D2.1: REQUIREMENTS AND USE CASES DOCUMENTS

Dave Lewis (TCD), Arle Lommel (DFKI), Felix Sasaki (DFKI)

Distribution: Public

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

FP7-ICT-2011-7

Project no: 287815

Document Information

Deliverable number:	2.1
Deliverable title:	Requirements and Use Cases Document
Dissemination level:	PU
Contractual date of delivery:	31st August 2012
Actual date of delivery:	24th May 2012
Author(s):	Dave Lewis, Arle Lommel, Felix Sasaki
Participants:	DFKI, TCD
Internal Reviewer:	W3C MLW-LT Working Group
Workpackage:	WP2
Task Responsible:	Dave Lewis
Workpackage Leader:	Dave Lewis

The deliverable has been published as an HTML file at

http://www.w3.org/TR/2012/WD-its2req-20120524/

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

W3C Working Draft

Requirements for Internationalization Tag Set (ITS) 2.0

W3C Working Draft 24 May 2012

This version:
http://www.w3.org/TR/2012/WD-its2req-20120524/
Latest version:
http://www.w3.org/TR/its2req/
Editors:
Dave Lewis (TCD)
Arle Lommel (DFKI)
Felix Sasaki (DFKI / W3C Fellow)

Abstract

This document gathers metadata proposed within the MultilingualWeb-LT Working Group for the Internationalization Tag Set Version 2.0 (ITS 2.0). The metadata targets web content (mainly HTML5) and deep Web content, for example content stored in a content management system (CMS) or XML files from which HTML pages are generated, that facilitates its interaction with multilingual technologies and localization processes.

Status of this Document

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

This document is a First Public Working Draft published by the MultilingualWeb-LT Working Group, part of the W3C Internationalization Activity. The Working Group expects to advance this Working Draft to Working Group Note (see W3C document maturity levels).

This document describes use cases for ITS 2.0 metadata. By publishing this working draft the working group does not express any consensus about the implementation approach, the use cases described or the proposed metadata items. The main purpose of this publication is to gather feedback from a wider audience. Comments discussed within the working group are marked as **COMMENT:**. A list of issues discussed within the Working Group is available.

Feedback about the content of this document is encouraged. See also issues discussed within the Working Group. Send your comments to public-multilingualweb-lt-comments@w3.org. Use "Comment on ITS 2.0 requirements WD" in the subject line of your email. The archives for this list are publicly available.

W3C Working Draft

Publication as a Working Draft does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

This document was produced by a group operating under the 5 February 2004 W3C Patent Policy. The group does not expect this document to become a W3C Recommendation. 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 Purpose of this Document
  - 1.2 Who should read this
  - 1.3 Terminology and Metadata Approach
  - 1.4 Implementation Approach
  - 1.5 Feedback
    - 1.5.1 Requirements Questionnaire
    - 1.5.2 Requirements Assessment
- 2 Glossary of Terms
  - 2.1 Key Defnitions
  - 2.2 Product Classes Implementing Requirements
  - 2.3 Use Case Actor Roles
- 3 Use Cases
  - 3.1 Authoring
  - 3.2 Automatic enrichment of the source content with named entity annotations
  - 3.3 CMS-Localization Exchange
  - 3.4 Quality Assurance (QA)
  - 3.5 Translation (Pre-QA)
  - 3.6 Translation Provenance and Quality Metadata
  - 3.7 Translation (Post-QA)
  - 3.8 CMS-Side Revision Management
  - 3.9 Publication Decision Support
  - 3.10 Real Time Translation Systems (RTTS)
- 4 Overview of proposed metadata categories
  - 4.1 Visualization
  - 4.2 Tabular Overview
  - 4.3 Identification of Language and Locale
- 5 Descriptions of proposed metadata categories
  - 5.1 Internationalization
    - 5.1.1 autoLanguageProcessingRule
    - 5.1.2 directionality
    - 5.1.3 locale-filter
    - 5.1.4 idValue
    - 5.1.5 ElementsWithinText
    - 5.1.6 preserveSpace
    - 5.1.7 ruby
    - 5.1.8 targetPointer
    - 5.1.9 translate
    - 5.1.10 localization note
    - 5.1.11 language information
  - 5.2 Process
    - 5.2.1 readiness
    - 5.2.2 progress-indicator
    - 5.2.3 cacheStatus

        - 5.3 Project Information
            - 5.3.1 domain
            - 5.3.2 formatType
            - 5.3.3 genre
            - 5.3.4 purpose
            - 5.3.5 register
            - 5.3.6 translatorQualification
        - 5.4 Provenance
            - 5.4.1 author
            - 5.4.2 contentLicensingTerms
            - 5.4.3 revisionAgent
            - 5.4.4 sourceLanguage
            - 5.4.5 translationAgent
        - 5.5 Quality
            - 5.5.1 qualityError
            - 5.5.2 qualityProfile
        - 5.6 Translation
            - 5.6.1 confidentiality
            - 5.6.2 context
            - 5.6.3 externalPlaceholder
            - 5.6.4 languageResource
            - 5.6.5 mtConfidence
            - 5.6.6 specialRequirements
        - 5.7 Terminology
            - 5.7.1 disambiguation
            - 5.7.2 namedEntity
            - 5.7.3 terminology
            - 5.7.4 textAnalysisAnnotation
- 6 Requirements
    - 6.1 Support ITS 1.0 Data Categories
    - 6.2 Limited Impact
    - 6.3 Round-trip interoperability with XLIFF 1.2
    - 6.4 Compatibility with multiple source content formats
    - 6.5 Optimize execution of ITS processing rules
    - 6.6 Removal, Archiving and Reintegration of ITS mark-up
    - 6.7 Process Model
- 7 References
- 8 Acknowledgements

1 Introduction

1.1 Purpose of this Document

This document gathers metadata proposed within the MultilingualWeb-LT Working Group for the Internationalization Tag Set Version 2.0 (ITS 2.0). The metadata is used to annotate web content (referred to henceforth just as content) to facilitates its interaction with multilingual technologies and localization processes with the aim of publishing that content on the Web in multiple languages. In this context, content can refer to static web content in HTML or XHTML, deep web content, for example content stored in a content management system (CMS) or XML files from which HTML or XHTML pages are generated.

1.2 Who should read this

The target audience for this document includes the following categories:

- Developers of localization tools
- Localizers involved with Web content

- Developers of language technology applications (e.g. machine translation) that are part of or that make use of the Web
- Developers and users of CMS systems
- Developers of authoring tools for Web content
- Authors of Web content
- Designers of content-related schema, e.g. XML based formats like DocBook or DITA
- Developers of Internet specifications at the World Wide Web Consortium and related bodies

Since a lot of the terminology is not known across communities, this document contains a glossary of terms.

1.3 Terminology and Metadata Approach

Following the terminology introduced in the Internationalization Tag Set (ITS) 1.0 specification, ITS 2.0 metadata items are called data categories. Data categories are defined conceptually (e.g. Translate). In ITS 1.0, they are implemented in XML, see the implementation for Translate. ITS 2.0 will provide additional definitions and offer implementations at least for HTML5.

To lower burden on implementors and to foster adoption, the data categories are proposed as in independent items. See the section on support of ITS 1.0 data categories for more details.

1.4 Implementation Approach

The MultilingualWeb-LT working group currently plans the following implementation approach.

- Conceptual, prose definitions of data categories will be given as in the ITS 1.0 specification.
- The implementation for HTML5 will rely on lower cased, custom attributes in HTML5 prefixed with its-, eg.: ... (Note that the prefix *its-* itself might still change). This approach is taken from the extensibility section of the HTML5 specification.
- In addition, the working group will provide an algorithm to convert its- attributes into RDFa and Microdata markup, to serve the needs of the Semantic Web community and of search engine optimization.
- The conversion to RDFa will add URIs to each metadata item in an HTML5 document. This is needed as reference points for the metadata items after extraction of RDF.
- In XML, the its- prefixed attributes will have a counterpart in a dedicated namespace. The ITS namespace http://www.w3.org/2005/11/its/ is under consideration.

1.5 Feedback

Please send feedback to the public-multilingualweb-lt-comments list (archive).

At the current stage, the working group has gathered a long list of potential ITS 2.0 data categories. We especially welcome feedback on the following aspects:

- Feasibility of the metadata approach and the implementation approach described above.
- Who is willing to implement a given data category in applications?
- What data categories can be merged with other data categories in the list?
- What data categories need to be defined more clearly?
- What usage scenarios and existing or to be created implementations are important for specific data categories?
- What types of content is in need of these data categories: HTML, XML, CMS configuration files, XLIFF, etc.

The working group will gather feedback until **end of June 2012**. This feedback will be the basis for creating the first draft of the data category standard definition. After June 2012, this document (the "requirements document") will not be updated anymore.

Requirements are used to define the set of data categories to be addressed in the standard definition which is due for a **feature freeze November 2012**. The WG has closed the open gathering of requirements by the end of April 2012, and has performed an initial round of consolidation for a working draft of the requirements document to be published for 20th May. The WG will continue a process of requirements consolidation, such that a prioritised and consistent set of data category requirements is available by the end of June 2012. A major milestone in this process will be an open requirements workshop to be conducted in Dublin 12-13 June.

1.5.1 Requirements Questionnaire

A public consultation questionnaire has been executed, resulting in 17 responses. A summary of results has been produced that assesses responses against current state of requirements.

1.5.2 Requirements Assessment

A requirements assessment conducted 4th May 2012, and is now being supplemented with implementation commitments. This will guide the prioritisation of work on the different data categories.

2 Glossary of Terms

2.1 Key Defnitions

The following terms common in multilingual technologies and localization processes are used in this document:

localization
: See http://www.w3.org/International/questions/qa-i18n#l10n

internationalization
: See http://www.w3.org/International/questions/qa-i18n#i18n

source language
: Refers to the language in which content is originally authored. Content is a source language is sometime referred to as source content.

target language
: Refers to the language into which source content is translated.

language service provider (LSP)
: An organisation offering commercial translation and localisation services.

locale
: A specific target market with known language, cultural and other requirements for the publication of content.

language service client
: An organisation making use of the services of an LSP to convert content from a form suitable for one locale to a form suitable for one or more other locales. In the context of localisation processes, sometime referred to just as the 'client'.

2.2 Product Classes Implementing Requirements

To clarify the product classes impacted by ITS 2.0 requirements, and referenced by use cases, the following classes are identified:

Content Authoring Tool
: Used by content authors to generate source language content and associated internationalisation mark-up. This class includes: tools for authoring static HTML/XHTML; authoring tools integrated into CMS and authoring tools producing XML files that are converted by CMS or XML stylesheet transforms into HTML/XHTML documents.

Source Quality Assurance (QA) Tool
: Used to assess the conformance of source content to style, controlled language, terminology and internationalisation guidelines.

Content Management System (CMS)
: Used to manage multiple content files or content components from authorship to publication, including version control and archiving.

Translation Management System
: Manages the localisation workflow process, collecting and distributing source and target content and associated language resources such as translation memories, term bases, context information and translation guidelines.

Computer Assisted Translation (CAT) tool
: Used by translators to improve productivity of content translation and translation post-editing. May include features such as TM match, terminology/glossary lookup, machine translation, concordancing, access to external reference and context material and in-context (WISYWIG) preview/editing.

Translation QA Tool
: Used for checking and reporting the quality of translations in relation to translation guidelines.

Machine Translation Service
: Online services that is used to automatically transform source language content into target language content.

Text Analytics Service
: Online services used to automatically generate annotations to specific pieces of content based on automated analysis of their lexical and semantic properties.

Web Browsers
: Applications that render HTML and XHTML documents for users.

2.3 Use Case Actor Roles

The following are descriptions of potential roles for use case actors that benefit from the use of data categories:

Content Author
: Author of web content.

Content Consumer
: User who reads translated content and may offer some feedback on its usefulness or quality if given the opportunity

Terminologist
: Working for the content generating organisation, this person is responsible for identifying terminology in the source content, cataloguing it so that it can receive consistent treatments and ensuring consistent translations are available in required target languages.

Client-based Localisation Manager
: Manages content localisation, either by passing content to be localised to an LSP or by invoking translation services directly on content held on the client's systems. Typically an employee of the organisation that owns the content.

Client-based Translator/Posteditor
: A translator who translates or post-edits suggested MT or TM translations text segments or terms presented via a specialised interface to a client's systems. Could be a professional translator or a volunteer working on a crowd-sourced translation project.

Client-based Translation Reviewer
: A bi-lingual person who provides a quality assessment of translated text, presented via a specialised interface to the client's system, at granularities from individual terms or segments up to a set of documents. Could be a professional reviewer or a volunteer working on a crowd-sourced translation project.

LSP-based Translation Process Manager
: A manager responsible for: the extraction of text to be translated from the client's systems; its preparation for translation; its machine translation and/or TM-matching; the packaging of provisional translation, source, source context and any relevant TMs or term-bases; the distribution of packages to translators; the monitoring of translation/postediting progress; and the collection of completed translation for return to client.

LSP-based Translation Review Process Manager
A manager responsible for: the extraction of translated text from a CMS; its the packaging of translation, source, source context and any relevant TMs or term-bases; the distribution of packages to reviewers; the monitoring of review progress; and the collection of completed completed reviews and the assembly of a report for the client.

LSP-based Translator/Posteditor
A professional translator who directly translates or post-edits suggested MT or TM translations of text segments or terms presented via a CAT tool.

LSP-based Translation Reviewer
A professional linguist(?) who provides a quality assessment of translated text, presented via a CAT tool, at granularities from individual terms or segments up to a set of documents.

Machine Translation (MT) service provider
The developer and operator of software systems that provide an MT service. Typically responsible for the ongoing reconfiguration/retraining of the service.

Text Analytics (TA) service provider
The developer and operator of software systems that provide an TA service. Typically responsible for the ongoing reconfiguration/retraining of the service.

CMS developer
The developer of CMS platform software.

Localisation Tool developer
The developer of software systems that support translation and postediting, multilingual terminology management, translation review and localisation workflow management.

System Integrator
A software developer contracted to develop plugins or connectors that interface two or more software systems sources from separate third parties.

Search Engine Web Crawler
An automated agent that crawls multilingual web pages in order to index them for search engine providers.

3 Use Cases

ITS 2.0 will support several business scenarios around the production of multilingual web content and the operation of localisation processes over web content. The following use case description serve as a broad orientation. Some use cases are linked to proposed data categories. More links will be created in a subsequent version of this document or in the to be published ITS 2.0 draft.

3.1 Authoring

Content authors can add internationalization meta-data to documents or document fragments that they are authoring. This metadata helps to ensure that content is translated correctly and in way that is appropriate to its intended use. It can also ensure that content is not translated unnecessarily, that certain terms are translated in a prescribed way and that special care is taken in translating specific content. Communicating this via meta-data reduces downstream content processing costs, reduces the likelihood of translation errors and improves the assurance of quality of translations. In all these cases, metadata items may be added, either automatically or manually by the content creator using content authoring tools.

- Relevant metadata:
- Since authoring is a central process for ITS 2.0 data categories, the following list only provides a broad overview.
    - author
    - purpose
    - translate
    - register
    - special requirements
    - …
- See note on CMS-level data category scoping

3.2 Automatic enrichment of the source content with named entity annotations

This use case elaborates on the mention of automatic meta data annotation Content Authoring use case description. The automation of meta-data annotation reduces the manual cost of annotation and may increase the accuracy, consistency and comprehensiveness of such annotations.

- The enrichment of source content with named entity annotations is one example of such an automatic process.
- To realize this use case, tooling is already available and will be tailored by working group participants. One main tool in this respect is the Enrycher tool. Enrycher adds metadata for semantic and contextual information. Using named entity extraction and disambiguation can provide links from literal terms to concrete concepts, even in ambiguous circumstances. These links to concepts can be used to indicate whether a particular fragment of text represents a term, whether it is of a particular type, and alternative terms that can be used for that concept in other languages. Concretely, Enrycher uses DBPedia to serve as a multilingual knowledge base in order to map concepts to terms in foreign languages. Given that it also outputs the type of the term even if the exact term is not known, it can still serve as input to translation rules that apply to specific term types (personal names, locations, etc.).
- The annotation procedure with Enrycher is implemented as an additive enrichment of HTML5 markup.

3.3 CMS-Localization Exchange

In localization, it is common that content is created by a client and then processed in the following manner:

- The client sends content (defined by client-based localization manager) to the LSP or indicates that content available on the client's systems is ready to be localised.
- The LSP obtains the content and localises it.
- The localized content is re-integrated into the client's systems. This process should also be triggered by the client-based localization manager (and not be ‘injected’ by the LSP), typically subject to some QA review conducted by the client, the translating LSP or a third party LSP.
- In this scenario, metadata such a content identifier specifying the position of translated content to be re-integration into the broader content document structures needs to be provided.
- Specialised file formats that contain the same content in multiple languages are often used for exchanging source and target content between systems such as CMS, TMS and CAT tools, that participate localisation processes. An important international standard in this regard is the OASIS XML Localisation Interchange File Format (XLIFF). The conversion of content format to XLIFF and back again, so-called *XLIFF roundtripping* is an important class of implementations for this use case.
- The accurate automated conversion of content files to multilingual localization file formats reduces file handling costs associated with file handling and reduces any associated errors or loss of content. It must however maintain the binding of certain meta-data to both source and target content.

3.4 Quality Assurance (QA)

QA metadata can be applied to either content documents or sub-document fragments (for example, some portions of a document may have been previously proofread and it is useful to know which parts need attention and which do not). These metadata support the systematic review of a document to identify any linguistic errors (e.g., mistranslations, typographic errors, text inappropriate left untranslated, grammatical errors, stylistic errors). This can help ensure QA consistency when more than one individual is involved in provision and assessment of translation (i.e., situations other than self-assessment), where information about the translation

process is needed. For example, an LSP may provide a translation, which gets sent to another LSP for review (and optionally returned to the first vendor for correction).

- Relevant metadata:
  - translate
  - author
  - purpose
  - links to related information (reference documentation, previously translated materials)
  - terminology
  - translation agent
  - proofread as part of the process model
  - quality, including type of error and error severity
  - conformance score/conformance rank (COMMENT: currently unmapped to a data category; needs an explanation) This relates to some statistical quality assurance technology VistaTEC is using technology from partner Digital Linguistics called Review Sentinel. Underlying technology is licensed from Trinity College, Dublin. See http://www.digitallinguistics.com

3.5 Translation (Pre-QA)

The translator uses the translate data category, related information and definitions during translation to improve the quality and accuracy of the translation and to ensure only the required content is translated (reducing wasteage of translation effort).

- The translator may add metadata to signify the content that was human translated.
- A lot of metadata is relevant for translation. The following is only a small subset. Pre-QA means that this metadata is relevant before a quality assurance step has taken place.
  - translate
  - author
  - purpose
  - terminology
  - translation agent
  - process model, esp. proofread state
  - localization note

3.6 Translation Provenance and Quality Metadata

The way in which content has been translated is often important. It is often necessary to distinguish between content that has been human translated, machine translated or results from human post-editing of machine translation.

- Client localisation managers may want to be assured that received target translation has been subjected to some human checking or postediting if that had been the contracted requirement.
- LSP-based Translation Review Process Manager may want to differentiate solely machine translated target content from human mediated translation when managing review processes and assigning QA review guidelines
- MT creators are unable to effectively discern human authored, full high quality translation content from non-reviewed automatically generated noise. They need to be able to control the MT training sets based on information describing the quality and which process has been used to create it.
- Serch engine web crawlers may rank translated content differnetly depending on whether it was just machine translated or subject to some human postediting or QA to maintain quality of search results
- Relevant metadata
  - source language
  - translate (*translate* attribute preserved so the process knows if the text was translated or not)
  - translation agent

    - MT confidence score (for raw MT output)
    - information on how the text was revised/post-edited

3.7 Translation (Post-QA)

- After a quality assurance process, translator fix errors and signify that all content has been re-verified. After the POST-QA process, the content is reintegrated into the original source, e.g. a CMS, an XML file or other types of content.

3.8 CMS-Side Revision Management

- For revision after the localization process, information about the time of translation and last revisions is useful. With this information, it can be decided whether the content should be published or whether a reversion to previous versions is necessary. Content managers shall be able to identify unsatisfactory content to be transmitted back to the LSP.

3.9 Publication Decision Support

- The content manager should be able to make decisions about publication depending on various pieces of information, such as:
    - (a) MT and/or human translation
    - (b) level and type of QA
    - (c) level of completion of translation process

3.10 Real Time Translation Systems (RTTS)

Real Time Translation Systems (RTTS) are systems that provides synchronous translation of basically two types:

- Interoperable, when the RTTS gives back the translation in real time and the client server published the content in the web by their means.
- Publishing, when the RTTS takes the source content already published by the client in the web and performs the translation and the publishing, both in real time.

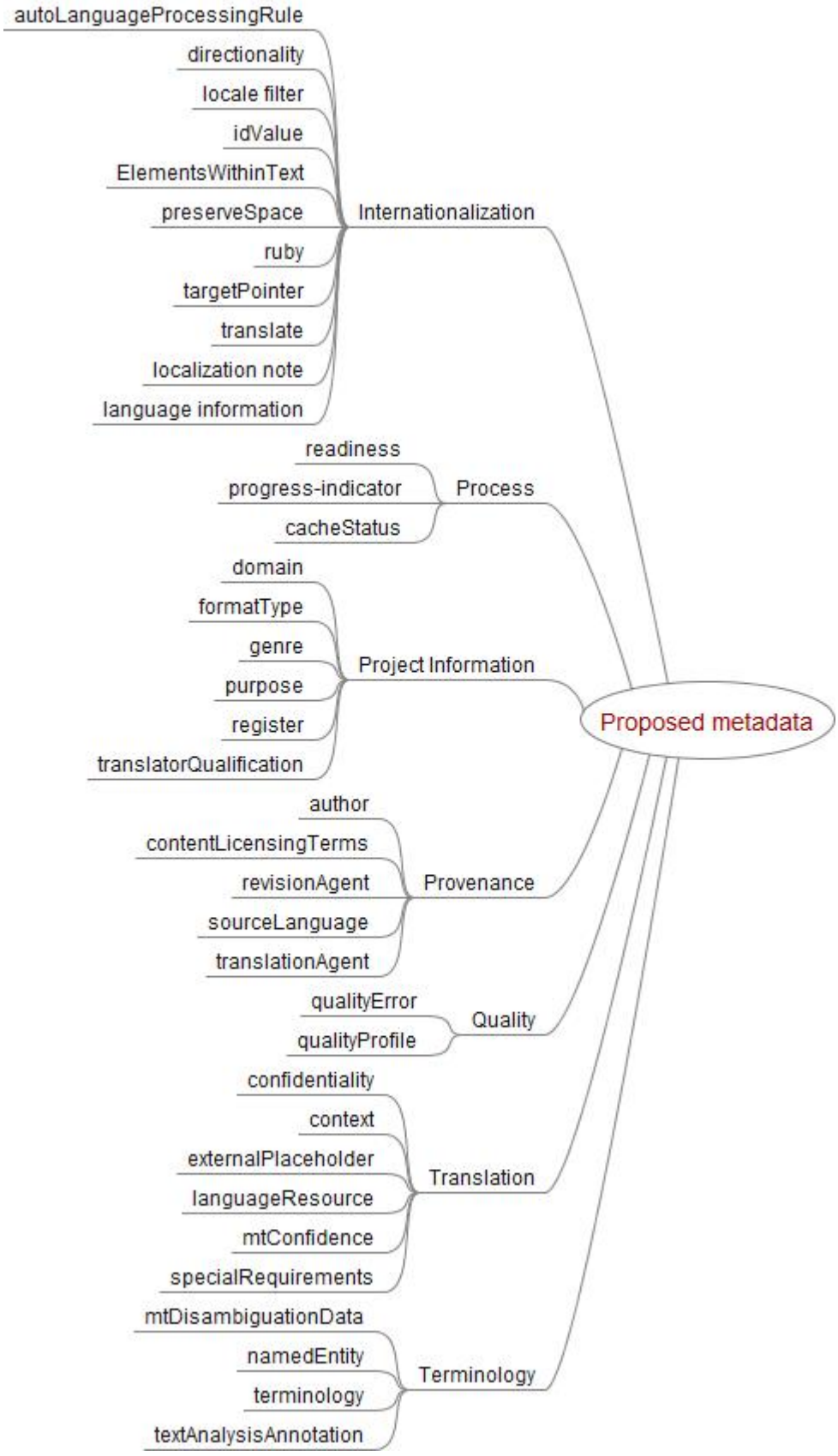
Both approaches need metadata to be used in the synchronous machine translation phase, the synchronous automatic publishing phase and the asynchronous post-editing or terminological tasks for quality machine translation improvements.

- Relevant metadata:
- TBC

4 Overview of proposed metadata categories

4.1 Visualization

The following figure provides a broad overview of the proposed data categories.


Proposed metadata
Internationalization
autoLanguageProcessingRule
directionality
locale filter
idValue
ElementsWithinText
preserveSpace
ruby
targetPointer
translate
localization note
language information
Process
readiness
progress-indicator
cacheStatus
Project Information
domain
formatType
genre
purpose
register
translatorQualification
Provenance
author
contentLicensingTerms
revisionAgent
sourceLanguage
translationAgent
Quality
qualityError
qualityProfile
Translation
confidentiality
context
externalPlaceholder
languageResource
mtConfidence
specialRequirements
Terminology
mtDisambiguationData
namedEntity
terminology
textAnalysisAnnotation

4.2 Tabular Overview

The table below lists proposed metadata elements with a brief description and statement about which level(s) they apply to (document = applies to the entire document, element = applies to defined elements in the document, span = applies to user/tool-defined spans). Links go to more detailed information below. For a table showing which data categories are needed by which work packages, see this document.

Name	Short description	Level
Internationalization		
autoLanguageProcessingRule	This data category captures information that it is acceptable to create target language content purely based on automated language processing (such as automated transliteration, or machine translation).	span
directionality	Improve handling of ITS directionality rules	element, span
locale filter	provides instruction that content should be excluded from translated version (not just untranslated, but deleted) in all cases or for specified locales	element, span
idValue	mechanism to associate ITS translateRule with unique IDs	element
ElementsWithinText	Provide a way to identify elements nested within other elements	element
preserveSpace	identifies whether white space should be preserved in the translation process	document, span
ruby	Improve ITS ruby model	span
targetPointer	identifies relationship between source and target in a file at the element level, e.g., specifies that the translation for a <source> element goes in a <target> element	document, element, span
translate	specifies whether the content of the element to which the attribute is applied should be translated or not	document, span
localization note	used to communicate notes to localizers about a particular item of content	document, span
language information	used to express the language of a given piece of content	document, span
Process		
readiness	provides positive guidance regarding steps to be undertaken in a CMS/localization process	document, span
progress-indicator	reports the proportion of a document that has completed by a process	document
cacheStatus	indicates need to (re)translate dynamic web content for real time MT	document, span
Project Information		
domain	information about the domain (subject field) of the content	document, span
formatType	provides information about the format or service for which the content is produced (e.g., subtitles, spoken text)	document, span
genre	information about the genre (text type) of the content	document, span

purpose	information about the purpose of the text	document, span
register	information about stylistic/register requirements (e.g., formality level)	document, span
translatorQualification	information about the qualifications required for the translator	document, span
Provenance		
author	provides information about the author of content (= dc:author)	
contentLicensingTerms	Licensing terms for content (e.g., can it be used in databases or for TM?)	document, span
revisionAgent	provides information concerning how a text was revised (e.g., human postediting)	document, span
sourceLanguage	provides information concerning what language the original text was in	document, span
translationAgent	provides information concerning how a text was translated (e.g., MT, HT)	document, span
Quality		
qualityError	describes an authoring or translation error	span
qualityProfile	describes the profile/results of a language-oriented quality assurance task	document, element, span
Translation		
confidentiality	States whether text is confidential (and thus cannot be exposed to public translation services)	document, element, span
context	Provides information about where the text occurs (e.g., in a button, a header, body text)	element, span
externalPlaceholder	Provides instructions for translators on how to deal with external resources	element
languageResource	states what translation-oriented languages resource(s) is/are to be used	document, span
mtConfidence	Information provided by an MT engine concerning its confidence in the result	span
specialRequirements	information about any special localization requirements (e.g., string length, character limitations)	span
Terminology		
mtDisambiguation	Information required to assist MT to distinguish between ambiguous cases	span
namedEntity	Values for types of named entities,	span
terminology	marking of information about terms used in the content	span
textAnalysisAnnotation	embed information generated by text analysis services	span

4.3 Identification of Language and Locale

In this document, **language** is identified via BCP 47 language tags.

Locale information is based on UTC #35, with the following approach to convert a language tag to a locale identifier:

- The hypen that separate subtags within a language tag are converted to underscore following the process described at

http://unicode.org/reports/tr35/#BCP_47_Language_Tag_Conversion

- Implementations of ITS 2.0 are not expected to process the "u" extension for further locale information as defined in RFC 6067.

An example language tag is de-de. An example locale is de_de.

Both language tags and locale identifiers are case insensitive and are written in lower case throughout this document.

5 Descriptions of proposed metadata categories

5.1 Internationalization

These categories relate primarily to the internationalization of content and are generated prior to translation (and may be consumed in translation). Includes any items that build on existing ITS functionality.

5.1.1 autoLanguageProcessingRule

Indicates how the span should be treated during automatic translation. This features goes beyond the *translate* category to provide instruction for cases where text should be transliterated rather than translated.
Data model
Possible values:
- transliteration
- machineTranslation

Notes
- source on ITS 1.0 wiki
- COMMENT: this proposed data category might be changed to be specific to the process of Transliteration - feedback is welcome on this proposal.

Example
- **Stellaris** is a brand name and should transliterated into Japanese as ステルラリス.

5.1.2 directionality

HTML5 brings new features to directionality. The ITS 1.0 feature should be updated to reflect the changes.

5.1.3 locale-filter

ITS 2.0 should support the indication of source content elements as only being suitable for inclusion for localisation to specific locales only, for not being suitable for localisation to specific locales or for not being suitable for localisation at all
Use Cases
localise a Swiss legal notice only in "de_ch;fr_ch;it_ch"
Data model
locale-filter-type : (positive|negative|none)
- "none" indicates that the element should not be passed for localization under any circumstances
- "positive" means the element MAY ONLY be localised for the locales specified in locale-filter-list
- "negative" means the element MUST NOT be localised for the locales specified in the locale-filter-list

locale-filter-list : list of locale identifiers

5.1.4 idValue

Using identifiers with content is a very common activity in localization and follows the best practices for internationalization (See http://www.w3.org/TR/xml-i18n-bp/#DevUniqueID). For example unique IDs can be used to leverage the same translation from one version of the document to another, or to align content between two versions.
The XML attribute xml:id is the standard way of representing an identifier in ITS 1.0 (See http://www.w3.org/TR/xml-i18n-bp/#AuthUniqueID. However, in some case the document may be using other attributes, and could be in non XML formats.
Such ID value must be persistent from one version of the content to the next, and, ideally, it should be globally unique. If it cannot be globally unique it should be unique value at the document level.
Ideally the mechanism should allow to build 'complex' values based on different parts of the document (e.g. attributes element or event hard-coded text.
For example, in the XML 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 you have a unique identifier for both the content of <text> and the content of <desc>, you can combine the value of the parent's id with the elements' name to obtain the values "id1_text" and "id1_desc" for the <text> and <desc> element respectively. (See Example A below).

Data model

to be determined

Notes

- There is an existing proposal for an idValue attribute for the translateRule of ITS 1.0 that may or may not be a solution: http://www.w3.org/International/its/wiki/IssuesAndProposedFeatures#Proposal:_idValue
- Such an ID value would also enable a number of other data categories either through rule references or through external reference to the span from stand off meta-data. A similar approach was taken in xml:tm.
- Such id value would be mapped to the XLIFF 'resname' attribute. XLIFF makes a distinction between 'id' and 'resname'. IDs are tool-specific and, while they can be the same as the 'resname', they do not necessarily persistent across different version of the document, or can even differ depending on the extraction options used on the same document.

```
Example A:

<doc>
 <msg name="id1">
  <text>Content of text</text>
  <desc>Context of desc</desc>
 </msg>
</doc>

--> Corresponding XLIFF output:

<trans-unit id='1' resname='id1_text'>
 <source>Content of text</source>
</trans-unit>
<trans-unit id='2' resname='id1_desc'>
 <source>Content of desc</source>
</trans-unit>
```

5.1.5 ElementsWithinText

ITS2.0 should support the elements within text data category from ITS1.0 and should consider the following extension for **local** elements within text.
There is no local rule for the "Element Within Text" data category. Having a local rule would allow ITS processor without XPath support to still identify element nested or within text from other elements.

Data model

Possibly, a locale attribute `withinText` with a value `yes|no|nested` (See Example A)

Notes

- See the definition for the Elements Within Text data category in ITS 1.0. That definition was only implemented as a global rule in ITS 1.0.
- There is a proposed solution for this listed here: :http://www.w3.org/International/its/wiki/IssuesAndProposedFeatures#Proposal:_Local_.22Elements_within_Text.22

```
Example A

<text
 xmlns:its="http://www.w3.org/2005/11/its"
 xmlns:itsx="http://www.w3.org/2008/12/its-extensions"
 its:version="1.0">
 <body>
  <par>Text with <bold itsx:withinText='yes'>bold</bold>.</par>
 </body>
</text>
```

5.1.6 preserveSpace

Knowing whether the white spaces in a given element (especially the line-breaks) are collapsible or not is important for proper segmentation and matching when using computer assisted translation tools.
There are two main types of white space usages:

- Text formatted for reasons not related to the final presentation of the document. For example a paragraph "pretty-printed" (See Example A below).
- Text where white spaces are meaningful. For example, where line-breaks can be segment-breaks and/or spaces is the only way to format the final output (See Example B below).

It is important for translation tools to make a difference between those two cases (text can be collapsed safely) and the last two (text should not be collapsed).
The indication of whether white spaces should be preserved or not should be accessible from the document itself, as defining the information at the rendering level (e.g. in a CSS style-sheet) may not be accessible for the translation tool.

Data model

to be determined

Notes

- The xml:space="preserve" attribute may provide a solution for some of these requirements at the document instance level.
- The xml:space attributes defines only "preserve" and "default", "default" not being necessarily "do-not-preserve", but means "do-whatever-you-want". Do we have situations where "do-not-preserve" would be needed?
- The whiteSpace constraint defined in the XML Schema Part 2 (http://www.w3.org/TR/xmlschema-2/#rf-whiteSpace) may provide a solution for these requirements at the schema level.
- The white-space property in CSS2 (http://www.w3.org/TR/CSS2/text.html#white-space-prop) defines a more complex set of values than xml:space. Are they needed?

- There is an existing proposal for a whitespace attribute for the translateRule of ITS 1.0 that may or may not be a solution: http://www.w3.org/International/its/wiki/IssuesAndProposedFeatures#Proposal:_whiteSpaces
- There is an existing extension to ITS that implements a solution for the preservation of white spaces: See itst:preserveSpaceRule in http://itstool.org/extensions/

```
Example A:

<para>This is the first
      sentence of the paragraph. It's followed
      by a second sentence.</para>
```

```
Example B:

<data name="CMD_USAGE">
 <value>Usage: po2xliff input[ options[ output]]
Where options are:
    -trg: create target entries
   -fill: fill the target entries with the source text</value>
</data>
```

5.1.7 ruby

The ITS 1.0 ruby model is based on the XHTML ruby specification. ITS 2.0 will update the ruby model to refer to HTML5. The related discussion is ongoing in the I18N Core working group.

Notes

- see ITS 1.0 Ruby data category

5.1.8 targetPointer

Various proprietary file formats (e.g. software resources, localization formats) store two or more language versions of the same text. Such format cannot be processed easily with a traditional XML filter because there is currently no way in ITS 1.0 to indicate where the *target* text for a given *source* is. (See Example A and B).

Data model

- Possibly an XPath expression that selects the node where the target is located relatively the location of its corresponding source. (See Example C). This addresses only cases for a single target.

Notes

- Real life use case here: http://tech.groups.yahoo.com/group/okapitools/message/2672
- There is an existing proposal for a targetPointer attribute for the translateRule of ITS 1.0 that may or may not be a solution: http://www.w3.org/International/its/wiki/IssuesAndProposedFeatures#Proposal:_targetPointer

```
Example A:

<file>
 <entry xml:id="one">
  <source>Text one of the source</source>
  <target>Text one of the target</target>
 </entry>
 <entry xml:id="two">
  <source>Text two of the source</source>
  <target></target>
 </entry>
</file>
```

```
Example B:

<file>
 <entry id='1'>
  <text loc='1'>Very important text</text>
  <text loc='2'>Texte très important</text>
  <text loc='3'>非常重要的文本</text>
  <text loc='4'>Zeer belangrijke tekst</text>
  <text loc='5'>Очень важный текст</text>
 </entry>
</file>
```

```
Example C:

<its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0"
 xmlns:itsx="http://www.w3.org/2008/12/its-extensions">
 <its:translateRule translate="no" selector="//file"/>
 <its:translateRule translate="yes" selector="//source"
  itsx:targetPointer="../target"/>
</its:rules>
```

5.1.9 translate

Specifies whether content should be translated or not

Data Model

- yes
- no

Notes

- Already implemented in html5 and in ITS 1.0 for XML content. ITS 2.0 will define how to apply this to CMS or other types of content.

5.1.10 localization note

Notes

- Based on ITS 1.0

5.1.11 language information

Notes

- Based on ITS 1.0

5.2 Process

These categories are used primarily for controlling or indicating the state of the content production process.

- COMMENT: The naming convention used here is inconsistent: some of the categories use "Status" and others "State". We should be consistent.

5.2.1 readiness

ITS2.0 should be able to indicate the readiness of an element for submission to different processes or provide an estimate of when and element will be ready for a particular process.

ITS2.0 should be able to indicate the relative priority elements should be subjected to when submitted to a process.
ITS2.0 should be able to indicate an expectation of when an a specific process should be completed for an element.
ITS2.0 should be able to specify if an element previously submitted to a process has subsequently been revised and therefore needs to be re-submitted to that process.

Data model

ready-to-process
the type of the next process requested

process-ref
a pointer to an external set of process type definitions used for ready-to-process if the default value set is not used

ready-at
defines the time the content is ready for the process, it could be some time in the past, or some time in the future

revised
(yes|no) - indicates is this is a different version of content that was previously marked as ready for the declared process

priority
(high|low) - should we should keep this simple?

complete-by
provides a target date-time for completing the process

Notes

- COMMENT: this combines previous data categories: processTrigger, legalStatus, processState, proofreadingState and revision state
- COMMENT: the definition of the process model is now extracted into a separate requirements under the subject process-model, since it applies now to several data categories
- COMMENT: The following attribute are relevant if the process type of 'ready-to-process' was of the class *translate*:
  - *contentType*, values: MIME or custom values - This indicates the format or the type of the content used in the content in order to apply the right filter or normalization rules, and the subsequent processes. For example, to express HTML we could use: “contentType: text/html
  - *sourceLang* – value: standard ISO 639 value - this value indicates the source language for the current translation requested. It is different from the sourceLanguage (provenance) Data Category , since this indicates the language the original source text was and sourceLang indicates the current source language to be used for the translation that can be different from the original source - **this should be considered as an attribute for proveance**
  - *contentResultSource* –value: yes / no. Indicates the format if the Localisation chain needs to give back the original
  - *contentResultTarget* – value: monolingual, multilingual; indicates if the resulting translation, in the cases of several target languages, should be delivered in several monolingual content files or in a single multilingual content file
  - *pivotLang* - value: standard ISO value. Indicates the intermediate language in the case is needed. Two examples: 1) Going from a source language to two language variants (eg. into Brazil and Portugal Portuguese), it is more cost-effective to go to one first (being this first variant a "pivot" language) and to revise later to the second variant; Going from one language to another via an intermediate language (eg. from Maltese into English and from English into Irish, because there is not direct Maltese into Irish available translation).
- COMMENT: There seems to be a not insignificant overlap with ISO/TS 11669 in this case. For the sake of consistency we should try to consolidate with that standard where possible.

5.2.2 progress-indicator

ITS 2.0 must be able to convey a simple indication of the proportion of a specified process that has been completed
Data model
progress-of-process : a process name
progress-indicator : 0-100
progress-units : (sentence|words) default: sentence
Notes

5.2.3 cacheStatus

Provides an indication of the status of the source and target(s) texts in a system cache for use by real-time translation, TMS, etc. to determine when retranslation is needed. A timestamp can be used to determine when the content was cached.
Examples:

- The original content is not saved in the cache (i.e., it is new or has been updated): (re)translation is needed
- The translated content is not saved in the cache (i.e., it has not been previously translated or has expired): translation is needed
- Neither the original nor the translated page are saved in the cache: both need to be cached

Data model

- **cache** - values: yes, no;
- **scope** - values: source, target, both
- **timestamp** - date and time

Notes

- COMMENT: I would suggest for the date and time that we use one of the following. I believe the first is better as it is more easily readable for humans and is ISO standards-based.
  - UTC + ISO 8601 (e.g., “20120405T060000” = April 5, 2012 at 06:00:00 UTC)
  - The Unix time stamp (e.g., "1333605600" = April 5, 2012 at 06:00:00 UTC).
- COMMENT: XML Schema data types for date and time might be better, e.g. 2012-04-05T060000

5.3 Project Information

These categories provide information about the project that may be useful for controlling processes, but they do not convey or control process state themselves.

5.3.1 domain

Specifies the domain of the text
Data model
text string
Notes

- It needs to be decided what ontology of domains to be used.
- COMMENT: should this be just a pointer to a concept node in an ontology, accompanied by a pointer to the ontology? This would need some conformance statement on the form of the ontology, but Semantic web ontologies naturally support this.
- COMMENT: A standard list of values has been suggested, but this seems hard to achieve.
- COMMENT: There might be a need to support multiple domains. For example, a text about the history of Russian legal reforms will have domain-specific content from at

least two domains (history, legal) that cannot be united into a single hierarchy. We need to think about the structure to support this.

Examples

- <meta name="its-domain" content="computer-aided design" /> (document level)
- […] (element level)

5.3.2 formatType

provides information about the format or service for which the content is produced (e.g., subtitles, spoken text)

Data model

to be determined

5.3.3 genre

information about the genre (text type) of the content

Data model

to be determined

Notes

- COMMENT: separate but related to *domain*

Examples

- <meta name="its-genre" content="advertising" /> (document level)

5.3.4 purpose

Information about the purpose of the text (e.g., advertising, educational)

Data model

to be determined

Notes

- Equivalent to Linport *purpose* (parameter 1d)

5.3.5 register

Defines the register expectations for the translation (e.g., formal)

Data model

picklist: (intimate|informal|consultative|formal|frozen) (Taken from Joos 1961)

Notes

- The original description stated it was for “style”, but the description was for register
- Corresponds to Linport *register* (parameter 10)
- COMMENT: There is no scholarly agreement on register divisions. The listing above is somewhat accepted for English, but would not always work for other languages.

Examples

- ** **In the courtroom proceedings in Thomas v. Thomas, Judge Thomson maintained that the Biblical statement ****“Thou shalt not commit adultery”**** was still considered the law of the land.** **

5.3.6 translatorQualification

Information about any qualifications required of the the translator

Data model

text string

Notes

- Corresponds to Linport *qualifications* (parameter 20a)
- Impossibly to enumerate all possible values. Primarily useful for human decision making processes.

Example

- <meta name="its-translatorQualification" content="certified English to Hungarian with expertise in musicology" />

5.4 Provenance

These categories provide a record of the origin of information and the agents that have acted on it.

5.4.1 author

provides information concerning the author of content
Data model
Description of author, to be defined
Notes
- COMMENT: Is this equivalent to dc:author?

5.4.2 contentLicensingTerms

MT creator should be aware not only of process and quality metadata but also about a legal provenance metadata. This would use RDF license linking mechanism. The aim is to provide machine readable information about content licensing terms and their implementation in MT related processes. In reference implementations, business rules should be defined to automatically include or not include data in training corpora, based on provided licensing information.
See also [http://www.meta-net.eu/whitepapers/meta-share/licenses META-SHARE work on language resource licensing
Data model
to be defined

5.4.3 revisionAgent

provides information concerning how a text was revised (e.g., human postediting)
Data model
Description of agent, to be defined
Notes
- Needs information on the action of the revisor as well, e.g., the degree of postediting: light, moderate, full.

5.4.4 sourceLanguage

provides information concerning what language the original source text was in
Data model
language/locale ID

5.4.5 translationAgent

provides information concerning how a text was translated (e.g., MT, human translation)
Data model
- type: (human|machine|social)

Notes
- COMMENT: Do we want to allow more granularity, e.g., some way to say "this was translated by Bing Translator v. 1.0.2" or "translated using SDL Trados Studio 2011"? If we do this, we make it harder to process the values. If we stick with the type values, we simplify decisions about how to trust the results.

5.5 Quality

These categories are used for explicit quality assurance steps undertaken on content (source or target).

5.5.1 qualityError

Describes the nature and severity of an error detected during a language-oriented quality assurance (QA) process

Data model

Note that the content of this element may be a span or may be empty in the case where an error does not enclose a span of content (e.g., something is missing in the content).

- **type**? (text) the type (name) of the rule that was violated, as defined in the **ruleSet** attribute. If no ruleSet attribute is present, the default value of "LISA QA Model" is assumed. (This parameter is optional since this element could be used with only the "note" attribute present for manual tasks where no formal system is used or where a manual note is added.)
- **ruleSet**? (text) the rule set referred to. If this parameter is used, the value should correspond to a rule declared in a *ruleSetName* attribute in the **qualityProfile** metadata category.
- **severity**? (text) the severity assigned to the error, if the QA profile uses severity. Note that the content of this attribute is native to the particular QA system and is not normalized.
- **note**? (text) contains any note text added in the QA process
- **agent**? (text) string identifying the agent responsible for adding the data

Example

(Assumes a declared QA Profile of “SAE J2450”)

- *The ****verbs agrees**** with the subject.*

Notes

- While any established metric may be used (or none at all if the "type" and "ruleSet" attributes are omitted), the default is to use the LISA QA Model, which seems to have the most general currency in the translation and localization industry, and other metrics should be declared in the **qualityProfile** data category.
- In principle, this can be used without any attributes at all as a pure marker, e.g., *The **<qualityError>**verbs agrees**</qualityError>** with the subject.*. However, inclusion of the attributes makes this data category more useful for real actions.
- COMMENT: Should we look at having a catalog of recognized rule sets that could be declared in this element without the need for the **qualityProfile** as a separate metadata element? That would promote data portability since individual pieces of content could be copied without an external reference? Then the dependency on **qualityProfile** would exist only if the user wants to declare a profile that is not in the catalog.
- COMMENT: I combined the previous score and weight items into severity. Since this applies to single errors, the score is by definition 1, but the severity is variable. The earlier formulation confused score and severity.
- COMMENT: If [severity] should be a number then it definitively should be an integer. With floats you will have to deal with rounding errors.

5.5.2 qualityProfile

Defines a source QA profile applied to the entire document, a section of a document, or an element, and, optionally, the results of that model

Data model

- **name**? (text) The name used to refer to this rule set in the document. If undefined the value of "LISA QA Model" is assumed.
- **uri**? The URI where the rule set can be found (if available).
- **pass**? The status of whether the content has passed the check. Suggested values include: pass, fail, warning
- **score**? The score or error count (whichever is appropriate) returned by the QA rule.
- **agent**? text description of the part responsible for supplying the score/pass.

Example

(This example assumes that the data category is declared as a meta element, but I'm sure there are better ways to handle this.

- <meta name="qualityProfile" content="name:LISA QA Model;uri:http://www.example.com;pass:no;score:85%" agent="ABCReview" />

Notes

- At least one of the attributes must be used (otherwise the data category is empty).
- COMMENT: Can be used without qualityError where this presents a summary of QA activities that are not tagged (e.g., when a reviewer uses the LISA QA Model software, which does not permit local tagging of errors).
- "agent" is defined in both this and qualityError. When declared here, agent is global in scope to indicate who did an assessment; when declared locally, it applies only to the local scope and overrides a global declaration.

5.6 Translation

These categories are used or generated in the translation process. (There is some conceptual overlap with Internationalization that we may want to resolve)

5.6.1 confidentiality

States whether the text can be submitted to public services (e.g., online MT engines or not)

Data model

- (confidential|nonconfidential)

Notes

- Any more complex confidentiality requirements (e.g., a statement that something is top-secret, community, corporate, etc.) would be handled by separate negotiation between the parties and are not covered here.

5.6.2 context

Provides information about where the text occurs (e.g., in a button, a header, body text)

Data model

- *picklist (to be defined)*
- grouping category (see ITS 1.0 wiki for details)

Notes

- Corresponds partially to the *termLocation* data category in TBX:

A location in a document, computer file, or other information medium, where the term frequently occurs, such as a user interface object (in software), a packaging element, a component in an industrial process, and so forth. The element content shall be expressed in plainText, and preferably be restricted to a set of picklist values. The following picklist values are recommended for software user interface locations in a Windows environment.

- checkBox
- comboBox
- comboBoxElement • dialogBox
- groupBox
- informativeMessage • interactiveMessage • menuItem
- progressBar
- pushButton
- radioButton

    - slider
    - spinBox
    - tab
    - tableText
    - textBox
    - toolTip
    - user-definedType
- Source in ITS wiki

5.6.3 externalPlaceholder

instructions on how to deal with external resources (e.g., graphics files) in translation. Derived from itst:externalRefRule
Data model
See the itst:externalRefRule description

5.6.4 languageResource

Identified what language resource(s) are to be used for translation memory, MT lexicon look-up, terminology management, and similar tasks
Data model

- type{1,1}: picklist with the values (terminology|lexicon|corpus|TM)
- location{1,1}: uri of resource
- format{1,1}: text string identifying the format (e.g., "Multiterm", "TBX", "TMX") (see notes)
- id{1,1}: id value used to bind other metadata to this item
- description{0,1}: text description for human consumption

Notes

- COMMENT: suggestion that we use MIME-types for *format* declaration, declaring private types if needed.
- COMMENT: there are not public MIME-type declarations for many common formats and we might need to allow "other" as an option.
- COMMENT: the issue of format needs to be resolved. I personally like the idea of MIME types if they work.

5.6.5 mtConfidence

used by MT systems to indicate their confidence in the provided translation
Data model

- a numeric value between 0.0 and 1.0

5.6.6 specialRequirements

Any special requirements about the translation/localization (e.g., string lengths, character limitations)
Data model
to be defined

5.7 Terminology

Data Categories related to the association of content with terminological data.

5.7.1 disambiguation

Includes data to be used by MT systems in disambiguating difficult content

Data model

- **concept reference**: points to a concept in an ontology that this fragment of text represents. May be an URI or an XPath pointer.
- **alternative labels**: expressions of that concept in other languages.
- **disambiguation data**: plain text content to be used by the system. This content is not defined and may be application specific, e.g., a code used by the system, a synonym, a pointer to a location in an ontology (Tadej - to remove in favor of concept reference)
- **semantic selector**: plain text content to be used by the system. This content is not defined and may be application specific, e.g., a code used by the system, a synonym, a pointer to a location in a semantic network. (Tadej - to remove in favor of concept reference)

Notes

- Text analysis should depend solely on the source language, independent of the target languages.
- The component must provide additive markup, preserving the original structure of the document.
- The annotations should refer to a span of content
- It should support the use case of terminology or concept translation via identification of these concepts;
- It should support the use case of marking up input data for training MT systems
- It should be extensible to support ontologies describing the linguistic properties of the fragments
- Open system to be used by MT, for example by the following applications :
    - domain selector: it can be used to select the dictionary or specialization that applies. (Tadej - to remove in favor of the 'domain' data category)
    - semantic selector: it can be used by disambiguation rules or filters. (Tadej - to remove in favor of concept reference)

5.7.2 namedEntity

When describing a fragment of text that has been identified as a named entity, we would like to specify the following pieces of information in order to help downstream consumers of the data, for instance when training MT systems

Data model

- **Entity type**: a pointer (URI) to a concept, defining a particular type of the entity. The entity URI space is open. Current recommendations for domains are the NERD ontology (Named Entity Disambiguation and Recognition) http://nerd.eurecom.fr/ontology and schema.org (http://schema.org/docs/full.html)
- Uses properties from 'disambiguation', if the entity is recognized;

5.7.3 terminology

Identification and marking of terms in content with associated information

Data model

* **Terminology lexicon** : the lexicon, used in the term reference. See subsection 'terminology resource';
* Uses properties from 'disambiguation' in order to point to concrete terms;

Notes

- The concept reference is translaable from ITS1.0 termInfo property: it can be identified by a URI (its:termInfoRef) or optionally XPath (its:termInfoPointer, its:termInfoRefPointer)
- COMMENT: should keep relevant standards (e.g., ITS 1.0 term data category, TBX, OLIF) in mind.

5.7.4 textAnalysisAnnotation

This data category allows the results of text analysis to be annotated in content.

Data Model

- **Annotation agent** - which tool produced the annotation.
- **Confidence score** - what is the system's confidence for this annotation, on the range of [0.0, 1.0].

6 Requirements

6.1 Support ITS 1.0 Data Categories

MLW-LT must support all ITS 1.0 data categories and their functionality, using the following approach:

- It will adopt the use of data categories to define discrete units of functionality.
- It will adopt the separation of data category definition from the mapping of the data category to a given content format
- It will adopt the conformance principle of ITS1.0 that an implementation only needs to implement one data category to claim conformance to the successor of ITS 1.0.
- A data category implementation only needs to support a single content format mapping in order to support a claim of MLW-LT conformance
- MLW-LT will specify implementations of data categories in the following: HTML5, XML
- MLW-LT will support all the ITS1.0 data category definitions
- Where ITS1.0 data categories are implemented in XML, the implementation must be conformant with the ITS1.0 mapping to XML to claim conformance to the successor of ITS 1.0

Notes

- ITS1.0 merely mentions XPATH 1.0 or its successor, but XPath 2.0 is the newest version and there are backward compatibility issues to consider

6.2 Limited Impact

- All solutions proposed should be designed to have as little impact as possible on the tree structure of the original document and on the content models in the original schema.

6.3 Round-trip interoperability with XLIFF 1.2

- Solutions should be able to be passed back and forth with XLIFF 1.2 with no data loss
- For the implementation approach envisaged, this means that its-* attributes would be converted in XLIFF to the relevant namespace, e.g. its-term in HTML5 to its:term.

6.4 Compatibility with multiple source content formats

- Solutions must work with many XML/HTML source formats

6.5 Optimize execution of ITS processing rules

- See *itst:match* at http://itstool.org/extensions/
- Proposed on the MLW-LT member mailing list

6.6 Removal, Archiving and Reintegration of ITS mark-up

- It should be possible to remove ITS 2.0 markup from a document without altering it original state - this may be useful when localization is deemed complete and the markup would be an overhead for publishing

- It should be possible to archive ITS 2.0 markup removed from a file in a form that it can be reintegrated into the file at a later date, e.g. if re-translation or revision is unexpectedly required
- These requirements extrapolated from CMS community feedback - see: http://lists.w3.org/Archives/Public/public-multilingualweb-lt/2012Apr/0011.html

6.7 Process Model

ITS2.0 should support an explicit expression of the process model to which ITS2.0 conformant content can be subjected.
The process model may be references from several currently proposed data categories, including readiness, progress-indicator and provenance
A default process model referenced from an ITS standard may be useful in clarifying different use cases for various data categories
The process model should be flexible, so that a subset can be configured to document a conformant implementation of ITS2.0
The process model should be extensible, so that an extended version can be used to document a conformant implementation of ITS2.0 subject to agreement on the definitions of extensions
Data model
Model 1

- *proposed for original process Trigger* Encodes the actions or workflow item requested (i.e., what should be triggers). The values could be user defined, since it is hard to generalize a set or combinations of actions for specific workflows. Some possible values are:
  - *contentQuote* - indicates that a quoting or pricing is requested, not to perform the job
  - *contentAlignment* - in case the content is to add to a Translation Memory (?)
  - *contentL10N* - localize the content
  - *contentI18N* - internationalize the content
  - *contentDtp* - desktop publishing of content
  - *contentSubtitle* - subtitling of content
  - *contentVoiceOver* - voice-over of content
-
  - *sourceRewrite: rewrite the source content (needs contentResultSource* - yes)
  - *sourceReview: review the source content (needs contentResultSource* - yes)
  - *sourceTranscribe: transcribe the source content (needs contentResultSource* - yes)
  - *sourceTransliteration: transliterate the source content (needs contentResultSource* - yes)
-
  - *hTranslate* - human translation
  - *mTranslate* - machine translation
  - *hTranscreate* - human transcreation
-
  - *posteditQA* - human postediting of mTranslate
  - *reviewQA* - human review for quality assurance only the target text, without the source text (see UNE 15038 “review”), by an expert for instance
  - *reviseQA* - human revision for quality assurance examining the translation and comparing source and target (see UNE 15038 “revision”)
  - *proofQA* - human checking of proofs before publishing for quality assurance (see UNE 15038 “proofreading”)
- **Model 2**
- *Proposed as part of table produced to analyse generation and consumption of data categories by different*
- this uses a hierarchical definition of processes, which may be useful for offering some flexibility and extensibility properties to the model
- *Generation of Source Content*
- *Translation*
- *Consumption of Translated Content*

7 References

A references section will be provided in a future version of this document.

8 Acknowledgements

This document has been created by participants of the MultilingualWeb-LT Working Group.

Members of the Working Group are (at the time of writing, and by alphabetical order): Mihael Arcan (DERI Galway at the National University of Ireland, Galway, Ireland), Pablo Badía (Linguaserve), Aaron Beaton (Opera Software), Luis Bellido (Universidad Politécnica de Madrid), Aljoscha Burchardt (German Research Center for Artificial Intelligence (DFKI) Gmbh), Nicoletta Calzolarl (CNR--Consiglio Nazionale delle Ricerche), Giuseppe Deriard (Linguaserve), Pedro Luis Díez Orzas (Linguaserve), David Filip (University of Limerick), Karl Fritsche (Cocomore AG), Daniel Grasmick (Lucy Software and Services GmbH), Declan Groves (Centre for Next Generation Localisation), Moritz Hellwig (Cocomore AG), Tao Hong (Baidu, Inc.), Dominic Jones (Trinity College Dublin), Milan Karásek (Moravia Worldwide), Jirka Kosek (University of Economics, Prague), Maxime Lefrançois (Institut National de Recherche en Informatique et en Automatique (INRIA)), David Lewis (Trinity College Dublin), Fredrik Liden (ENLASO Corporation), Arle Lommel (German Research Center for Artificial Intelligence (DFKI) Gmbh), Jan Nelson (Microsoft Corporation), Des Oates (Adobe Systems Inc.), Carina Pellar (Cocomore AG), Georg Rehm (German Research Center for Artificial Intelligence (DFKI) Gmbh), Phil Ritchie (VistaTEC), Thomas Rüdesheim (Lucy Software and Services GmbH), Felix Sasaki (W3C/ERCIM), Yves Savourel (ENLASO Corporation), Najib Tounsi (Ecole Mohammadia d'Ingenieurs Rabat (EMI)), Ronny Unger (Cocomore AG), Piek Vossen (Vrije Universiteit), Tadej Štajner (Jozef Stefan Institute).