mtConfidence` attribute with a value that represents the translation confidence score as a rational number in the interval 0 to 1 (inclusive). The value follows the XML Schema double data type with the constraining facets minInclusive set to 0 and maxInclusive set to 1.

W3C Recommendation

Example 79: Global usage of mtConfidenceRule in a HTML document to specify the confidence scores for the translation into English of the title attributes of two img elements.

```
<!DOCTYPE html>
<html lang=en>
 <head>
  <meta charset=utf-8>
  <link href=EX-mtconfidence-global-html5-1-external-rules.xml rel=its-rules>
   <title>Machine translated title attributes of img elements give MT
         confidence scores using global rules</title>
  </head>
  <body its-annotators-ref="mt-confidence|file:///tools.xml#T1">
   <p>
    <img src="http://upload.wikimedia.org/wikipedia/commons/9/93/Trinity_College.
         title="Front gate of Trinity College Dublin"
         alt="alternative description"/>
   <img src="http://upload.wikimedia.org/wikipedia/commons/c/cc/Molly_alone.jpg"
        title="A tart with a cart"
        alt="alternative description"/>
  </p>
 </body>
</html>
```

[Source file: examples/html5/EX-mtConfidence-global-html5-1.html]

Where the external ITS rules file is as shown:

Example 80: XML file with external rules references from an HTML file.

```
<?xml version="1.0" encoding="UTF-8"?>
<its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0"
    xmlns:h="http://www.w3.org/1999/xhtml">
    <its:mtConfidenceRule mtConfidence="0.785"
      selector="//h:img[@title='Front gate of Trinity College Dublin']/@title"/>
    <its:mtConfidenceRule mtConfidence="0.805"
      selector="//h:img[@title='A tart with a cart']/@title"/>
</its:rules>
```

[Source file: examples/html5/EX-mtconfidence-global-html5-1-external-rules.xml]

LOCAL: the following local markup is available for the MT Confidence data category:

- A mtConfidence attribute with a value that represents the translation confidence score as a rational number in the interval 0 to 1 (inclusive). The value follows the XML Schema double data type with the constraining facets minInclusive set to 0 and maxInclusive set to 1.

Example 81: The MT Confidence data category expressed locally for the content of a span in an XML document.

```
<text xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0"
  its:annotatorsRef="mt-confidence|file:///tools.xml#T1">
  <body>
    <p>
      <span its:mtConfidence="0.8982">Dublin is the capital city of
```

```
        Ireland.</span>
     </p>
   </body>
</text>
```

[Source file: examples/xml/EX-mtConfidence-local-1.xml]

Example 82: The MT Confidence data category expressed locally for the content of two separate spans in a HTML document.

```
<!DOCTYPE html>
<html lang=en >
   <head>
      <meta charset=utf-8>
      <title>Sentences about Dublin and Prague
         machine translated from Czech with mtConfidence locally.</title>
   </head>
    <body its-annotators-ref="mt-confidence|file:///tools.xml#T1">
       <p>
         <span its-mt-confidence=0.8982>Dublin is the capital of Ireland.</span>
         <span its-mt-confidence=0.8536 >The capital of the Czech Republic is Pra
      </p>
   </body>
</html>
```

[Source file: examples/html5/EX-mtConfidence-html5-local-1.html]

8.19 Allowed Characters

8.19.1 Definition

The Allowed Characters data category is used to specify the characters that are permitted in a given piece of content.

This data category can be used for various purposes, including the following examples:

- Limiting the characters that may be used in the UI of a game due to font restrictions.
- Preventing illegal characters from being entered as text content that represents file or directory names.
- Controlling what characters can be used when translating examples of a login name in content.

Note:

The Allowed Characters data category is not intended to disallow HTML markup. The purpose is to restrict the content to various characters only, e.g., when the content is to be used for URL or filename generation. In most Content Management Systems, content is divided into several fields, some of which may be restricted to plain text, while in other fields HTML fragments may be allowed. Enforcing such restrictions is outside the scope of this data category.

The set of characters that are allowed is specified using a regular expression. That is, each character in the selected content MUST be included in the set specified by the regular expression.

The regular expression is the character class construct `charClass` defined as follows:

```
[1] charClass ::= singleCharEsc | charClassExpr | wildcardEsc
[2] singleCharEsc ::= '\' [nrt\|.?*+(){}#x2D#x5B#x5D#x5E]
[3] charClassExpr ::= '[' charGroup ']'
[4] charGroup ::= posCharGroup | negCharGroup
[5] posCharGroup ::= ( charRange | singleCharEsc )+
[6] charRange ::= seRange | xmlCharIncDash
[7] seRange ::= charOrEsc '-' charOrEsc
[8] charOrEsc ::= xmlChar | singleCharEsc
[9] xmlChar ::= [^\#x2D#x5B#x5D]
[10] xmlCharIncDash ::= [^\#x5B#x5D]
[11] negCharGroup ::= '^' posCharGroup
[12] wildcardEsc ::= '.'
```

The `.` metacharacter also matches CARRIAGE RETURN (U+000D) and LINE FEED (U+000F). That is the *dot-all* option is set.

This construct is a sub-set of the Character Classes construct of XML Schema [XML Schema Part 2] and is compatible with most other regular expression engines.

> **Note:**
>
> Users may want to use a regular expression to make sure that they follow the definition given above. Sample regular expressions to verify the regular expression in allowed characters are provided: for XML and for Java.

Example of expressions (shown as XML source):

- "`[abc]`": allows the characters 'a', 'b' and 'c'.
- "`[a-c]`": allows the characters 'a', 'b' and 'c'.
- "`[a-zA-Z]`": allows the characters from 'a' to 'z' and from 'A' to 'Z'.
- "`[^abc]`": allows any characters except 'a', 'b', and 'c'.
- "`[^&#x0061;-c]`": allows any characters except 'a', 'b', and 'c'.
- "`[^&lt;>:&quot;\\/|\?*]`": allows only the characters valid for Windows file names.
- "`.`": allows any character.
- "": allows no character.

8.19.2 Implementation

The Allowed Characters data category can be expressed with global rules, or locally on individual elements. For elements, the data category information inherits to the textual content of the element, *including* child elements, but *excluding* attributes.

GLOBAL: The `allowedCharactersRule` element contains the following:

- A required `selector` attribute. It contains an absolute selector that selects the nodes to which this rule applies.
- Exactly one of the following:
  - An `allowedCharacters` attribute that contains the regular expression indicating the allowed characters.
  - An `allowedCharactersPointer` attribute that contains a relative selector pointing to a node with the exact same semantics as `allowedCharacters`.

Example 83: The Allowed Characters data category expressed globally in XML

The `allowedCharactersRule` element states that the translated content of elements `content` cannot contain the characters `*` and `+`.

```
<?xml version="1.0"?>
<myRes xmlns:its="http://www.w3.org/2005/11/its">
  <head>
    <its:rules version="2.0">
      <its:allowedCharactersRule allowedCharacters="[^*+]" selector="//content"/>
    </its:rules>
  </head>
  <body>
    <content>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam
      nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed
      diam voluptua.</content>
  </body>
</myRes>
```

[Source file: examples/xml/EX-allowedCharacters-global-1.xml]

Example 84: Mapping the Allowed Characters data category in XML

The attribute `allowedCharactersPointer` is used to map the data category to the non-ITS attribute `set` in this document. The attribute has the same semantics as `allowedCharacters`.

```
<?xml version="1.0"?>
<res xmlns:its="http://www.w3.org/2005/11/its">
  <head>
    <its:rules version="2.0">
      <its:allowedCharactersRule selector="//record" allowedCharactersPointer="@s
    </its:rules>
  </head>
  <record id="a1" set="[ &#xFF01;-&#xFF5E;]">ＦＵＬＬ　ＷＩＤＴＨ　ＯＮＬＹ</record>
</res>
```

[Source file: examples/xml/EX-allowedCharacters-global-2.xml]

LOCAL: the following local markup is available for the Allowed Characters data category:

- A `allowedCharacters` attribute that contains the regular expression indicating the allowed characters.

Example 85: The Allowed Characters data category expressed locally in XML

The local `allowedCharacters` attribute specifies that the translated content of element `panelmsg` is only allowed to contain Unicode characters between U+0020 and U+00FE.

```
<?xml version="1.0"?>
<messages xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <msg num="123">Click the <panelmsg its:allowedCharacters="[&#x0020;-&#x00FE;]"
      >CONTINUE</panelmsg> Button on the printer panel</msg>
</messages>
```

[Source file: examples/xml/EX-allowedCharacters-local-1.xml]

Example 86: The Allowed Characters data category expressed locally in HTML

The local `its-allowed-characters` attribute specifies that the translated content of element `code` cannot contain the characters other than 'a' to 'z' in any case and the characters underscore and minus.

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <title>Example</title>
  </head>
  <body>
    <p>Login names can only use letters from A to Z (upper or lowercase)
    and the character underscore (_) and minus (-).
    For example: <code its-allowed-characters=[a-zA-Z_\-]>Huck_Finn</code>.</p>
  </body>
</html>
```

[Source file: examples/html5/EX-allowedCharacters-html5-local-1.html]

8.20 Storage Size

8.20.1 Definition

The Storage Size data category is used to specify the maximum storage size of a given content.

This data category can be used for various purposes, including the following examples:

- Verify during translation if a string fits into a fixed-size database field.
- Control the size of a string that is stored in a fixed-size memory buffer at run-time.

The storage size is always expressed in bytes and excludes any leading Byte-Order-Markers. It is provided along with the character encoding and the line break type that will be used when the content is stored. If the encoding form does not use the byte as its unit (e.g. UTF-16 uses 16-bit code units) the storage size MUST still be given in byte (e.g., for UTF-16: 2 bytes per 16-bit code unit).

An application verifying the storage size for a given content is expected to perform the following steps:

- All the LINE FEED (U+000A) characters of the content to verify are replaced by the character or characters specified by the line break type.
- The resulting string is converted to an array of bytes using a character encoder for the specified encoding. If a character cannot be represented with the specified encoding, an error is generated.
- If the leading bytes represent a Byte-Order-Mark, they are stripped from that array.
- The length of the resulting array is compared to the storage size provided. The content is too long if the length is greater than the storage size.

Note:

Storage size is not directly related to the display length of a text, and therefore is not intended as a display length constraint mechanism.

8.20.2 Implementation

The Storage Size data category can be expressed with global rules, or locally on individual elements. There is no inheritance. The default value of the character encoding is "UTF-8", and the default value for the line break is "lf" (LINE FEED (U+000A)).

GLOBAL: The `storageSizeRule` element contains the following:

- A required `selector` attribute. It contains an absolute selector that selects the nodes to which this rule applies.
- Exactly one of the following:
  - A `storageSize` attribute. It contains the maximum number of bytes the text of the selected node is allowed in storage.
  - A `storageSizePointer` attribute that contains a relative selector pointing to a node with the exact same semantics as `storageSize`.
- None or exactly one of the following:
  - A `storageEncoding` attribute. It contains the name of the character encoding used to calculate the number of bytes of the selected text. The name MUST be one of the names or aliases listed in the IANA Character Sets registry [IANA Character Sets]. The default value is the string "UTF-8".
  - A `storageEncodingPointer` attribute that contains a relative selector pointing to a node with the exact same semantics as `storageEncoding`.
- An optional `lineBreakType` attribute. It indicates what type of line breaks the storage uses. The possible values are: "cr" for CARRIAGE RETURN (U+000D), "lf" for LINE FEED (U+000A), or "crlf" for CARRIAGE RETURN (U+000D) followed by LINE FEED (U+000A). The default value is "lf".

Example 87: The Storage Size data category expressed globally in XML

The `storageSizeRule` element is used to specify that, when encoded in ISO-8859-1, the content of the `country` element cannot be more than 25 bytes. The name "Papouasie-Nouvelle-Guinée" is 25 character long and fits because all characters in ISO-8859-1 are encoded as a single byte.

```
<?xml version="1.0"?>
<db>
  <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
    <its:storageSizeRule selector="//country" storageSize="25"
      storageEncoding="ISO-8859-1"/>
  </its:rules>
  <data>
    <country id="123">Papouasie-Nouvelle-Guinée</country>
    <country id="139">République Dominicaine</country>
  </data>
</db>
```

[Source file: examples/xml/EX-storageSize-global-1.xml]

Example 88: Mapping the Storage Size data category in XML

The `storageSizePointer` attribute is used to map the non-ITS attribute `max` to the same functionality as `storageSize`. There is no character encoding specified, so the default UTF-8 is assumed. Note that, while the name "Papouasie-Nouvelle-Guinée" is 25 characters long, the character 'é' is encoded into two bytes in UTF-8. Therefore this name is one byte too long to fit in its storage destination.

```
<?xml version="1.0"?>
<fields>
  <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
    <its:storageSizeRule selector="//field" storageSizePointer="@max"/>
  </its:rules>
  <field type="country" id="123" max="25">Papouasie-Nouvelle-Guinée</field>
  <field type="country" id="139" max="25">République Dominicaine</field>
</fields>
```

[Source file: examples/xml/EX-storageSize-global-2.xml]

LOCAL: the following local markup is available for the Storage Size data category:

- A `storageSize` attribute. It contains the maximum number of bytes the text of the selected node is allowed in storage.
- An optional `storageEncoding` attribute. It contains the name of the character encoding used to calculate the number of bytes of the selected text. The name MUST be one of the names or aliases listed in the IANA Character Sets registry [IANA Character Sets]. The default value is the string "UTF-8".
- An optional `lineBreakType` attribute. It indicates what type of line breaks the storage uses. The possible values are: "cr" for CARRIAGE RETURN (U+000D), "lf" for LINE FEED (U+000A), or "crlf" for CARRIAGE RETURN (U+000D) followed by LINE FEED (U+000A). The default value is "lf".

Example 89: The Storage Size data category expressed locally in XML

The `storageSize` attribute allows specification of different maximum storage sizes throughout the document. Note that the string `CONTINUE` does not fit the specified restriction of 8 bytes. The minimal number of bytes to store such a string in UTF-16 is 16.

```
<?xml version="1.0"?>
<messages xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <var num="panelA1_Continue" its:storageSize="8" its:storageEncoding="UTF-16">CO
  <var num="panelA1_Stop" its:storageSize="8" its:storageEncoding="UTF-16">STOP</
  <var num="panelB5_Cancel" its:storageSize="12" its:storageEncoding="UTF-16">CAN
</messages>
```

[Source file: examples/xml/EX-storageSize-local-1.xml]

Example 90: The Storage Size data category expressed locally in HTML

The `its-storage-size` is used here to specify the maximum number of bytes the two editable strings can have in UTF-8.

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <title>Example</title>
  </head>
  <body>
    <p>String to translate:</p>
    <p contenteditable=true id=123 its-storage-size=25>Papua New-Guinea</p>
    <p contenteditable=true id=139 its-storage-size=25>Dominican Republic</p>
  </body>
</html>
```

[Source file: examples/html5/EX-storageSize-html5-local-1.html]

A References

This section is normative.

BCP47
Addison Phillips, Mark Davis. *Tags for Identifying Languages*, September 2009. Available at http://www.rfc-editor.org/rfc/bcp/bcp47.txt.

HTML 4.01
Dave Raggett et al. *HTML 4.01*. W3C Recommendation 24 December 1999. Available at http://www.w3.org/TR/1999/REC-html401-19991224/. The latest version of HTML 4.01 is available at http://www.w3.org/TR/html401.

HTML5
Robin Berjon et al. *HTML5*. W3C Candidate Recommendation 06 August 2013. Available at http://www.w3.org/TR/2013/CR-html5-20130806/. The latest version of HTML5 is available at http://www.w3.org/TR/html5/.

IANA Character Sets
Character Sets Available at http://www.iana.org/assignments/character-sets.

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.

Selectors Level 3
Tantek Çelik, Elika J. Etemad, Daniel Glazman, Ian Hickson, Peter Linss, John Williams *Selectors Level 3*. W3C Recommendation 29 September 2011. Available at http://www.w3.org/TR/2011/REC-css3-selectors-20110929/. The latest version of Selectors Level 3 is available at http://www.w3.org/TR/css3-selectors/.

Unicode
The Unicode Consortium. *The Unicode Standard, Version 6.2.0*, , ISBN 978-1-936213-07-8, as updated from time to time by the publication of new versions. (See http://www.unicode.org/unicode/standard/versions for the latest version and additional information on versions of the standard and of the Unicode Character Database).

XLink 1.1
Steve DeRose, Eve Maler, David Orchard, Norman Walsh. *XML Linking Language 1.1*. W3C Recommendation 6 May 2010. Available at http://www.w3.org/TR/2010/REC-xlink11-20100506/. The latest version of XLink 1.1 is available at http://www.w3.org/TR/xlink11/.

XML 1.0
Tim Bray, Jean Paoli, C.M. Sperberg-McQueen, et al., editors. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*, W3C Recommendation 26 November 2008. Available at

http://www.w3.org/TR/2008/REC-xml-20081126//. The latest version of XML 1.0 is available at http://www.w3.org/TR/xml/.

XML ID

Jonathan Marsh, Daniel Veillard, Norman Walsh. *xml:id Version 1.0*. W3C Recommendation 9 September 2005. Available at http://www.w3.org/TR/2005/REC-xml-id-20050909/. The latest version of xml:id Version 1.0 is available at http://www.w3.org/TR/xml-id/.

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/.

XML Schema Part 2

Paul V. Biron, Ashok Malhotra. *XML Schema Part 2: Datatypes Second Edition*. W3C Recommendation 28 October 2004. Available at http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/. The latest version of XML Schema is available at http://www.w3.org/TR/xmlschema-2/.

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 Internationalization Tag Set (ITS) MIME Type

This section is normative.

This section defines a MIME type for Internationalization Tag Set (ITS) documents. It covers both ITS 1.0 and ITS 2.0.

Type name: application

Subtype name: its+xml

Required parameters: none

Optional parameters: charset

This parameter has identical semantics to the charset parameter of the "application/xml" media type as specified in IETF RFC 3023.

Encoding considerations: Identical to those of "application/xml" as described in IETF RFC 3023, section 3.2, as applied to an ITS document.

Security considerations: An ITS 1.0 or ITS 2.0 document may cause arbitrary URIs or IRIs to be dereferenced, via the @xlink:href attribute at the its:rules element. Therefore, the security issues of [RFC 3987] Section 8 should be considered. In addition, the contents of resources identified by file: URIs can in some cases be accessed, processed and returned as results. An implementation of ITS global rules requires the support of XPath 1.0 or its successor. Hence, processing of global rules might encompass dereferencing of URIs or IRIs during computation of XPath expressions. Arbitrary recursion is possible, as is arbitrarily large memory usage, and implementations may place limits on

CPU and memory usage, as well as restricting access to system-defined functions. ITS 1.0 and ITS 2.0 permit extensions. Hence it is possible that application/its+xml may describe content that has security implications beyond those described here.

Interoperability considerations: There are no known interoperability issues.

Published specification: http://www.w3.org/TR/2007/REC-its-20070403/ and http://www.w3.org/TR/its20/.

Any XML document containing ITS 1.0 "its:rules" elements http://www.w3.org/TR/its/#selection-global can be labeled with `application/its+xml`. http://www.w3.org/TR/its/EX-link-external-rules-2.xml Provides an example of a document linking to a file with ITS 1.0 and ITS 2.0 "rules". The link target is at http://www.w3.org/TR/its/EX-link-external-rules-1.xml. There is no need that the link target has "its:rules" as a root element. The processing semantics is that rules are gathered in document order.

Applications that use this media type: This new media type is being registered to allow for deployment of ITS 1.0 and ITS 2.0 on the World Wide Web., e.g., by localization tools.

Additional information:

- Magic number(s): none
- File extension(s): .its
- Macintosh file type code(s): TEXT

Person & email address to contact for further information: World Wide Web Consortium <web-human at w3.org>

Intended usage: COMMON

Restrictions on usage: none

Author / Change controller: The Internationalization Tag Set (ITS) 1.0 and 2.0 specifications are a work product of the World Wide Web Consortium's Internationalization Tag Set Working Group. The W3C has change control over this specification.

C Values for the Localization Quality Issue Type

This section is normative.

The `locQualityIssueType` attribute provides a basic level of interoperability between different localization quality assurance tools. It offers a list of high-level quality issue types common in fully automatic and manual localization quality assessment. Tools can map their internal types to these types in order to exchange information about the kinds of issues they identify and take appropriate action even if another tool does not know the specific issues identified by the generating tool.

> **Note:**
>
> Note: The values of locQualityIssueType were derived from an early version of the QTLaunchPad project's Multidimensional Quality Metrics (MQM) framework. MQM is based on a careful analysis of existing translation quality assessment tools and models, such as the LISA QA Model, SAE J2450, and various commercial tools. The values represent common issue types found in those models and are designed to provide interoperability between models. Differences in granularity and in issue types may prevent full interoperability, but using the shared values will maximize interoperability where possible.

The scope column in the following table identifies whether the issue type applies to the source content (“S”), target content (“T”) or both (“S or T”).

W3C Recommendation

The values listed in the following table are allowed for locQualityIssueType. Ideally the values a tool implementing the data category produces for the attribute matches one of the values provided in this table and are as semantically accurate as possible. For example, marking the phrase “These man is” as a terminology issue, rather than as a grammar issue would be semantically inaccurate. Tools are encouraged to map their internal values to these types. The value other is reserved strictly for values that cannot be mapped.

> **Note:**
>
> For tools *generating* ITS 2.0 Localization Quality Issue markup, if one internal issue type can be categorized as multiple ITS 2.0 issue types, *the first applicable one from the following table should be used*. The list is ordered with more specific types first. For example, if a terminology database specifies that the term “USB memory stick” should be used instead of “USB pen drive” but the translated content has “Insert a USB pen drive into any available USB port”, terminology would be used instead of mistranslation because terminology occurs earlier in the list and is more specific than a (general) mistranslation. In the case where multiple separate issues must be marked on a single span (e.g., it contains both a mistranslation and a grammar issue), implementers may wish to use standoff annotation, as shown in Example 75 and Example 76.

> **Note:**
>
> The ITS Interest Group maintains informative mappings of tool-specific quality issue types and ITS 2.0 localization quality types. The ITS IG Wiki provides information on how to update that list. The purpose of these mappings is to document how tool internal information relates to the ITS 2.0 quality types. To foster interoperability, implementers are strongly encouraged to implement the ITS 2.0 quality types natively.

Value	Description	Example	Scope	Notes
terminology	An incorrect term or a term from the wrong domain was used or terms are used inconsistently.	- The localization had “Pen Drive” when corporate terminology specified that “USB Stick” was to be used. - The localized text inconsistently used "Start" and "Begin". - A text renders the Hungarian term *recsegőhid* as “buzzer bridge” in English (a literal translation), but the term to be used in English is	S or T	This value is not intended for simple typographical errors or word choice not related to definec terminologies. For example, a mistyping of “pin” as “pen” or the use of “imply” instead of “infer” (mistaking two commonly confused words) would not count as terminology issues and is best categorized as either spelling errors or mistranslations, depending on the nature of the issue. Terminology refers *only* to cases where incorrect choices about terms (either formal or commonly defined in a domain) are involved.

Value	Description	Example	Scope	Notes
		“wedge block,” as specified in a terminology list supplied to the translator.		
`mistranslation`	The content of the target mistranslates the content of the source.	• The English source reads "An ape succeeded in grasping a banana lying outside its cage with the help of a stick" but the Italian translation reads "l'ape riuscì a prendere la banana posta tuori dall sua gabbia aiutandosi con un bastone" ("A bee succeeded...")	T	Issues related to translation of specific terms related to the domain or task-specific language are to be categorized as `terminology` issues.
`omission`	Necessary text has been omitted from the localization or source.	• One or more segments found in the source that have been intended for translation are missing in the target. • After an alignment, a verification tool flags the pairs of aligned segments where the target has no corresponding source because of incorrect segmentation or some alignment issue. In such case the 'omission' type may apply to	S or T	This value is not to be used for missing whitespace or formatting codes, but instead has to be reserved for linguistic content.

Value	Description	Example	Scope	Notes
		the source entry.		
`untranslated`	Content that has been intended for translation is left untranslated.	• The source segment reads "The Professor said to Smith that he would hear from his lawyer" but the Hungarian localization reads "A professzor azt mondta Smithnek, hogy he would hear from his lawyer."	T	`omission` takes precedence ove `untranslated`. Omissions are distinct in that they address cases where text is not present, while `untranslated` addresses cases where text has been carried from the source untranslated.
`addition`	The translated text contains inappropriate additions.	• The translated text contains a note from the translator to himself to look up a term; the note ought to have been deleted but was not.	T	
`duplication`	Content has been duplicated improperly.	• A section of the target text was inadvertently copied twice in a copy and paste operation.	T	
`inconsistency`	The text is inconsistent with itself or is translated inconsistently (NB: not for use with terminology inconsistency).	• The text states that an event happened in 1912 in one location but in another states that it happened in 1812. • The translated text uses different wording for multiple instances of a single regulatory	S or T	

Value	Description	Example	Scope	Notes
		notice that occurs in multiple locations in a series of manuals.		
`grammar`	The text contains a grammatical error (including errors of syntax and morphology).	• The text reads "The guidelines says that users should use a static grounding strap."	S or T	
`legal`	The text is legally problematic (e.g., it is specific to the wrong legal system).	• The localized text is intended for use in Thailand but includes U.S. regulatory notices. • A text translated into German contains comparative advertising claims that are not allowed by German law.	S or T	
`register`	The text is written in the wrong linguistic register of uses slang or other language variants inappropriate to the text.	• A financial text in U.S. English refers to dollars as "bucks".	S or T	
`locale-specific-content`	The localization contains content that does not apply to the locale for which it was prepared.	• A text translated for the Japanese market contains call center numbers in Texas and refers to special offers available only in the U.S.	S or T	Legally inappropriate material is t be classified as `legal`.
`locale-violation`	Text violates norms for the intended locale.	• A text localized into	S or T	This value can be used for spelling errors only if they relate specifically to locale expectations

Value	Description	Example	Scope	Notes
		German has dates in `mm/dd/yyyy` format instead of `DD.MM.YYYY`. - A text for the Irish market uses American-style foot and inch measurements instead of centimeters. - A text intended for a U.S.-based audience uses U.K. spellings such as “centre” and “colour.”		(e.g., a text consistently uses British instead of U.S. spellings fo a text intended for the U.S.). If these errors are not systematic (e.g., a text uses U.S. spellings but has a single instance of “centre”), they are instead to be counted as spelling errors.
`style`	The text contains stylistic errors.	- Company style guidelines dictate that all individuals be referred to as Mr. or Ms. with a family name, but the text refers to “Jack Smith”.	S or T	
`characters`	The text contains characters that are garbled or incorrect or that are not used in the language in which the content appears.	- A text ought to have a '•' but instead has a '¥' sign. - A German text erroneously uses û, ô, and â instead of the appropriate 'ü', 'ö', and 'ä'. - A Japanese text has been garbled and appears with Devanagari characters.	S or T	Characters ought to be used in cases of garbling or systematic use of inappropriate characters, not for spelling issues where individual characters are replaced with incorrect one.
`misspelling`	The text contains a misspelling.	- A German text misspells the word "Zustellung"	S or T	

Value	Description	Example	Scope	Notes
		as "Zustlelung".		
`typographical`	The text has typographical errors such as omitted/incorrect punctuation, incorrect capitalization, etc.	- An English text has the following sentence: "The man whom, we saw, was in the Military and carried it's insignias".	S or T	
`formatting`	The text is formatted incorrectly.	- Warnings in the text are supposed to be set in italic face, but instead appear in bold face. - Margins of the text are narrower than specified.	S or T	
`inconsistent-entities`	The source and target text contain different named entities (dates, times, place names, individual names, etc.)	- The name "Thaddeus Cahill" appears in an English source but is rendered as "Tamaš Cahill" in the Czech version. - The date "February 9, 2007" appears in the source but the translated text has "2. September 2007".	S or T	
`numbers`	Numbers are inconsistent between source and target.	- A source text states that an object is 120 cm long, but the target text says that it is 129 cm. long.	S or T	Some tools may correct for differences in units of measurement to reduce false positives (e.g., a tool might adjus for differences in values between inches and centimeters to avoid flagging numbers that seem to be different but are in fact equivalent).

Value	Description	Example	Scope	Notes
`markup`	There is an issue related to markup or a mismatch in markup between source and target.	• The source segment has five markup tags but the target has only two. • An opening tag in the text is missing a closing tag.	S or T	
`pattern-problem`	The text fails to match a pattern that defines allowable content (or matches one that defines non-allowable content).	• The tool disallows the regular expression pattern ["”’][\.,] but the translated text contains "A leading “expert”, a political hack, claimed otherwise." • A tool uses a regular expression to ensure that the content of an element is an IRI and flags what appears to be a malformed IRI.	S or T	Defining what is or is not an allowable pattern is up to the processing application and is beyond the scope of this specification. Best practice would be to use the Comment attribute to specify the pattern that led to the issue.
`whitespace`	There is a mismatch in whitespace between source and target content or the text violates specific rules related to the use of whitespace.	• A source segment starts with six space characters but the corresponding target segment has two non-breaking spaces at the start. • The text uses a run of 12 space characters instead of a tab character to align numbers in a table.	S or T	

Value	Description	Example	Scope	Notes
		• Two space characters appear after a period even though only a single space is to be used.		
`internationalization`	There is an issue related to the internationalization of content.	• A line of programming code has embedded language-specific strings. • A user interface element leaves no room for text expansion. • A form allows only for U.S.-style postal addresses and expects five digit U.S. ZIP codes.	S or T	There are many kinds of internationalization issues. This value is therefore very heterogeneous in what it can refe to.
`length`	There is a significant difference in source and target length.	• The translation of a segment is five times as long as the source.	S or T	What constitutes a "significant" difference in length is determined by the model referred to in the `locQualityIssueProfileRef`
`non-conformance`	The content is deemed to show poor statistical conformance to a reference corpus. Higher severity values reflect poorer conformance.	The sentence "The harbour connected which to printer is busy or configared not properly." would have poor conformance.	S or T	Non-conformance is determined through the use of multiple statistical measures of similarity t a corpus of known-good content. For example, in a system that uses classification techniques the poor conformance might be a function of combined incorrect terminology, wrong spelling and bad grammar, or other features a determined by the system.
`uncategorized`	The issue either has not been categorized or cannot be categorized.	• A new version of a tool returns information on an issue that has not been previously checked and that is not yet classified.	S or T	This value has the following uses 1. A tool can use it to pass through quality data fron another tool in cases where the issues from the other tool are not classified (for example, a localization quality assurance tool interfaces

Value	Description	Example	Scope	Notes
		- A text is defective in ways that defy categorization, such as the appearance of nonsense garbled text of unknown origin (e.g., a translation shows an unintelligible result and/or appears unrelated to the source material).		with a third-party grammar checker). 2. A tool’s issues are not yet assigned to values, and, until an updated assignment is made, they may be listed as `uncategorized`. In thi case it is recommended that issues be assigned to appropriate values as soon as possible since `uncategorized` does not foster interoperabilit 3. `uncategorized` can b used where a portion of text is defective in a way that defies assignment t a value in either the originating system or in any other ITS localizatio quality markup to indicate that it is uncategorizable.
`other`	Any issue that cannot be assigned to any values listed above.		S or T	- This value allows for the inclusion of any issues not included in the previously listed values. It is encouraged not to use this value for any tool- or model-specific issues that can be mapped to the values listed above. - In addition, this value is not synonymous with `uncategorized` in tha `uncategorized` issue may be assigned to another precise value, while other issues cannot. - If a system has an "miscellaneous" or "other" value, it is better to map this to this value even if the specific instance of the issue might be mapped to another value.

Note:

The value `uncategorized` is used for issues that have not (yet) been categorized into a more specific value. For example, an automatic process might flag issues for attention but not provide any further detail or categorization: such issues would be listed as `uncategorized` in ITS 2.0. It may also be used when the exact nature of an issue is unclear and it cannot be categorized as a result (e.g., text is seriously garbled and the cause it unclear). By contrast other is used when the nature of an issue is clear but it cannot be categorized in one of the ITS 2.0 categories (or when a model or tool has its own “other” category). For example, in translation of subtitles there is a “respeaking” error category that does not correspond to any ITS 2.0 category and is highly specific to that environment; respeaking errors would therefore be categorized as `other` in ITS 2.0.

D Schemas for ITS

This section is informative.

Note:

The schemas are only informative and may be updated any time. An updated version of the schemas can be found in the ITS 2.0 test suite.

The following schemas define ITS elements and attributes and can 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.

Foreign elements can be used only inside `rules`. Foreign attributes can be used on any element defined in ITS.

The following four schemas are provided:

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

Example 91: NVDL schema for ITS

```
<?xml version="1.0" encoding="UTF-8"?>
<rules xmlns="http://purl.oclc.org/dsdl/nvdl/ns/structure/1.0">
  <namespace ns="http://www.w3.org/2005/11/its">
    <validate schema="its20-elements.rng"/>
  </namespace>
  <namespace ns="http://www.w3.org/2005/11/its" match="attributes">
    <validate schema="its20-attributes.rng"/>
  </namespace>
  <anyNamespace>
    <allow/>
  </anyNamespace>
</rules>
```

[Source file: schemas/its20.nvdl]

2. RELAX NG schema for elements and attributes: The NVDL schema depends on the following two schemas: RELAX NG schema for ITS elements, and RELAX NG schema for all ITS local attributes.

Example 92: RELAX NG schema for ITS elements

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar xmlns="http://relaxng.org/ns/structure/1.0">
  <include href="its20.rng"/>
  <start>
    <choice>
      <ref name="its-rules"/>
      <ref name="its-span"/>
      <ref name="its-standoff"/>
    </choice>
  </start>
</grammar>
```

[Source file: schemas/its20-elements.rng]

(RELAX NG compact syntax version of schema)

Example 93: RELAX NG schema for all ITS local attributes

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar xmlns="http://relaxng.org/ns/structure/1.0">
  <include href="its20.rng"/>
  <start>
    <group>
      <optional>
        <ref name="its-local.attributes"/>
      </optional>
      <optional>
        <ref name="its-attribute.version"/>
      </optional>
    </group>
  </start>
</grammar>
```

[Source file: schemas/its20-attributes.rng]

(RELAX NG compact syntax version of schema)

3. Base RELAX NG schema for ITS: All ITS elements and attributes referenced by previous two schemas are defined in the base RELAX NG schema for ITS.

Example 94: Base RELAX NG schema for ITS

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar ns="http://www.w3.org/2005/11/its" xmlns:a="http://relaxng.org/ns/compat
  <include href="its20-types.rng"/>
  <define name="its-attribute.translate">
    <attribute name="its:translate">
      <ref name="its-translate.type"/>
    </attribute>
  </define>
  <define name="its-attribute.translate.nons">
    <attribute name="translate">
      <ref name="its-translate.type"/>
    </attribute>
  </define>
  <define name="its-attribute.dir">
```

```
      <attribute name="its:dir">
        <ref name="its-dir.type"/>
      </attribute>
    </define>
    <define name="its-attribute.dir.nons">
      <attribute name="dir">
        <ref name="its-dir.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locNote">
      <attribute name="its:locNote">
        <ref name="its-locNote.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locNote.nons">
      <attribute name="locNote">
        <ref name="its-locNote.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locNoteType">
      <attribute name="its:locNoteType">
        <ref name="its-locNoteType.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locNoteType.nons">
      <attribute name="locNoteType">
        <ref name="its-locNoteType.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locNoteRef">
      <attribute name="its:locNoteRef">
        <ref name="its-locNoteRef.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locNoteRef.nons">
      <attribute name="locNoteRef">
        <ref name="its-locNoteRef.type"/>
      </attribute>
    </define>
    <define name="its-attribute.termInfoRef">
      <attribute name="its:termInfoRef">
        <ref name="its-termInfoRef.type"/>
      </attribute>
    </define>
    <define name="its-attribute.termInfoRef.nons">
      <attribute name="termInfoRef">
        <ref name="its-termInfoRef.type"/>
      </attribute>
    </define>
    <define name="its-attribute.term">
      <attribute name="its:term">
        <ref name="its-term.type"/>
      </attribute>
    </define>
    <define name="its-attribute.term.nons">
      <attribute name="term">
        <ref name="its-term.type"/>
    </attribute>
  </define>
  <define name="its-attribute.termConfidence">
    <attribute name="its:termConfidence">
      <ref name="its-termConfidence.type"/>
    </attribute>
  </define>
  <define name="its-attribute.termConfidence.nons">
    <attribute name="termConfidence">
      <ref name="its-termConfidence.type"/>
    </attribute>
  </define>
  <define name="its-attribute.withinText">
    <attribute name="its:withinText">
      <ref name="its-withinText.type"/>
    </attribute>
  </define>
  <define name="its-attribute.withinText.nons">
    <attribute name="withinText">
      <ref name="its-withinText.type"/>
    </attribute>
  </define>
  <define name="its-attribute.domainMapping">
    <attribute name="its:domainMapping">
      <ref name="its-domainMapping.type"/>
    </attribute>
  </define>
  <define name="its-attribute.domainMapping.nons">
    <attribute name="domainMapping">
      <ref name="its-domainMapping.type"/>
    </attribute>
  </define>
  <define name="its-attribute.taConfidence">
    <attribute name="its:taConfidence">
      <ref name="its-taConfidence.type"/>
    </attribute>
  </define>
  <define name="its-attribute.taConfidence.nons">
    <attribute name="taConfidence">
      <ref name="its-taConfidence.type"/>
    </attribute>
  </define>
  <define name="its-attribute.taClassRef">
    <attribute name="its:taClassRef">
      <ref name="its-taClassRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.taClassRef.nons">
    <attribute name="taClassRef">
      <ref name="its-taClassRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.taIdent">
    <attribute name="its:taIdent">
      <ref name="its-taIdent.type"/>
    </attribute>
  </define>
```

```
<define name="its-attribute.taIdent.nons">
  <attribute name="taIdent">
    <ref name="its-taIdent.type"/>
  </attribute>
</define>
<define name="its-attribute.taIdentRef">
  <attribute name="its:taIdentRef">
    <ref name="its-taIdentRef.type"/>
  </attribute>
</define>
<define name="its-attribute.taIdentRef.nons">
  <attribute name="taIdentRef">
    <ref name="its-taIdentRef.type"/>
  </attribute>
</define>
<define name="its-attribute.taSource">
  <attribute name="its:taSource">
    <ref name="its-taSource.type"/>
  </attribute>
</define>
<define name="its-attribute.taSource.nons">
  <attribute name="taSource">
    <ref name="its-taSource.type"/>
  </attribute>
</define>
<define name="its-attribute.localeFilterList">
  <attribute name="its:localeFilterList">
    <ref name="its-localeFilterList.type"/>
  </attribute>
</define>
<define name="its-attribute.localeFilterList.nons">
  <attribute name="localeFilterList">
    <ref name="its-localeFilterList.type"/>
  </attribute>
</define>
<define name="its-attribute.localeFilterType">
  <attribute name="its:localeFilterType">
    <ref name="its-localeFilterType.type"/>
  </attribute>
</define>
<define name="its-attribute.localeFilterType.nons">
  <attribute name="localeFilterType">
    <ref name="its-localeFilterType.type"/>
  </attribute>
</define>
<define name="its-attribute.person">
  <attribute name="its:person">
    <ref name="its-person.type"/>
  </attribute>
</define>
<define name="its-attribute.person.nons">
  <attribute name="person">
    <ref name="its-person.type"/>
  </attribute>
</define>
<define name="its-attribute.personRef">
  <attribute name="its:personRef">
```

```
      <ref name="its-personRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.personRef.nons">
    <attribute name="personRef">
      <ref name="its-personRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.org">
    <attribute name="its:org">
      <ref name="its-org.type"/>
    </attribute>
  </define>
  <define name="its-attribute.org.nons">
    <attribute name="org">
      <ref name="its-org.type"/>
    </attribute>
  </define>
  <define name="its-attribute.orgRef">
    <attribute name="its:orgRef">
      <ref name="its-orgRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.orgRef.nons">
    <attribute name="orgRef">
      <ref name="its-orgRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.tool">
    <attribute name="its:tool">
      <ref name="its-tool.type"/>
    </attribute>
  </define>
  <define name="its-attribute.tool.nons">
    <attribute name="tool">
      <ref name="its-tool.type"/>
    </attribute>
  </define>
  <define name="its-attribute.toolRef">
    <attribute name="its:toolRef">
      <ref name="its-toolRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.toolRef.nons">
    <attribute name="toolRef">
      <ref name="its-toolRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.revPerson">
    <attribute name="its:revPerson">
      <ref name="its-revPerson.type"/>
    </attribute>
  </define>
  <define name="its-attribute.revPerson.nons">
    <attribute name="revPerson">
      <ref name="its-revPerson.type"/>
    </attribute>
  </define>
  <define name="its-attribute.revPersonRef">
    <attribute name="its:revPersonRef">
      <ref name="its-revPersonRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.revPersonRef.nons">
    <attribute name="revPersonRef">
      <ref name="its-revPersonRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.revOrg">
    <attribute name="its:revOrg">
      <ref name="its-revOrg.type"/>
    </attribute>
  </define>
  <define name="its-attribute.revOrg.nons">
    <attribute name="revOrg">
      <ref name="its-revOrg.type"/>
    </attribute>
  </define>
  <define name="its-attribute.revOrgRef">
    <attribute name="its:revOrgRef">
      <ref name="its-revOrgRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.revOrgRef.nons">
    <attribute name="revOrgRef">
      <ref name="its-revOrgRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.revTool">
    <attribute name="its:revTool">
      <ref name="its-revTool.type"/>
    </attribute>
  </define>
  <define name="its-attribute.revTool.nons">
    <attribute name="revTool">
      <ref name="its-revTool.type"/>
    </attribute>
  </define>
  <define name="its-attribute.revToolRef">
    <attribute name="its:revToolRef">
      <ref name="its-revToolRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.revToolRef.nons">
    <attribute name="revToolRef">
      <ref name="its-revToolRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.provRef">
    <attribute name="its:provRef">
      <ref name="its-provRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.provRef.nons">
```

```
      <attribute name="provRef">
        <ref name="its-provRef.type"/>
      </attribute>
    </define>
    <define name="its-attribute.provenanceRecordsRef">
      <attribute name="its:provenanceRecordsRef">
        <ref name="its-provenanceRecordsRef.type"/>
      </attribute>
    </define>
    <define name="its-attribute.provenanceRecordsRef.nons">
      <attribute name="provenanceRecordsRef">
        <ref name="its-provenanceRecordsRef.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityIssuesRef">
      <attribute name="its:locQualityIssuesRef">
        <ref name="its-locQualityIssuesRef.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityIssuesRef.nons">
      <attribute name="locQualityIssuesRef">
        <ref name="its-locQualityIssuesRef.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityIssueType">
      <attribute name="its:locQualityIssueType">
        <ref name="its-locQualityIssueType.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityIssueType.nons">
      <attribute name="locQualityIssueType">
        <ref name="its-locQualityIssueType.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityIssueComment">
      <attribute name="its:locQualityIssueComment">
        <ref name="its-locQualityIssueComment.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityIssueComment.nons">
      <attribute name="locQualityIssueComment">
        <ref name="its-locQualityIssueComment.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityIssueSeverity">
      <attribute name="its:locQualityIssueSeverity">
        <ref name="its-locQualityIssueSeverity.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityIssueSeverity.nons">
      <attribute name="locQualityIssueSeverity">
        <ref name="its-locQualityIssueSeverity.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityIssueProfileRef">
      <attribute name="its:locQualityIssueProfileRef">
        <ref name="its-locQualityIssueProfileRef.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityIssueProfileRef.nons">
      <attribute name="locQualityIssueProfileRef">
        <ref name="its-locQualityIssueProfileRef.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityIssueEnabled">
      <attribute name="its:locQualityIssueEnabled">
        <ref name="its-locQualityIssueEnabled.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityIssueEnabled.nons">
      <attribute name="locQualityIssueEnabled">
        <ref name="its-locQualityIssueEnabled.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityRatingScore">
      <attribute name="its:locQualityRatingScore">
        <ref name="its-locQualityRatingScore.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityRatingScore.nons">
      <attribute name="locQualityRatingScore">
        <ref name="its-locQualityRatingScore.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityRatingVote">
      <attribute name="its:locQualityRatingVote">
        <ref name="its-locQualityRatingVote.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityRatingVote.nons">
      <attribute name="locQualityRatingVote">
        <ref name="its-locQualityRatingVote.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityRatingScoreThreshold">
      <attribute name="its:locQualityRatingScoreThreshold">
        <ref name="its-locQualityRatingScoreThreshold.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityRatingScoreThreshold.nons">
      <attribute name="locQualityRatingScoreThreshold">
        <ref name="its-locQualityRatingScoreThreshold.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityRatingVoteThreshold">
      <attribute name="its:locQualityRatingVoteThreshold">
        <ref name="its-locQualityRatingVoteThreshold.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityRatingVoteThreshold.nons">
      <attribute name="locQualityRatingVoteThreshold">
        <ref name="its-locQualityRatingVoteThreshold.type"/>
      </attribute>
    </define>
```

```
    <define name="its-attribute.locQualityRatingProfileRef">
      <attribute name="its:locQualityRatingProfileRef">
        <ref name="its-locQualityRatingProfileRef.type"/>
      </attribute>
    </define>
    <define name="its-attribute.locQualityRatingProfileRef.nons">
      <attribute name="locQualityRatingProfileRef">
        <ref name="its-locQualityRatingProfileRef.type"/>
      </attribute>
    </define>
    <define name="its-attribute.mtConfidence">
      <attribute name="its:mtConfidence">
        <ref name="its-mtConfidence.type"/>
      </attribute>
    </define>
    <define name="its-attribute.mtConfidence.nons">
      <attribute name="mtConfidence">
        <ref name="its-mtConfidence.type"/>
      </attribute>
    </define>
    <define name="its-attribute.allowedCharacters">
      <attribute name="its:allowedCharacters">
        <ref name="its-allowedCharacters.type"/>
      </attribute>
    </define>
    <define name="its-attribute.allowedCharacters.nons">
      <attribute name="allowedCharacters">
        <ref name="its-allowedCharacters.type"/>
      </attribute>
    </define>
    <define name="its-attribute.storageSize">
      <attribute name="its:storageSize">
        <ref name="its-storageSize.type"/>
      </attribute>
    </define>
    <define name="its-attribute.storageSize.nons">
      <attribute name="storageSize">
        <ref name="its-storageSize.type"/>
      </attribute>
    </define>
    <define name="its-attribute.storageEncoding">
      <attribute name="its:storageEncoding">
        <ref name="its-storageEncoding.type"/>
      </attribute>
    </define>
    <define name="its-attribute.storageEncoding.nons">
      <attribute name="storageEncoding">
        <ref name="its-storageEncoding.type"/>
      </attribute>
    </define>
    <define name="its-attribute.lineBreakType">
      <attribute name="its:lineBreakType">
        <ref name="its-lineBreakType.type"/>
      </attribute>
    </define>
    <define name="its-attribute.lineBreakType.nons">
      <attribute name="lineBreakType">
        <ref name="its-lineBreakType.type"/>
      </attribute>
    </define>
    <define name="its-attribute.annotatorsRef">
      <attribute name="its:annotatorsRef">
        <ref name="its-annotatorsRef.type"/>
      </attribute>
    </define>
    <define name="its-attribute.annotatorsRef.nons">
      <attribute name="annotatorsRef">
        <ref name="its-annotatorsRef.type"/>
      </attribute>
    </define>
    <define name="its-attribute.version">
      <attribute name="its:version">
        <a:documentation>Version of ITS</a:documentation>
        <ref name="its-version.type"/>
      </attribute>
    </define>
    <define name="its-attribute.version.nons">
      <attribute name="version">
        <a:documentation>Version of ITS</a:documentation>
        <ref name="its-version.type"/>
      </attribute>
    </define>
    <define name="its-attribute.queryLanguage">
      <attribute name="its:queryLanguage">
        <ref name="its-queryLanguage.type"/>
      </attribute>
    </define>
    <define name="its-attribute.queryLanguage.nons">
      <attribute name="queryLanguage">
        <ref name="its-queryLanguage.type"/>
      </attribute>
    </define>
    <define name="its-attribute.xlink.href">
      <attribute name="xlink:href">
        <data type="anyURI"/>
      </attribute>
    </define>
    <define name="its-attribute.xlink.type">
      <attribute name="xlink:type">
        <value>simple</value>
      </attribute>
    </define>
    <define name="its-attribute.selector">
      <attribute name="selector">
        <ref name="its-absolute-selector.type"/>
      </attribute>
    </define>
    <define name="its-foreign-attribute">
      <attribute>
        <anyName>
          <except>
            <nsName ns=""/>
          </except>
        </anyName>
```

```
      </attribute>
    </define>
    <define name="its-foreign-no-xml-id-attribute">
      <attribute>
        <anyName>
          <except>
            <nsName ns=""/>
            <name>xml:id</name>
          </except>
        </anyName>
      </attribute>
    </define>
    <define name="its-foreign-no-xlink-attribute">
      <attribute>
        <anyName>
          <except>
            <nsName ns=""/>
            <nsName ns="http://www.w3.org/1999/xlink"/>
          </except>
        </anyName>
      </attribute>
    </define>
    <define name="its-any-attribute">
      <attribute>
        <anyName/>
      </attribute>
    </define>
    <define name="its-any-element">
      <element>
        <anyName/>
        <zeroOrMore>
          <choice>
            <zeroOrMore>
              <ref name="its-any-attribute"/>
            </zeroOrMore>
            <text/>
            <ref name="its-any-element"/>
          </choice>
        </zeroOrMore>
      </element>
    </define>
    <define name="its-foreign-element">
      <element>
        <anyName>
          <except>
            <nsName/>
          </except>
        </anyName>
        <zeroOrMore>
          <choice>
            <zeroOrMore>
              <ref name="its-any-attribute"/>
            </zeroOrMore>
            <text/>
            <ref name="its-foreign-element"/>
          </choice>
        </zeroOrMore>
      </element>
    </define>
    <define name="its-rules">
      <element name="rules">
        <a:documentation>Container for global rules</a:documentation>
        <ref name="its-rules.content"/>
        <ref name="its-rules.attributes"/>
      </element>
    </define>
    <define name="its-rules.content">
      <zeroOrMore>
        <ref name="its-param"/>
      </zeroOrMore>
      <zeroOrMore>
        <choice>
          <ref name="its-translateRule"/>
          <ref name="its-locNoteRule"/>
          <ref name="its-termRule"/>
          <ref name="its-dirRule"/>
          <ref name="its-langRule"/>
          <ref name="its-withinTextRule"/>
          <ref name="its-domainRule"/>
          <ref name="its-textAnalysisRule"/>
          <ref name="its-localeFilterRule"/>
          <ref name="its-provRule"/>
          <ref name="its-locQualityIssueRule"/>
          <ref name="its-mtConfidenceRule"/>
          <ref name="its-externalResourceRefRule"/>
          <ref name="its-targetPointerRule"/>
          <ref name="its-idValueRule"/>
          <ref name="its-preserveSpaceRule"/>
          <ref name="its-allowedCharactersRule"/>
          <ref name="its-storageSizeRule"/>
          <ref name="its-foreign-element"/>
        </choice>
      </zeroOrMore>
    </define>
    <define name="its-rules.attributes">
      <ref name="its-attribute.version.nons"/>
      <optional>
        <ref name="its-attribute.xlink.href"/>
        <optional>
          <ref name="its-attribute.xlink.type"/>
        </optional>
      </optional>
      <optional>
        <ref name="its-attribute.queryLanguage.nons"/>
      </optional>
      <zeroOrMore>
        <ref name="its-foreign-no-xlink-attribute"/>
      </zeroOrMore>
    </define>
    <define name="its-param">
      <element name="param">
        <a:documentation>Declaration of variable used in selectors</a:documentation
        <ref name="its-param.content"/>
        <ref name="its-param.attributes"/>
      </element>
    </define>
    <define name="its-param.content">
      <text/>
    </define>
    <define name="its-param.attributes">
      <attribute name="name">
        <data type="string"/>
      </attribute>
      <zeroOrMore>
        <ref name="its-foreign-attribute"/>
      </zeroOrMore>
    </define>
    <define name="its-local.attributes">
      <interleave>
        <optional>
          <ref name="its-attribute.translate"/>
        </optional>
        <optional>
          <ref name="its-attribute.dir"/>
        </optional>
        <optional>
          <choice>
            <ref name="its-attribute.locNote"/>
            <ref name="its-attribute.locNoteRef"/>
          </choice>
          <optional>
            <ref name="its-attribute.locNoteType"/>
          </optional>
        </optional>
        <optional>
          <ref name="its-attribute.term"/>
          <optional>
            <ref name="its-attribute.termInfoRef"/>
          </optional>
          <optional>
            <ref name="its-attribute.termConfidence"/>
          </optional>
        </optional>
        <optional>
          <ref name="its-attribute.withinText"/>
        </optional>
        <optional>
          <interleave>
            <optional>
              <ref name="its-attribute.taConfidence"/>
            </optional>
            <interleave>
              <optional>
                <ref name="its-attribute.taClassRef"/>
              </optional>
              <optional>
                <choice>
                  <group>
                    <ref name="its-attribute.taSource"/>
                    <ref name="its-attribute.taIdent"/>
                  </group>
```

```
                  <ref name="its-attribute.taIdentRef"/>
                </choice>
              </optional>
            </interleave>
          </interleave>
        </optional>
        <optional>
          <ref name="its-attribute.localeFilterList"/>
          <optional>
            <ref name="its-attribute.localeFilterType"/>
          </optional>
        </optional>
        <optional>
          <choice>
            <interleave>
              <optional>
                <choice>
                  <ref name="its-attribute.person"/>
                  <ref name="its-attribute.personRef"/>
                </choice>
              </optional>
              <optional>
                <choice>
                  <ref name="its-attribute.org"/>
                  <ref name="its-attribute.orgRef"/>
                </choice>
              </optional>
              <optional>
                <choice>
                  <ref name="its-attribute.tool"/>
                  <ref name="its-attribute.toolRef"/>
                </choice>
              </optional>
              <optional>
                <choice>
                  <ref name="its-attribute.revPerson"/>
                  <ref name="its-attribute.revPersonRef"/>
                </choice>
              </optional>
              <optional>
                <choice>
                  <ref name="its-attribute.revOrg"/>
                  <ref name="its-attribute.revOrgRef"/>
                </choice>
              </optional>
              <optional>
                <choice>
                  <ref name="its-attribute.revTool"/>
                  <ref name="its-attribute.revToolRef"/>
                </choice>
              </optional>
              <optional>
                <ref name="its-attribute.provRef"/>
              </optional>
            </interleave>
            <ref name="its-attribute.provenanceRecordsRef"/>
          </choice>
        </optional>
        <optional>
          <choice>
            <ref name="its-attribute.locQualityIssuesRef"/>
            <interleave>
              <interleave>
                <optional>
                  <ref name="its-attribute.locQualityIssueType"/>
                </optional>
                <optional>
                  <ref name="its-attribute.locQualityIssueComment"/>
                </optional>
              </interleave>
              <optional>
                <ref name="its-attribute.locQualityIssueSeverity"/>
              </optional>
              <optional>
                <ref name="its-attribute.locQualityIssueProfileRef"/>
              </optional>
              <optional>
                <ref name="its-attribute.locQualityIssueEnabled"/>
              </optional>
            </interleave>
          </choice>
        </optional>
        <optional>
          <choice>
            <group>
              <ref name="its-attribute.locQualityRatingScore"/>
              <optional>
                <ref name="its-attribute.locQualityRatingScoreThreshold"/>
              </optional>
            </group>
            <group>
              <ref name="its-attribute.locQualityRatingVote"/>
              <optional>
                <ref name="its-attribute.locQualityRatingVoteThreshold"/>
              </optional>
            </group>
          </choice>
          <optional>
            <ref name="its-attribute.locQualityRatingProfileRef"/>
          </optional>
        </optional>
        <optional>
          <ref name="its-attribute.mtConfidence"/>
        </optional>
        <optional>
          <ref name="its-attribute.allowedCharacters"/>
        </optional>
        <optional>
          <ref name="its-attribute.storageSize"/>
          <optional>
            <ref name="its-attribute.storageEncoding"/>
          </optional>
          <optional>
            <ref name="its-attribute.lineBreakType"/>
        </optional>
      </optional>
      <optional>
        <ref name="its-attribute.annotatorsRef"/>
      </optional>
    </interleave>
  </define>
  <define name="its-local.nons.attributes">
    <interleave>
      <optional>
        <ref name="its-attribute.translate.nons"/>
      </optional>
      <optional>
        <ref name="its-attribute.dir.nons"/>
      </optional>
      <optional>
        <choice>
          <ref name="its-attribute.locNote.nons"/>
          <ref name="its-attribute.locNoteRef.nons"/>
        </choice>
        <optional>
          <ref name="its-attribute.locNoteType.nons"/>
        </optional>
      </optional>
      <optional>
        <ref name="its-attribute.term.nons"/>
        <optional>
          <ref name="its-attribute.termInfoRef.nons"/>
        </optional>
        <optional>
          <ref name="its-attribute.termConfidence.nons"/>
        </optional>
      </optional>
      <optional>
        <ref name="its-attribute.withinText.nons"/>
      </optional>
      <optional>
        <interleave>
          <optional>
            <ref name="its-attribute.taConfidence.nons"/>
          </optional>
          <interleave>
            <optional>
              <ref name="its-attribute.taClassRef.nons"/>
            </optional>
            <optional>
              <choice>
                <group>
                  <ref name="its-attribute.taSource.nons"/>
                  <ref name="its-attribute.taIdent.nons"/>
                </group>
                <ref name="its-attribute.taIdentRef.nons"/>
              </choice>
            </optional>
          </interleave>
        </interleave>
      </optional>
```

```
        <optional>
          <ref name="its-attribute.localeFilterList.nons"/>
          <optional>
            <ref name="its-attribute.localeFilterType.nons"/>
          </optional>
        </optional>
        <optional>
          <choice>
            <interleave>
              <optional>
                <choice>
                  <ref name="its-attribute.person.nons"/>
                  <ref name="its-attribute.personRef.nons"/>
                </choice>
              </optional>
              <optional>
                <choice>
                  <ref name="its-attribute.org.nons"/>
                  <ref name="its-attribute.orgRef.nons"/>
                </choice>
              </optional>
              <optional>
                <choice>
                  <ref name="its-attribute.tool.nons"/>
                  <ref name="its-attribute.toolRef.nons"/>
                </choice>
              </optional>
              <optional>
                <choice>
                  <ref name="its-attribute.revPerson.nons"/>
                  <ref name="its-attribute.revPersonRef.nons"/>
                </choice>
              </optional>
              <optional>
                <choice>
                  <ref name="its-attribute.revOrg.nons"/>
                  <ref name="its-attribute.revOrgRef.nons"/>
                </choice>
              </optional>
              <optional>
                <choice>
                  <ref name="its-attribute.revTool.nons"/>
                  <ref name="its-attribute.revToolRef.nons"/>
                </choice>
              </optional>
              <optional>
                <ref name="its-attribute.provRef.nons"/>
              </optional>
            </interleave>
            <ref name="its-attribute.provenanceRecordsRef.nons"/>
          </choice>
        </optional>
        <optional>
          <choice>
            <ref name="its-attribute.locQualityIssuesRef.nons"/>
            <interleave>
              <interleave>
```

```
                <optional>
                  <ref name="its-attribute.locQualityIssueType.nons"/>
                </optional>
                <optional>
                  <ref name="its-attribute.locQualityIssueComment.nons"/>
                </optional>
              </interleave>
              <optional>
                <ref name="its-attribute.locQualityIssueSeverity.nons"/>
              </optional>
              <optional>
                <ref name="its-attribute.locQualityIssueProfileRef.nons"/>
              </optional>
              <optional>
                <ref name="its-attribute.locQualityIssueEnabled.nons"/>
              </optional>
            </interleave>
          </choice>
        </optional>
        <optional>
          <choice>
            <group>
              <ref name="its-attribute.locQualityRatingScore.nons"/>
              <optional>
                <ref name="its-attribute.locQualityRatingScoreThreshold.nons"/>
              </optional>
            </group>
            <group>
              <ref name="its-attribute.locQualityRatingVote.nons"/>
              <optional>
                <ref name="its-attribute.locQualityRatingVoteThreshold.nons"/>
              </optional>
            </group>
          </choice>
          <optional>
            <ref name="its-attribute.locQualityRatingProfileRef.nons"/>
          </optional>
        </optional>
        <optional>
          <ref name="its-attribute.mtConfidence.nons"/>
        </optional>
        <optional>
          <ref name="its-attribute.allowedCharacters.nons"/>
        </optional>
        <optional>
          <ref name="its-attribute.storageSize.nons"/>
          <optional>
            <ref name="its-attribute.storageEncoding.nons"/>
          </optional>
          <optional>
            <ref name="its-attribute.lineBreakType.nons"/>
          </optional>
        </optional>
        <optional>
          <ref name="its-attribute.annotatorsRef.nons"/>
        </optional>
      </interleave>
```

```
    </define>
    <define name="its-span">
      <element name="span">
        <a:documentation>Inline element to contain ITS information</a:documentatio
        <ref name="its-span.content"/>
        <ref name="its-span.attributes"/>
      </element>
    </define>
    <define name="its-span.content">
      <zeroOrMore>
        <choice>
          <text/>
          <ref name="its-span"/>
        </choice>
      </zeroOrMore>
    </define>
    <define name="its-span.attributes">
      <interleave>
        <ref name="its-local.nons.attributes"/>
        <zeroOrMore>
          <ref name="its-foreign-attribute"/>
        </zeroOrMore>
      </interleave>
    </define>
    <define name="its-translateRule">
      <element name="translateRule">
        <a:documentation>Rule about the Translate data category</a:documentation>
        <ref name="its-translateRule.content"/>
        <ref name="its-translateRule.attributes"/>
      </element>
    </define>
    <define name="its-translateRule.content">
      <empty/>
    </define>
    <define name="its-translateRule.attributes">
      <ref name="its-attribute.selector"/>
      <ref name="its-attribute.translate.nons"/>
      <zeroOrMore>
        <ref name="its-foreign-attribute"/>
      </zeroOrMore>
    </define>
    <define name="its-locNoteRule">
      <element name="locNoteRule">
        <a:documentation>Rule about the Localization Note data category</a:document
        <ref name="its-attribute.selector"/>
        <ref name="its-attribute.locNoteType.nons"/>
        <choice>
          <ref name="its-locNote"/>
          <ref name="its-attribute.locNotePointer.nons"/>
          <ref name="its-attribute.locNoteRef.nons"/>
          <ref name="its-attribute.locNoteRefPointer.nons"/>
        </choice>
        <zeroOrMore>
          <ref name="its-foreign-attribute"/>
        </zeroOrMore>
      </element>
    </define>
```

```
<define name="its-attribute.locNotePointer.nons">
  <attribute name="locNotePointer">
    <ref name="its-relative-selector.type"/>
  </attribute>
</define>
<define name="its-attribute.locNoteRefPointer.nons">
  <attribute name="locNoteRefPointer">
    <ref name="its-relative-selector.type"/>
  </attribute>
</define>
<define name="its-locNote">
  <element name="locNote">
    <a:documentation>Localization note</a:documentation>
    <ref name="its-locNote.content"/>
    <ref name="its-locNote.attributes"/>
  </element>
</define>
<define name="its-locNote.content">
  <zeroOrMore>
    <choice>
      <text/>
      <ref name="its-span"/>
    </choice>
  </zeroOrMore>
</define>
<define name="its-locNote.attributes">
  <ref name="its-local.nons.attributes"/>
  <zeroOrMore>
    <ref name="its-foreign-attribute"/>
  </zeroOrMore>
</define>
<define name="its-termRule">
  <element name="termRule">
    <a:documentation>Rule about the Terminology data category</a:documentation>
    <ref name="its-termRule.content"/>
    <ref name="its-termRule.attributes"/>
  </element>
</define>
<define name="its-termRule.content">
  <empty/>
</define>
<define name="its-termRule.attributes">
  <ref name="its-attribute.selector"/>
  <ref name="its-attribute.term.nons"/>
  <optional>
    <choice>
      <ref name="its-attribute.termInfoPointer.nons"/>
      <ref name="its-attribute.termInfoRef.nons"/>
      <ref name="its-attribute.termInfoRefPointer.nons"/>
    </choice>
  </optional>
  <zeroOrMore>
    <ref name="its-foreign-attribute"/>
  </zeroOrMore>
</define>
<define name="its-attribute.termInfoPointer.nons">
  <attribute name="termInfoPointer">
```

```
        <ref name="its-relative-selector.type"/>
      </attribute>
    </define>
    <define name="its-attribute.termInfoRefPointer.nons">
      <attribute name="termInfoRefPointer">
        <ref name="its-relative-selector.type"/>
      </attribute>
    </define>
    <define name="its-dirRule">
      <element name="dirRule">
        <a:documentation>Rule about the Directionality data category</a:documentati
        <ref name="its-dirRule.content"/>
        <ref name="its-dirRule.attributes"/>
      </element>
    </define>
    <define name="its-dirRule.content">
      <empty/>
    </define>
    <define name="its-dirRule.attributes">
      <ref name="its-attribute.selector"/>
      <ref name="its-attribute.dir.nons"/>
      <zeroOrMore>
        <ref name="its-foreign-attribute"/>
      </zeroOrMore>
    </define>
    <define name="its-langRule">
      <element name="langRule">
        <a:documentation>Rule about the Language Information data category</a:docum
        <ref name="its-langRule.content"/>
        <ref name="its-langRule.attributes"/>
      </element>
    </define>
    <define name="its-langRule.content">
      <empty/>
    </define>
    <define name="its-langRule.attributes">
      <ref name="its-attribute.selector"/>
      <ref name="its-attribute.langPointer.nons"/>
      <zeroOrMore>
        <ref name="its-foreign-attribute"/>
      </zeroOrMore>
    </define>
    <define name="its-attribute.langPointer.nons">
      <attribute name="langPointer">
        <ref name="its-relative-selector.type"/>
      </attribute>
    </define>
    <define name="its-withinTextRule">
      <element name="withinTextRule">
        <a:documentation>Rule about the Elements Within Text data category</a:docum
        <ref name="its-withinTextRule.content"/>
        <ref name="its-withinTextRule.attributes"/>
      </element>
    </define>
    <define name="its-withinTextRule.content">
      <empty/>
    </define>
```

```
<define name="its-withinTextRule.attributes">
  <ref name="its-attribute.selector"/>
  <ref name="its-attribute.withinText.nons"/>
  <zeroOrMore>
    <ref name="its-foreign-attribute"/>
  </zeroOrMore>
</define>
<define name="its-domainRule">
  <element name="domainRule">
    <a:documentation>Rule about the Domain data category</a:documentation>
    <ref name="its-domainRule.content"/>
    <ref name="its-domainRule.attributes"/>
  </element>
</define>
<define name="its-domainRule.content">
  <empty/>
</define>
<define name="its-domainRule.attributes">
  <ref name="its-attribute.selector"/>
  <ref name="its-attribute.domainPointer.nons"/>
  <optional>
    <ref name="its-attribute.domainMapping.nons"/>
  </optional>
  <zeroOrMore>
    <ref name="its-foreign-attribute"/>
  </zeroOrMore>
</define>
<define name="its-attribute.domainPointer.nons">
  <attribute name="domainPointer">
    <ref name="its-relative-selector.type"/>
  </attribute>
</define>
<define name="its-textAnalysisRule">
  <element name="textAnalysisRule">
    <a:documentation>Rule about the Disambiguation data category</a:documentati
    <ref name="its-textAnalysisRule.content"/>
    <ref name="its-textAnalysisRule.attributes"/>
  </element>
</define>
<define name="its-textAnalysisRule.content">
  <empty/>
</define>
<define name="its-textAnalysisRule.attributes">
  <ref name="its-attribute.selector"/>
  <optional>
    <ref name="its-attribute.taClassRefPointer.nons"/>
  </optional>
  <optional>
    <choice>
      <group>
        <ref name="its-attribute.taSourcePointer.nons"/>
        <ref name="its-attribute.taIdentPointer.nons"/>
      </group>
      <ref name="its-attribute.taIdentRefPointer.nons"/>
    </choice>
  </optional>
  <zeroOrMore>
```

```
      <ref name="its-foreign-attribute"/>
    </zeroOrMore>
  </define>
  <define name="its-attribute.taClassRefPointer.nons">
    <attribute name="taClassRefPointer">
      <ref name="its-taClassRefPointer.type"/>
    </attribute>
  </define>
  <define name="its-attribute.taIdentPointer.nons">
    <attribute name="taIdentPointer">
      <ref name="its-taIdentPointer.type"/>
    </attribute>
  </define>
  <define name="its-attribute.taSourcePointer.nons">
    <attribute name="taSourcePointer">
      <ref name="its-taSourcePointer.type"/>
    </attribute>
  </define>
  <define name="its-attribute.taIdentRefPointer.nons">
    <attribute name="taIdentRefPointer">
      <ref name="its-taIdentRefPointer.type"/>
    </attribute>
  </define>
  <define name="its-localeFilterRule">
    <element name="localeFilterRule">
      <a:documentation>Rule about the LocaleFilter data category</a:documentation
      <ref name="its-localeFilterRule.content"/>
      <ref name="its-localeFilterRule.attributes"/>
    </element>
  </define>
  <define name="its-localeFilterRule.content">
    <empty/>
  </define>
  <define name="its-localeFilterRule.attributes">
    <ref name="its-attribute.selector"/>
    <ref name="its-attribute.localeFilterList.nons"/>
    <optional>
      <ref name="its-attribute.localeFilterType.nons"/>
    </optional>
    <zeroOrMore>
      <ref name="its-foreign-attribute"/>
    </zeroOrMore>
  </define>
  <define name="its-provRule">
    <element name="provRule">
      <a:documentation>Rule about the Provenance data category</a:documentation>
      <ref name="its-provRule.content"/>
      <ref name="its-provRule.attributes"/>
    </element>
  </define>
  <define name="its-provRule.content">
    <empty/>
  </define>
  <define name="its-provRule.attributes">
    <ref name="its-attribute.selector"/>
    <ref name="its-attribute.provenanceRecordsRefPointer.nons"/>
    <zeroOrMore>
      <ref name="its-foreign-attribute"/>
    </zeroOrMore>
  </define>
  <define name="its-attribute.provenanceRecordsRefPointer.nons">
    <attribute name="provenanceRecordsRefPointer">
      <ref name="its-relative-selector.type"/>
    </attribute>
  </define>
  <define name="its-externalResourceRefRule">
    <element name="externalResourceRefRule">
      <a:documentation>Rule about the External Resource data category</a:document
      <ref name="its-externalResourceRefRule.content"/>
      <ref name="its-externalResourceRefRule.attributes"/>
    </element>
  </define>
  <define name="its-externalResourceRefRule.content">
    <empty/>
  </define>
  <define name="its-externalResourceRefRule.attributes">
    <ref name="its-attribute.selector"/>
    <ref name="its-attribute.externalResourceRefPointer.nons"/>
    <zeroOrMore>
      <ref name="its-foreign-attribute"/>
    </zeroOrMore>
  </define>
  <define name="its-attribute.externalResourceRefPointer.nons">
    <attribute name="externalResourceRefPointer">
      <ref name="its-relative-selector.type"/>
    </attribute>
  </define>
  <define name="its-targetPointerRule">
    <element name="targetPointerRule">
      <a:documentation>Rule about the Target Pointer data category</a:documentati
      <ref name="its-targetPointerRule.content"/>
      <ref name="its-targetPointerRule.attributes"/>
    </element>
  </define>
  <define name="its-targetPointerRule.content">
    <empty/>
  </define>
  <define name="its-targetPointerRule.attributes">
    <ref name="its-attribute.selector"/>
    <ref name="its-attribute.targetPointer.nons"/>
    <zeroOrMore>
      <ref name="its-foreign-attribute"/>
    </zeroOrMore>
  </define>
  <define name="its-attribute.targetPointer.nons">
    <attribute name="targetPointer">
      <ref name="its-relative-selector.type"/>
    </attribute>
  </define>
  <define name="its-idValueRule">
    <element name="idValueRule">
      <a:documentation>Rule about the Id Value data category</a:documentation>
      <ref name="its-idValueRule.content"/>
      <ref name="its-idValueRule.attributes"/>
    </element>
  </define>
  <define name="its-idValueRule.content">
    <empty/>
  </define>
  <define name="its-idValueRule.attributes">
    <ref name="its-attribute.selector"/>
    <ref name="its-attribute.idValue.nons"/>
    <zeroOrMore>
      <ref name="its-foreign-attribute"/>
    </zeroOrMore>
  </define>
  <define name="its-attribute.idValue.nons">
    <attribute name="idValue">
      <ref name="its-xpath-expression.type"/>
    </attribute>
  </define>
  <define name="its-preserveSpaceRule">
    <element name="preserveSpaceRule">
      <a:documentation>Rule about the Preserve Space data category</a:documentati
      <ref name="its-preserveSpaceRule.content"/>
      <ref name="its-preserveSpaceRule.attributes"/>
    </element>
  </define>
  <define name="its-preserveSpaceRule.content">
    <empty/>
  </define>
  <define name="its-preserveSpaceRule.attributes">
    <ref name="its-attribute.selector"/>
    <ref name="its-attribute.space.nons"/>
    <zeroOrMore>
      <ref name="its-foreign-attribute"/>
    </zeroOrMore>
  </define>
  <define name="its-attribute.space.nons">
    <attribute name="space">
      <choice>
        <value>default</value>
        <value>preserve</value>
      </choice>
    </attribute>
  </define>
  <define name="its-locQualityIssueRule">
    <element name="locQualityIssueRule">
      <a:documentation>Rule about the Localization Quality Issue data category</a
      <ref name="its-locQualityIssueRule.content"/>
      <ref name="its-locQualityIssueRule.attributes"/>
    </element>
  </define>
  <define name="its-locQualityIssueRule.content">
    <empty/>
  </define>
  <define name="its-locQualityIssueRule.attributes">
    <ref name="its-attribute.selector"/>
    <choice>
      <choice>
        <ref name="its-attribute.locQualityIssuesRef.nons"/>
        <ref name="its-attribute.locQualityIssuesRefPointer.nons"/>
      </choice>
      <group>
        <oneOrMore>
          <choice>
            <ref name="its-attribute.locQualityIssueType.nons"/>
            <ref name="its-attribute.locQualityIssueComment.nons"/>
          </choice>
        </oneOrMore>
        <optional>
          <ref name="its-attribute.locQualityIssueSeverity.nons"/>
        </optional>
        <optional>
          <ref name="its-attribute.locQualityIssueProfileRef.nons"/>
        </optional>
        <optional>
          <ref name="its-attribute.locQualityIssueEnabled.nons"/>
        </optional>
      </group>
    </choice>
    <zeroOrMore>
      <ref name="its-foreign-attribute"/>
    </zeroOrMore>
  </define>
  <define name="its-attribute.locQualityIssuesRefPointer.nons">
    <attribute name="locQualityIssuesRefPointer">
      <ref name="its-relative-selector.type"/>
    </attribute>
  </define>
  <define name="its-mtConfidenceRule">
    <element name="mtConfidenceRule">
      <a:documentation>Rule about the MT Confidence data category</a:documentatic
      <ref name="its-mtConfidenceRule.content"/>
      <ref name="its-mtConfidenceRule.attributes"/>
    </element>
  </define>
  <define name="its-mtConfidenceRule.content">
    <empty/>
  </define>
  <define name="its-mtConfidenceRule.attributes">
    <ref name="its-attribute.selector"/>
    <ref name="its-attribute.mtConfidence.nons"/>
    <zeroOrMore>
      <ref name="its-foreign-attribute"/>
    </zeroOrMore>
  </define>
  <define name="its-allowedCharactersRule">
    <element name="allowedCharactersRule">
      <a:documentation>Rule about the Allowed Characters data category</a:documen
      <ref name="its-allowedCharactersRule.content"/>
      <ref name="its-allowedCharactersRule.attributes"/>
    </element>
  </define>
  <define name="its-allowedCharactersRule.content">
    <empty/>
  </define>
  <define name="its-allowedCharactersRule.attributes">
```

```
      <ref name="its-attribute.selector"/>
      <choice>
        <ref name="its-attribute.allowedCharacters.nons"/>
        <ref name="its-attribute.allowedCharactersPointer.nons"/>
      </choice>
      <zeroOrMore>
        <ref name="its-foreign-attribute"/>
      </zeroOrMore>
    </define>
    <define name="its-attribute.allowedCharactersPointer.nons">
      <attribute name="allowedCharactersPointer">
        <ref name="its-relative-selector.type"/>
      </attribute>
    </define>
    <define name="its-storageSizeRule">
      <element name="storageSizeRule">
        <a:documentation>Rule about the Allowed Characters data category</a:documen
        <ref name="its-storageSizeRule.content"/>
        <ref name="its-storageSizeRule.attributes"/>
      </element>
    </define>
    <define name="its-storageSizeRule.content">
      <empty/>
    </define>
    <define name="its-storageSizeRule.attributes">
      <ref name="its-attribute.selector"/>
      <choice>
        <ref name="its-attribute.storageSize.nons"/>
        <ref name="its-attribute.storageSizePointer.nons"/>
      </choice>
      <optional>
        <choice>
          <ref name="its-attribute.storageEncoding.nons"/>
          <ref name="its-attribute.storageEncodingPointer.nons"/>
        </choice>
      </optional>
      <optional>
        <ref name="its-attribute.lineBreakType.nons"/>
      </optional>
      <zeroOrMore>
        <ref name="its-foreign-attribute"/>
      </zeroOrMore>
    </define>
    <define name="its-attribute.storageSizePointer.nons">
      <attribute name="storageSizePointer">
        <ref name="its-relative-selector.type"/>
      </attribute>
    </define>
    <define name="its-attribute.storageEncodingPointer.nons">
      <attribute name="storageEncodingPointer">
        <ref name="its-relative-selector.type"/>
      </attribute>
    </define>
    <define name="its-standoff">
      <choice>
        <ref name="its-provenanceRecords"/>
        <ref name="its-locQualityIssues"/>
      </choice>
    </define>
    <define name="its-provenanceRecords">
      <element name="its:provenanceRecords">
        <a:documentation>Standoff markup for Provenance data category</a:documentat
        <oneOrMore>
          <ref name="its-provenanceRecord"/>
        </oneOrMore>
        <attribute name="xml:id">
          <data type="ID"/>
        </attribute>
        <optional>
          <ref name="its-attribute.version.nons"/>
        </optional>
        <zeroOrMore>
          <ref name="its-foreign-no-xml-id-attribute"/>
        </zeroOrMore>
      </element>
    </define>
    <define name="its-provenanceRecord">
      <element name="its:provenanceRecord">
        <a:documentation>Provenance record used in Provenance standoff markup</a:do
        <ref name="its-provenanceRecord.attributes"/>
      </element>
    </define>
    <define name="its-provenanceRecord.attributes">
      <interleave>
        <optional>
          <choice>
            <ref name="its-attribute.person.nons"/>
            <ref name="its-attribute.personRef.nons"/>
          </choice>
        </optional>
        <optional>
          <choice>
            <ref name="its-attribute.org.nons"/>
            <ref name="its-attribute.orgRef.nons"/>
          </choice>
        </optional>
        <optional>
          <choice>
            <ref name="its-attribute.tool.nons"/>
            <ref name="its-attribute.toolRef.nons"/>
          </choice>
        </optional>
        <optional>
          <choice>
            <ref name="its-attribute.revPerson.nons"/>
            <ref name="its-attribute.revPersonRef.nons"/>
          </choice>
        </optional>
        <optional>
          <choice>
            <ref name="its-attribute.revOrg.nons"/>
            <ref name="its-attribute.revOrgRef.nons"/>
          </choice>
        </optional>
        <optional>
          <choice>
            <ref name="its-attribute.revTool.nons"/>
            <ref name="its-attribute.revToolRef.nons"/>
          </choice>
        </optional>
        <optional>
          <ref name="its-attribute.provRef.nons"/>
        </optional>
        <zeroOrMore>
          <ref name="its-foreign-attribute"/>
        </zeroOrMore>
      </interleave>
    </define>
    <define name="its-locQualityIssues">
      <element name="its:locQualityIssues">
        <a:documentation>Standoff markup for Localization Quality Issue data catego
        <oneOrMore>
          <ref name="its-locQualityIssue"/>
        </oneOrMore>
        <attribute name="xml:id">
          <data type="ID"/>
        </attribute>
        <optional>
          <ref name="its-attribute.version.nons"/>
        </optional>
        <zeroOrMore>
          <ref name="its-foreign-no-xml-id-attribute"/>
        </zeroOrMore>
      </element>
    </define>
    <define name="its-locQualityIssue">
      <element name="its:locQualityIssue">
        <a:documentation>Issue recorded in Localization Quality standoff markup</a:
        <ref name="its-locQualityIssue.attributes"/>
      </element>
    </define>
    <define name="its-locQualityIssue.attributes">
      <interleave>
        <optional>
          <ref name="its-attribute.locQualityIssueType.nons"/>
        </optional>
        <optional>
          <ref name="its-attribute.locQualityIssueComment.nons"/>
        </optional>
        <optional>
          <ref name="its-attribute.locQualityIssueSeverity.nons"/>
        </optional>
        <optional>
          <ref name="its-attribute.locQualityIssueProfileRef.nons"/>
        </optional>
        <optional>
          <ref name="its-attribute.locQualityIssueEnabled.nons"/>
        </optional>
        <zeroOrMore>
          <ref name="its-foreign-attribute"/>
        </zeroOrMore>
    </interleave>
  </define>
</grammar>
```

[Source file: schemas/its20.rng]

(RELAX NG compact syntax version of schema)

4. Data type definitions: All datatypes used in the base RELAX NG schema are defined the following schema.

Example 95: RELAX NG schema with datatypes for ITS

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar xmlns:a="http://relaxng.org/ns/compatibility/annotations/1.0" xmlns="htt
  <define name="its-version.type">
    <a:documentation>Version of ITS</a:documentation>
    <data type="string">
      <param name="pattern">[0-9]+\.[0-9]+</param>
    </data>
  </define>
  <define name="its-queryLanguage.type">
    <a:documentation>The query language to be used for processing the rules</a:do
    <choice>
      <value>xpath</value>
      <value>css</value>
      <text/>
    </choice>
  </define>
  <define name="its-absolute-selector.type">
    <data type="string" datatypeLibrary="">
      <a:documentation>Absolute selector</a:documentation>
    </data>
  </define>
  <define name="its-relative-selector.type">
    <data type="string" datatypeLibrary="">
      <a:documentation>Relative selector</a:documentation>
    </data>
  </define>
  <define name="its-xpath-expression.type">
    <data type="string" datatypeLibrary=""/>
  </define>
  <define name="its-confidence.type">
    <data type="double">
      <param name="minInclusive">0</param>
      <param name="maxInclusive">1</param>
    </data>
  </define>
  <define name="its-translate.type">
    <a:documentation>The Translate data category information to be attached to th
    <choice>
      <value>yes</value>
      <a:documentation>The nodes need to be translated</a:documentation>
      <value>no</value>
      <a:documentation>The nodes must not be translated</a:documentation>
    </choice>
```

```
</define>
<define name="its-locNote.type">
  <data type="string" datatypeLibrary=""/>
</define>
<define name="its-locNoteType.type">
  <a:documentation>The type of localization note</a:documentation>
  <choice>
    <value>alert</value>
    <a:documentation>Localization note is an alert</a:documentation>
    <value>description</value>
    <a:documentation>Localization note is a description</a:documentation>
  </choice>
</define>
<define name="its-locNoteRef.type">
  <data type="anyURI"/>
</define>
<define name="its-termInfoRef.type">
  <data type="anyURI"/>
</define>
<define name="its-term.type">
  <a:documentation>Indicates a term locally</a:documentation>
  <choice>
    <value>yes</value>
    <a:documentation>The value 'yes' means that this is a term</a:documentation
    <value>no</value>
    <a:documentation>The value 'no' means that this is not a term</a:documentat
  </choice>
</define>
<define name="its-termConfidence.type">
  <ref name="its-confidence.type"/>
</define>
<define name="its-dir.type">
  <a:documentation>The text direction for the context</a:documentation>
  <choice>
    <value>ltr</value>
    <a:documentation>Left-to-right text</a:documentation>
    <value>rtl</value>
    <a:documentation>Right-to-left text</a:documentation>
    <value>lro</value>
    <a:documentation>Left-to-right override</a:documentation>
    <value>rlo</value>
    <a:documentation>Right-to-left override</a:documentation>
  </choice>
</define>
<define name="its-withinText.type">
  <a:documentation>States whether current context is regarded as "within text"<
  <choice>
    <value>yes</value>
    <a:documentation>The element and its content are part of the flow of its pa
    <value>no</value>
    <a:documentation>The element splits the text flow of its parent element and
    <value>nested</value>
    <a:documentation>The element is part of the flow of its parent element, its
  </choice>
</define>
<define name="its-domainMapping.type">
  <a:documentation>A comma separated list of mappings between values in the cor
```

```
and workflow specific values. The values may contain spaces; in
that case they MUST be delimited by quotation marks.</a:documentation>
    <data type="string" datatypeLibrary=""/>
  </define>
  <define name="its-taConfidence.type">
    <ref name="its-confidence.type"/>
  </define>
  <define name="its-taClassPointer.type">
    <ref name="its-relative-selector.type"/>
  </define>
  <define name="its-taClassRefPointer.type">
    <ref name="its-relative-selector.type"/>
  </define>
  <define name="its-taClassRef.type">
    <data type="anyURI"/>
  </define>
  <define name="its-taIdentRef.type">
    <data type="anyURI"/>
  </define>
  <define name="its-taIdent.type">
    <data type="string" datatypeLibrary=""/>
  </define>
  <define name="its-taSource.type">
    <data type="string" datatypeLibrary=""/>
  </define>
  <define name="its-taIdentPointer.type">
    <ref name="its-relative-selector.type"/>
  </define>
  <define name="its-taIdentRefPointer.type">
    <ref name="its-relative-selector.type"/>
  </define>
  <define name="its-taSourcePointer.type">
    <ref name="its-relative-selector.type"/>
  </define>
  <define name="its-localeFilterList.type">
    <data type="string" datatypeLibrary=""/>
  </define>
  <define name="its-localeFilterType.type">
    <choice>
      <value>include</value>
      <value>exclude</value>
    </choice>
  </define>
  <define name="its-provenanceRecordsRef.type">
    <data type="anyURI"/>
  </define>
  <define name="its-person.type">
    <data type="string" datatypeLibrary=""/>
  </define>
  <define name="its-personRef.type">
    <data type="anyURI"/>
  </define>
  <define name="its-org.type">
    <data type="string" datatypeLibrary=""/>
  </define>
  <define name="its-orgRef.type">
    <data type="anyURI"/>
```

```
</define>
<define name="its-tool.type">
  <data type="string" datatypeLibrary=""/>
</define>
<define name="its-toolRef.type">
  <data type="anyURI"/>
</define>
<define name="its-revPerson.type">
  <data type="string" datatypeLibrary=""/>
</define>
<define name="its-revPersonRef.type">
  <data type="anyURI"/>
</define>
<define name="its-revOrg.type">
  <data type="string" datatypeLibrary=""/>
</define>
<define name="its-revOrgRef.type">
  <data type="anyURI"/>
</define>
<define name="its-revTool.type">
  <data type="string" datatypeLibrary=""/>
</define>
<define name="its-revToolRef.type">
  <data type="anyURI"/>
</define>
<define name="its-provRef.type">
  <list>
    <oneOrMore>
      <data type="anyURI"/>
    </oneOrMore>
  </list>
</define>
<define name="its-externalResourceRefPointer.type">
  <ref name="its-relative-selector.type"/>
</define>
<define name="its-targetPointer.type">
  <ref name="its-relative-selector.type"/>
</define>
<define name="its-idValue.type">
  <data type="string" datatypeLibrary=""/>
</define>
<define name="its-space.type">
  <choice>
    <value>default</value>
    <value>preserve</value>
  </choice>
</define>
<define name="its-locQualityIssuesRef.type">
  <data type="anyURI"/>
</define>
<define name="its-locQualityIssuesRefPointer.type">
  <ref name="its-relative-selector.type"/>
</define>
<define name="its-locQualityIssueType.type">
  <choice>
    <value>terminology</value>
    <value>mistranslation</value>
```

```
      <value>omission</value>
      <value>untranslated</value>
      <value>addition</value>
      <value>duplication</value>
      <value>inconsistency</value>
      <value>grammar</value>
      <value>legal</value>
      <value>register</value>
      <value>locale-specific-content</value>
      <value>locale-violation</value>
      <value>style</value>
      <value>characters</value>
      <value>misspelling</value>
      <value>typographical</value>
      <value>formatting</value>
      <value>inconsistent-entities</value>
      <value>numbers</value>
      <value>markup</value>
      <value>pattern-problem</value>
      <value>whitespace</value>
      <value>internationalization</value>
      <value>length</value>
      <value>non-conformance</value>
      <value>uncategorized</value>
      <value>other</value>
    </choice>
  </define>
  <define name="its-locQualityIssueTypePointer.type">
    <ref name="its-relative-selector.type"/>
  </define>
  <define name="its-locQualityIssueComment.type">
    <data type="string" datatypeLibrary=""/>
  </define>
  <define name="its-locQualityIssueCommentPointer.type">
    <ref name="its-relative-selector.type"/>
  </define>
  <define name="its-locQualityIssueSeverity.type">
    <data type="double">
      <param name="minInclusive">0</param>
      <param name="maxInclusive">100</param>
    </data>
  </define>
  <define name="its-locQualityIssueSeverityPointer.type">
    <ref name="its-relative-selector.type"/>
  </define>
  <define name="its-locQualityIssueProfileRef.type">
    <data type="anyURI"/>
  </define>
  <define name="its-locQualityIssueProfileRefPointer.type">
    <ref name="its-relative-selector.type"/>
  </define>
  <define name="its-locQualityIssueEnabled.type">
    <choice>
      <value>yes</value>
      <value>no</value>
    </choice>
  </define>
  <define name="its-locQualityRatingScore.type">
    <data type="double">
      <param name="minInclusive">0</param>
      <param name="maxInclusive">100</param>
    </data>
  </define>
  <define name="its-locQualityRatingVote.type">
    <data type="integer"/>
  </define>
  <define name="its-locQualityRatingScoreThreshold.type">
    <data type="double">
      <param name="minInclusive">0</param>
      <param name="maxInclusive">100</param>
    </data>
  </define>
  <define name="its-locQualityRatingVoteThreshold.type">
    <data type="integer"/>
  </define>
  <define name="its-locQualityRatingProfileRef.type">
    <data type="anyURI"/>
  </define>
  <define name="its-mtConfidence.type">
    <ref name="its-confidence.type"/>
  </define>
  <define name="its-allowedCharacters.type">
    <data type="string" datatypeLibrary=""/>
  </define>
  <define name="its-allowedCharactersPointer.type">
    <ref name="its-relative-selector.type"/>
  </define>
  <define name="its-storageSize.type">
    <data type="nonNegativeInteger"/>
  </define>
  <define name="its-storageSizePointer.type">
    <ref name="its-relative-selector.type"/>
  </define>
  <define name="its-storageEncoding.type">
    <data type="string" datatypeLibrary=""/>
  </define>
  <define name="its-storageEncodingPointer.type">
    <ref name="its-relative-selector.type"/>
  </define>
  <define name="its-lineBreakType.type">
    <choice>
      <value>cr</value>
      <value>lf</value>
      <value>crlf</value>
    </choice>
  </define>
  <define name="its-annotatorsRef.type">
    <data type="string" datatypeLibrary=""/>
  </define>
</grammar>
```

[Source file: schemas/its20-types.rng]

(RELAX NG compact syntax version of schema)

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

Example 96: Schematron schema for ITS

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://purl.oclc.org/dsdl/schematron" queryBinding="xslt2">
  <ns uri="http://www.w3.org/2005/11/its" prefix="its"/>
  <ns uri="http://www.w3.org/1999/xlink" prefix="xlink"/>
  <pattern>
    <title>Indicating the Version of ITS</title>
    <rule context="*[@its:*]">
      <assert test="ancestor-or-self::*/@its:version | //its:rules/@version">
        The version is indicated by the ITS version attribute.
        This attribute is mandatory for the rules element, where it MUST be in no
        If there is no rules element in an XML document, a prefixed ITS version a
        (e.g. its:version) MUST be on the element where the ITS markup is used,
        or on one of its ancestors.</assert>
    </rule>
    <rule context="its:provenanceRecords | its:locQualityIssues">
      <assert test="self::*/@version | ancestor::*/@its:version | //its:rules/@ve
        The version is indicated by the ITS version attribute.
        This attribute is mandatory for the rules element, where it MUST be in no
        If there is no rules element in an XML document, a prefixed ITS version a
        (e.g. its:version) MUST be on the element where the ITS markup is used, o
        on one of its ancestors. For standoff markup unprefixed version attribute
    </rule>
    <rule context="*[@its:version]">
      <assert test="if (@its:version and //its:rules/@version)
        then //its:rules/@version = @its:version else true()">
        There MUST NOT be two different versions of ITS in the same document.</as
      <assert test="every $v in //*/@its:version satisfies $v = @its:version">
        There MUST NOT be two different versions of ITS in the same document.</as
    </rule>
    <rule context="its:provenanceRecords | its:locQualityIssues">
      <assert test="if (@version and //its:rules/@version)
        then //its:rules/@version = @version else true()">
        There MUST NOT be two different versions of ITS in the same document.</as
      <assert test="every $v in //*/@its:version satisfies $v = @version">
        There MUST NOT be two different versions of ITS in the same document.</as
    </rule>
  </pattern>
  <pattern>
    <title>Global, Rule-based Selection</title>
    <rule context="its:rules">
      <assert test="every $rules in //its:rules satisfies $rules/@version = curre
        If there is more than one rules element in an XML document, the rules fro
        to be processed at the same precedence level. The rules sections are to b
        and the ITS rules with them processed sequentially.
        The versions of these rules elements MUST NOT be different.</assert>
    </rule>
  </pattern>
  <pattern>
    <title>Link to External Rules</title>
    <rule context="its:rules[@xlink:href]">
      <assert test="count(doc(resolve-uri(@xlink:href, base-uri()))//its:rules) l
        The referenced document must be a valid XML document containing at most o
    </rule>
  </pattern>
  <pattern>
    <title>ITS Tools Annotation</title>
    <rule context="*[@its:annotatorsRef]">
      <assert test="every $ref in tokenize(@its:annotatorsRef, '\s+') satisfies
        matches($ref, '
        (translate|localization-note|terminology|directionality|language-informat
        elements-within-text|domain|text-analysis|locale-filter|provenance|extern
        target-pointer|id-value|preserve-space|localization-quality-issue|localiz
        mt-confidence|allowed-characters|storage-size)\|.+')">
        The value of annotatorsRef is a space-separated list of references where
        each reference is composed of two parts: a data category identifier and a
        These two parts are separated by a character | VERTICAL LINE (U+007C).</a
    </rule>
  </pattern>
  <pattern>
    <title>Source of confidence</title>
    <rule context="*[@its:termConfidence]">
      <assert test="ancestor-or-self::*[@its:annotatorsRef]
        [matches(@its:annotatorsRef, '.*\s*terminology\|.+')]">
        Any node selected by the terminology data category with the termConfidenc
        MUST be contained in an element with the annotatorsRef attribute
        specified for the Terminology data category.</assert>
    </rule>
    <rule context="*[@its:taConfidence]">
      <assert test="ancestor-or-self::*[@its:annotatorsRef]
        [matches(@its:annotatorsRef, '.*\s*text-analysis\|.+')]">
        Any node selected by the Text Analysis data category with the taConfidenc
        MUST be contained in an element with the annotatorsRef attribute
        specified for the Text Analysis data category.</assert>
    </rule>
    <rule context="*[@its:mtConfidence]">
      <assert test="ancestor-or-self::*[@its:annotatorsRef]
        [matches(@its:annotatorsRef, '.*\s*mt-confidence\|.+')]">
        Any node selected by the MT Confidence data category MUST be
        contained in an element with the annotatorsRef attribute
        specified for the MT Confidence data category.</assert>
    </rule>
  </pattern>
  <pattern>
    <title>Text analysis</title>
    <rule context="its:textAnalysisRule">
      <assert test="@taClassRefPointer | @taSourcePointer | @taIdentPointer | @ta
        Text analysis rule must specify at least target type class or target iden
    </rule>
  </pattern>
  <pattern>
    <title>Provenance standoff markup</title>
    <rule context="its:provenanceRecord">
      <assert test="@person | @personRef | @org | @orgRef | @tool | @toolRef | @r
        @revOrg | @revOrgRef | @revTool | @revToolRef | @provRef">
        At least one attribute must be specified on the provenanceRecord element.
    </rule>
  </pattern>
</schema>
```

[Source file: schemas/its20.sch]

Note:

In order to make it easy to integrate ITS markup into schemas based on W3C XML Schema language the following informative schemas are provided:

- its20.xsd – base schema for ITS
- its20-types.xsd – schema defining datatypes used in ITS markup

Please note that W3C XML Schema is less expressive then RELAX NG and some content models are more loose. A document can validate against W3C XML Schema while it is not conforming to ITS specification and it is not valid according to RELAX NG schema.

E Informative References

This section is informative.

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

Charmod Norm
Yergeau, François, Martin J. Dürst, Richard Ishida, Addison Phillips, Misha Wolf, Tex Texin. *Character Model for the World Wide Web 1.0: Normalization*. W3C Working Draft 1 May 2012. Available at http://www.w3.org/TR/2012/WD-charmod-norm-20120501/. The latest version of Charmod Norm is available at http://www.w3.org/TR/charmod-norm/ .

CheckMate Quality Check
Okapi Project. *CheckMate – Quality Check Configuration*. Available at http://www.opentag.com/okapi/wiki/index.php?title=CheckMate_-_Quality_Check_Configuration.

CSS 2.1
Bert Bos, Tantek Çelik, Ian Hickson Håkon Wium Lie. *Cascading Style Sheets, level 2 revision 1 CSS 2.1 Specification*. W3C Recommendation 7 June 2011. Available at http://www.w3.org/TR/2011/REC-CSS2-20110607/. The latest version of CSS2 is available at http://www.w3.org/TR/CSS21/.

DBpedia
DBpedia. Available at: http://dbpedia.org/OnlineAccess.

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 https://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 30042
(International Organization for Standardization). *TermBase eXchange (TBX)*. [Geneva]: International Organization for Standardization, 2008.

ISO/TS 11669:2002
(International Organization for Standardization). *Translation projects – General guidance*. [Geneva]: International Organization for Standardization, 2012.

ITS 1.0
Christian Lieske and Felix Sasaki. *?Internationalization Tag Set (ITS) Version 1.0?*. W3C Recommendation 03 April 2007. Available at http://www.w3.org/TR/2007/REC-its-20070403/. The latest version of ITS 1.0 is available at? http://www.w3.org/TR/its/.

ITS RDF
ITS RDF Ontology, version May 2013. Available at http://www.w3.org/2005/11/its/rdf# .

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/.

Microdata
Ian Hickson *HTML Microdata*. W3C Working Draft 25 October 2012. Available at http://www.w3.org/TR/2012/WD-microdata-20121025/.

MLW US IMPL
Christian Lieske (ed.). *Metadata for the Multilingual Web - Usage Scenarios and Implementations* . W3C Working Draft 7 March 2013. Available at http://www.w3.org/TR/2013/WD-mlw-metadata-us-impl-20130307/. The latest version of MLW Metadata US IMPL is available at http://www.w3.org/TR/mlw-metadata-us-impl/ .

Multidimensional Quality Metrics
Lommel, Arle. *Useful Quality Metrics (for Humans, Not Researchers)*. Presentation at the Workshop on UserCentric Machine Translation & Evaluation, MT Summit 2013.

NERD
Named Entity Recognition and Disambiguation ontology (NERD). available at: http://nerd.eurecom.fr/ontology

NIF
Hellmann, S. et al. (ed.). *NIF 2.0 Core Ontology*, as of August 2013. Available at http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core# under CC-BY 3.0 license maintained by the NLP2RDF project.

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 https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office. The latest version of OpenDocument is available at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office.

PROV-DM
Moreau, Luc and Paolo Missier (eds.). *Provenance data model*. W3C Recommendation 30 April 2013. Available at http://www.w3.org/TR/2013/REC-prov-dm-20130430/. The latest version of The PROV Data Model is available at http://www.w3.org/TR/prov-dm/.

RDFaLite
Manu Sporny (ed.). *RDFa Lite 1.1*. W3C Recommendation 07 June 2012. Available at http://www.w3.org/TR/2012/REC-rdfa-lite-20120607/.

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

Structured Specifications
(BYU Translation Research Group). *Structured Specifications and Translation Parameters*. Available at http://www.ttt.org/specs.

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

WordNet
WordNet. Princeton University, 2010. Available at: http://wordnet.princeton.edu.

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/.

XLIFF 1.2
Savourel, Yves, John Reid, Tony Jewtushenko and Rodolfo M. Raya. *XLIFF Version 1.2*. OASIS Standard 1 February 2008. Available at http://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html.

XLIFF 2.0
Comerford, Tom, David Filip, Rodolfo M. Raya and Yves Savourel. *XLIFF Version 2.0*. Committee Specification Draft 01 / Public Review Draft 01. Available at http://docs.oasis-open.org/xliff/xliff-core/v2.0/csprd01/xliff-core-v2.0-csprd01.html. The latest version of XLIFF 2.0 is available at http://docs.oasis-open.org/xliff/xliff-core/v2.0/xliff-core-v2.0.html

XML i18n BP
Yves Savourel, Jirka Kosek, Richard Ishida. *Best Practices for XML Internationalization*. Available at http://www.w3.org/TR/2008/NOTE-xml-i18n-bp-20080213/. 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 https://developer.mozilla.org/en-US/docs/XUL.

F Conversion to NIF

This section is informative.

This section provides an informative algorithm to convert XML or HTML documents (or their DOM representations) that contain ITS metadata to the RDF format based on [NIF]. The conversion results in RDF triples.

Note:

The algorithm creates URIs that in the query part contain the characters "[" and "]", as part of XPath expressions. In the conversion output (see an example), The URIs are escaped as "%5B" and "%5D". For readability the URIs shown in this section do not escape these characters.

Note:

The algorithm is intended to extract the text from the XML/HTML/DOM for an NLP tool. It can produce a lot of "*phantom*" predicates from excessive whitespace, which 1) increases the size of the intermediate mapping and 2) extracts this whitespace as text, and therefore might decrease NLP performance. It is strongly recommended to normalize whitespace in the input XML/HTML/DOM in order to minimize such phantom predicates. A normalized example is given below. The whitespace normalization algorithm itself is format dependent (for example, it differs for HTML compared to general XML).

Note:

The output of the algorithm shown below uses the ITS RDF ontology [ITS RDF] and its namespace
http://www.w3.org/2005/11/its/rdf#
Like the algorithm, this ontology is not a normative part of the ITS 2.0 specification and is being discussed in the ITS Interest Group.

Example 97: Example (see source code) of an HTML document with whitespace character normalization as preparation for the conversion to NIF. Note that text nodes in the `head` element are not taken into account.

```
<!DOCTYPE html><html xmlns="http://www.w3.org/1999/xhtml">
<head><meta http-equiv="Content-Type" content="text/html;charset=utf-8" >
<title>NIF conversion example</title></head>
<body><h2 translate="yes">Welcome to <span
    its-ta-ident-ref="http://dbpedia.org/resource/Dublin" its-within-text="yes"
    translate="no">Dublin</span> in <b translate="no" its-within-text="yes">Irela
```

The conversion algorithm to generate NIF consists of seven steps:

- STEP 1: Get an ordered list of all text nodes of the document.
- STEP 2: Generate an XPath expression for each non-empty text node of all leaf elements and memorize them.
- STEP 3: Get the text for each text node and make a tuple with the corresponding XPath expression (X,T). Since the text nodes have a certain order we now have a list of ordered tuples ((x0,t0), (x1,t1), ..., (xn,tn)).
- STEP 4 (optional): Serialize as XML or as RDF. The list with the XPath-to-text mapping can also be kept in memory. Part of a serialization example is given below. The upper part is in RDF Turtle Syntax while the lower part is in XML (the `mappings` element).

```
# Turtle example:
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> .
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> .
<http://example.com/myitsservice?informat=html&intype=url&input=http://example.co
 nif:wasConvertedFrom  <http://example.com/myitsservice?informat=html&intype=url&
<http://example.com/myitsservice?informat=html&intype=url&input=http://example.co
 nif:wasConvertedFrom <http://example.com/myitsservice?informat=html&intype=url&i
# ...
<http://example.com/myitsservice?informat=html&intype=url&input=http://example.co
 nif:wasConvertedFrom <http://example.com/myitsservice?informat=html&intype=url&i
<!-- XML Example -->
<mappings>
 <mapping x="xpath(x0)" b="b0" e="e0" />
 <mapping x="xpath(x1)" b="b1" e="e1" />
 <!-- ... -->
 <mapping x="xpath(xn)" b="bn" e="en" />
</mappings>
```

where

```
b0 = 0
e0 = b0 + (Number of characters of t0)
b1 = e0
e1 = b1 + (Number of characters of t1)
...
bn = e(n-1)
en = bn + (Number of characters of tn)
```

Example (continued)

```
# Turtle example:
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> .
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> .
# "Welcome to "
<http://example.com/myitsservice?informat=html&intype=url&input=http://example.co
 nif:wasConvertedFrom
 <http://example.com/myitsservice?informat=html&intype=url&input=http://example.c
# "Dublin"
<http://example.com/myitsservice?informat=html&intype=url&input=http://example.co
 nif:wasConvertedFrom
 <http://example.com/myitsservice?informat=html&intype=url&input=http://example.c
# " in "
<http://example.com/myitsservice?informat=html&intype=url&input=http://example.co
 nif:wasConvertedFrom
 <http://example.com/myitsservice?informat=html&intype=url&input=http://example.c
# "Ireland"
<http://example.com/myitsservice?informat=html&intype=url&input=http://example.co
 nif:wasConvertedFrom
 <http://example.com/myitsservice?informat=html&intype=url&input=http://example.c
# "!"
<http://example.com/myitsservice?informat=html&intype=url&input=http://example.co
 nif:wasConvertedFrom
 <http://example.com/myitsservice?informat=html&intype=url&input=http://example.c
# "Welcome to Dublin Ireland!"
<http://example.com/myitsservice?informat=html&intype=url&input=http://example.co
 nif:wasConvertedFrom
 <http://example.com/myitsservice?informat=html&intype=url&input=http://example.c
<!-- XML Example -->
<mappings>
 <mapping x="xpath(/html/body[1]/h2[1]/text()[1])" b="0" e="11" />
 <mapping x="xpath(/html/body[1]/h2[1]/span[1]/text()[1])" b="11" e="17" />
 <mapping x="xpath(/html/body[1]/h2[1]/text()[2])" b="17" e="21" />
 <mapping x="xpath(/html/body[1]/h2[1]/b[1]/text()[1])" b="21" e="28" />
 <mapping x="xpath(/html/body[1]/h2[1]/text()[3])" b="28" e="29" />
 <mapping x="xpath(/html/body[1]/h2[1])" b="0" e="29" />
</mappings>
```

- STEP 5: Create a context URI and attach the whole concatenated text `$(t0+t1+t2+...+tn)` of the document as reference.
- STEP 6: Attach any ITS metadata annotations from the XML/HTML/DOM input to the respective NIF URIs.
- STEP 7: Omit all URIs that do not carry annotations (to avoid bloating the data).

```
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> .
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#>
<http://example.com/myitsservice?informat=html&intype=url&input=http://example.co
    rdf:type             nif:Context ;
    rdf:type             nif:RFC5147String ;
# concatenate the whole text
    nif:isString         "$(t0+t1+t2+...+tn)" ;
    nif:beginIndex         "0" ;
    nif:endIndex         "29" ;
    itsrdf:translate     "yes";
    nif:sourceUrl      <http://example.com/doc.html> .
<http://example.com/myitsservice?informat=html&intype=url&input=http://example.co
```

```
    rdf:type              nif:RFC5147String ;
    nif:beginIndex         "11" ;
    nif:endIndex         "17" ;
    itsrdf:translate     "no";
    itsrdf:taIdentRef  <http://dbpedia.org/resource/Dublin> ;
    nif:referenceContext <http://example.com/myitsservice?informat=html&intype=ur
<http://example.com/myitsservice?informat=html&intype=url&input=http://example.co
    rdf:type              nif:RFC5147String ;
    nif:beginIndex         "21" ;
    nif:endIndex         "28" ;
    itsrdf:translate     "no";
    nif:referenceContext <http://example.com/myitsservice?informat=html&intype=ur
```

A complete sample output in RDF/XML format after step 7, given the input document Example 97, is available at examples/nif/EX-nif-conversion-output.ttl.

Note:

The conversion to NIF is a possible basis for a natural language processing (NLP) application that creates, for example, named entity annotations. A non-normative algorithm to integrate these annotations into the original input document is given in Appendix G: Conversion NIF2ITS. Many decisions to be made in this algorithm depend on the particular NLP application being used.

Note:

NIF allows an URL for a String resource to be referenced as URIs that are fragments of the original document in the form:
`http://example.com/myitsservice?informat=html&intype=url&input=http://example.com/doc.html&char=0,11`
or
`http://example.com/myitsservice?informat=html&intype=url&input=http://example.com/doc.html&xpath=/html/body[1]/h2[1]/text()[1]`
This offers a convenient mechanism for linking NIF resources in RDF back to the original document. The NIF Web Service Access Specification defines the parameters for NIF web services.

RDF treats URIs as opaque and does not impose any semantic constraints on the used fragment identifiers, thus enabling their usage in RDF in a consistent manner. However, fragment identifiers get interpreted according to the retrieved mime type, if a retrieval action occurs as is the case in Linked Data. The char fragment is defined currently only for text/plain while the xpath fragment is not defined for HTML. Therefore this URL recipe does fulfil the ITS requirements to support both XML and HTML and the aim of this mapping to produce resources adhering to the Linked Data principle of dereferenceablility. The future definition and registration of these fragment types, while a potentially attractive feature, is beyond the scope of this specification.

G Conversion NIF2ITS

This section is informative.

The following algorithm relies on Example 97. It is assumed that the example has been converted to NIF, leading to the output exemplified for the ITS2NIF conversion algorithm.

This example uses DBpedia Spotlight as an example natural language processing (NLP) tool. In it, DBpedia Spotlight linked "Ireland" to DBpedia:

```
<http://example.com/myitsservice?informat=html&intype=url&input=http://example.co
 rdf:type nif:RFC5147String;
 itsrdf:taIdentRef <http://dbpedia.org/resource/Ireland> .
<http://dbpedia.org/resource/Ireland>
 rdf:type <http:/nerd.eurecom.fr/ontology#Country> .
```

The conversion algorithm to generate ITS out of NIF consists of two steps:

- STEP 1: NIF Web services accept two different types of input. It is possible to either send the extracted text (the object of the `nif:isString` property) directly or NIF RDF to the NLP tool, i.e. the text is sent as a `nif:Context` node and included as `nif:isString`. Either way, the output of the Web service will be a NIF representation.

  Accepting text will be the minimal requirement of a NIF web service. Ideally, you would be able to send the `nif:Context` node with the isString as RDF directly, which has the advantage, that all other annotations can be used by the NLP tool:

  ```
  <http://example.com/myitsservice?informat=html&intype=url&input=http://exa
   rdf:type nif:RFC5147String ;
   rdf:type nif:Context ;
   nif:beginIndex "0" ;
   nif:endIndex "29" ;
   nif:isString  "Welcome to Dublin in Ireland!" .
  ```

- STEP 2: Use the mapping from ITS2NIF (available after step 7 of the ITS2NIF algorithm) to reintegrate annotations in the original ITS annotated document.

For step 2, three cases can occur.

CASE 1: The NLP annotation created in NIF matches the text node. Solution: Attach the annotation to the parent element of the text node.

```
# Based on:
<http://example.com/myitsservice?informat=html&intype=url&input=http://example.co
 nif:wasConvertedFrom
 <http://example.com/myitsservice?informat=html&intype=url&input=http://example.c
# and:
<http://example.com/myitsservice?informat=html&intype=url&input=http://example.co
 itsrdf:taIdentRef  <http://dbpedia.org/resource/Ireland> .
# we can attach the metadata to the parent node:
<b its-ta-ident-ref="http://dbpedia.org/resource/Ireland"
   translate="no">Ireland</b>
```

CASE 2: The NLP annotation created in NIF is a substring of the text node. Solution: Create a new element, e.g., for HTML "span". A different input example is given below as case 2 is not covered in the original example input.

```
# Input:
```

```
<html>
  <body>
     <h2>Welcome to Dublin in Ireland!</h2>
  </body>
</html>

# ITS2NIF

<http://example.com/myitsservice?informat=html&intype=url&input=http://example.co
 nif:wasConvertedFrom
 <http://example.com/myitsservice?informat=html&intype=url&input=http://example.c

# DBpedia Spotlight returns:

<http://example.com/myitsservice?informat=html&intype=url&input=http://example.co
    itsrdf:taIdentRef  <http://dbpedia.org/resource/Ireland> .

# NIF2ITS

<html>
  <body>
     <h2>Welcome to Dublin in <span
          its-ta-ident-ref="http://dbpedia.org/resource/Ireland">Ireland</span>!<
  </body>
</html>
```

Case 3: The NLP annotation created in NIF starts in one region and ends in another. Solution: No straight mapping is possible; a mapping can be created if both regions have the same parent.

H Localization Quality Guidance

This section is informative.

The Localization Quality Issue data category description uses the following terms as defined below for the purposes of this document.

- *Quality assessment*. The task of evaluating the quality of translated content to determine its quality and to assign a value to it. Localization quality assessment is commonly conducted by identifying, categorizing, and counting issues in the translated content.
- *Issue*. A quality issue is a potential error detected in content. Issues may be detected automatically (e.g., by using a grammar checker or translation-specific tool) or manually, by human checking of content. Issues may or may not be errors (e.g., an apparent mistranslation may be deliberate and appropriate in some contexts) and should be confirmed by review.
- *Metric*. A metric is a formal system used in quality assessment tasks to identify issues, evaluate them, and determine quality. Metrics provide specific reference points for categorizing issues (as opposed to subjective assessment of quality, which does not use a metric) and may include weights for issues.
- *Model*. A model is the underlying description of the system that underlies a metric. (For example, some models may allow variable weights to be assigned to different issue types, in which case the specific metric used for a task will have these weights defined, even though the underlying model does not.)
- *Profile*. A quality profile is the adaptation of a model to specific requirements. It specifies specific conditions for using a model. It may include instructions and other guidelines that are not included in the actual metric used. If a model allows for no customization, it has a

single profile that is identical to the model; if it allows customization, each customization is a distinct profile.

- *Review*. The task of examining a text to identify any issues that occur in it. Review may be tied to the task of fixing any issues, a task generally referred to as revision.
- *Specifications*. Specifications (sometimes called a translation brief) are a description of the various expectations and requirements for a translation task. These may include statements about the type of translation expected, guidance on terminology to be used, information about audience, and so forth. Translation specifications are described in detail in ISO/TS-11669.
- *Tool*. As used here, a tool is software that generates localization quality markup. Tools may be fully automatic (e.g., a tool that identifies potential issues with terminology and grammar and marks them without human intervention) or may required human input (e.g., a system that allows users to highlight spans of text and mark them with appropriate issues).

For more information on setting translation project specifications and determining quality expectations, implementers are encouraged to consult the ISO standard definition of translation project specifications included in [ISO/TS 11669:2002]. Details about translation specifications are available at [Structured Specifications]. While these documents do not directly address the definition of quality metrics, they provide useful guidance for implementers interested in determining which localization quality issue values should be used for specific scenarios.

The issue types defined in Localization Quality Issue were derived from the QTLaunchPad project's Multidimensional Quality Metrics (MQM) framework. Additional guidance on this project may be found at [Multidimensional Quality Metrics].

The topic of localization quality is rapidly evolving and ITS 2.0 represents the first step in standardizing this area and will serve for basic interoperability needs. For situations requiring additional expressive capability or categories, further custom markup may be required.

I List of ITS 2.0 Global Elements and Local Attributes

This section is informative.

The following table lists global ITS 2.0 elements inside `rules` element and local ITS 2.0 markup in XML and HTML. Note that for the local markup there are various constraints on what local attributes can be used together. Here these constraints are expressed via occurrence indicators: optional "?", alternatives "|", or groups "(...)". Please check the related sub sections in Section 8: Description of Data Categories defining local markup normatively.

In addition to below markup, ITS 2.0 provides a means to refer to the tools used to generate the markup: for XML the `annotatorsRef` attribute and for HTML the `annotators-ref` attribute. See Section 5.7: ITS Tools Annotation for details, especially the note on annotatorsRef usage scenarios.

Data category	Global element inside `rules` element	Local XML attributes in ITS namespace	HTML attributes
Translate	`translateRule`	`translate`	`translate`
Localization Note	`locNoteRule`	(`locNote` | `locNoteRef`), `locNoteType`?	(`its-loc-note` | `its-loc-note-ref`), `its-loc-note-type`?
Terminology	`termRule`	`term`, `termInfoRef`?, `termConfidence`?	`its-term`, `its-term-info-ref`?,

W3C Recommendation

Data category	Global element inside rules element	Local XML attributes in ITS namespace	HTML attributes
			its-term-confidence?
Directionality	dirRule	dir	dir
Language Information	langRule	xml:lang	lang
Elements Within Text	withinTextRule	withinText	its-within-text
Domain	domainRule	-	-
Text Analysis	textAnalysisRule	taConfidence?, at least one of (taClassRef, ((taSource, taIdent) | taIdentRef))	its-ta-confidence?, at least one of (its-ta-class-ref, ((its-ta-source, its-ta-ident) | its-ta-ident-ref))
Locale Filter	localeFilterRule	localeFilterList	its-locale-filter-list
Provenance	provRule	(at least one of ((person | personRef), (org | orgRef), (tool | toolRef), (revPerson | revPersonRef), (revOrg | revOrgRef), (revTool | revToolRef), provRef)) | provenanceRecordsRef	(at least one of ((its-person | its-person-ref), (its-org | its-org-ref), (its-tool | its-tool-ref), (its-rev-person | its-rev-person-ref), (its-rev-org | its-rev-org-ref), (its-rev-tool | its-rev-tool-ref), its-prov-ref)) | its-provenance-records-ref
External Resource	externalResourceRefRule	-	-
Target Pointer	targetPointerRule	-	-
ID Value	idValueRule	xml:id	id
Preserve Space	preserveSpaceRule	xml:space	-

W3C Recommendation

Data category	Global element inside rules element	Local XML attributes in ITS namespace	HTML attributes
Localization Quality Issue	locQualityIssueRule	(at least one of (locQualityIssueType, locQualityIssueComment), locQualityIssueSeverity?, locQualityIssueProfileRef?, locQualityIssueEnabled?) | locQualityIssuesRef	(at least one of (its-loc-quality-issue-type, its-loc-quality-issue-comment), its-loc-quality-issue-severity?, its-loc-quality-issue-profile-ref?, its-loc-quality-issue-enabled?) | its-loc-quality-issues-ref
Localization Quality Rating	-	(locQualityRatingScore, locQualityRatingScoreThreshold?) | (locQualityRatingVote, locQualityRatingVoteThreshold?), locQualityRatingProfileRef?	(its-loc-quality-rating-score, its-loc-quality-rating-score-threshold?) | (its-loc-quality-rating-vote, its-loc-quality-rating-vote-threshold?), its-loc-quality-rating-profile-ref?
MT Confidence	mtConfidenceRule	mtConfidence	its-mt-confidence
Allowed Characters	allowedCharactersRule	allowedCharacters	its-allowed-characters

Data category	Global element inside rules element	Local XML attributes in ITS namespace	HTML attributes
Storage Size	storageSizeRule	storageSize, storageEncoding?, lineBreakType?	its-storage-size, its-storage-encoding?, its-line-break-type?

J Revision Log

This section is informative.

The following log records major changes that have been made to this document since the ITS 2.0 Proposed Recommendation 24 September 2013:

1. In response to Working Group discussion and AC review, fixed an error in Appendix D: Schemas for ITS and made the section informative.
2. Added an informative reference to [Multidimensional Quality Metrics] and reformatting of a few references.
3. Editorial fixes, see related mail.

K Acknowledgements

This document has been developed with contributions by the MultilingualWeb-LT Working Group and collaborators: Mihael Arcan (DERI Galway at the National University of Ireland, Galway, Ireland), Pablo Badía (Linguaserve), Aaron Beaton (Opera Software), Renat Bikmatov (Logrus Plus LLC), Aljoscha Burchardt (German Research Center for Artificial Intelligence (DFKI GmbH)), Nicoletta Calzolari (CNR--Consiglio Nazionale delle Ricerche), Somnath Chandra (Department of Information Technology, Government of India), John Colosi (Verisign, Inc.), Mauricio del Olmo (Linguaserve), Giuseppe Deriard (Linguaserve), Pedro Luis Díez Orzas (Linguaserve), David Filip (University of Limerick), Leroy Finn (Trinity College Dublin), Karl Fritsche (Cocomore AG), Serge Gladkoff (Logrus Plus LLC), Tatiana Gornostay (Tilde), Daniel Grasmick (Lucy Software and Services GmbH), Declan Groves (Centre for Next Generation Localisation), Manuel Honegger (University of Limerick), Dominic Jones (Trinity College Dublin), Matthias Kandora (]init[), Milan Karásek (Moravia Worldwide), Jirka Kosek (University of Economics, Prague), Michael Kruppa (Cocomore AG), Alejandro Leiva (Cocomore AG), Swaran Lata (Department of Information Technology, Government of India), David Lewis (Trinity College Dublin), Fredrik Liden (ENLASO Corporation), Christian Lieske (SAP AG), Qun Liu (Centre for Next Generation Localisation), Arle Lommel (German Research Center for Artificial Intelligence (DFKI GmbH)), Priyanka Malik (Department of Information Technology, Government of India), Shaun McCance ((public) Invited expert), Sean Mooney (University of Limerick), Jan Nelson (Microsoft Corporation), Pablo Nieto Caride (Linguaserve), Pēteris Ņikiforovs (Tilde), Naoto Nishio (University of Limerick), Philip O'Duffy (University of Limerick), Des Oates (Adobe Systems Inc.), Georgios Petasis (Institute of Informatics & Telecommunications (IIT), NCSR), Mārcis Pinnis (Tilde), Prashant Verma Prashant (Department of Information Technology, Government of India), Georg Rehm (German Research Center for Artificial Intelligence (DFKI GmbH)), Phil Ritchie (VistaTEC), Thomas Rüdesheim (Lucy Software and Services GmbH), Nieves Sande (German Research Center for Artificial Intelligence (DFKI GmbH)), Felix Sasaki (DFKI / W3C Fellow), Yves Savourel (ENLASO Corporation), Jörg Schütz (W3C Invited Experts), Sebastian Sklarß (]init[), Ankit Srivastava (Centre for Next Generation Localisation), Tadej Štajner (Jozef Stefan Institute), Olaf-Michael Stefanov ((public) Invited expert), Najib Tounsi (Ecole Mohammadia d'Ingenieurs Rabat (EMI)), Naitik Tyagi Tyagi (Department of Information Technology, Government of India), Stephan Walter (Cocomore AG), Clemens Weins (Cocomore AG).

A special thanks goes to the following persons:

- Sebastian Hellmann for introducing us to [NIF] and for contributing to the creation of the ITS 2.0 ontology and NIF testing.
- Daniel Naber for introducing us to LanguageTool and for implementing Localization Quality Issue Type functionality in language tool.