D2.3: METADATA TEST SUITE

Leroy Finn, David Lewis and Dominic Jones

Distribution: Consortium Internal Report

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

FP7-ICT-2011-7

Project no: 287815

Document Information

Deliverable number:	2.3
Deliverable title:	Metadata Test Suite
Dissemination level:	PU
Contractual date of delivery:	15th September 2013
Actual date of delivery:	15th September 2013
Author(s):	Leroy Finn, David Lewis and Dominic Jones
Participants:	TCD, Cocomore, DCU, Enlaso,]init[, JSI, Linguaserve, Lorgus, Lucy Software, Moravia, Shaun McCance, Tilde, UL, Sebastian Hellmann VistaTEC and Felix Sasaki
Internal Reviewer:	TCD
Workpackage:	WP3
Task Responsible:	WP3
Workpackage Leader:	Felix Sasaki

Revision History

Revision	Date	Author	Organization	Description
1	02/09/2013	Leroy Finn, David Lewis and Dominic Jones	TCD	Initial draft of how to use the Test Suite
2	5/09/2013	Leroy Finn	TCD	Final Draft

CONTENTS

1. INTRODUCTION

The ITS 2.0 test suite is located at the following address https://github.com/w3c/its-2.0-testsuite. There are four main parts to the ITS 2.0 test suite which are:

1. Conformance testing for HTML & XML
2. Conformance testing for NIF
3. Input file validation
4. XLIFF samples

2. CONFORMANCE TESTING FOR HTML AND XML

2.1. What is Conformance testing?

Conformance testing is a type of testing where a system has to meet some specified standard. In the case of ITS 2.0 this standard is dictated by the W3C standards document http://www.w3.org/TR/its20/. To test for compliance a series of tests have been developed externally by TCD. The specification states that there are 4 different types of conformance which have to be tested for and more details on this can be found at the following address http://www.w3.org/TR/its20/#conformance.

2.2. Details of the conformance tests

The test suite is used to test user implementations conformance to the ITS 2.0 standard. The test suite has a set of test documents for both XML and HTML which are then used to validate the different ITS 2.0 constructs available for each data category. The ITS 2.0 test suite has 225 input test files which have been created for the 19 data categories. There are 136 XML input files and 89 HTML input files. All of these files have been validated successfully against the schemas for ITS 2.0. Section 4 of this document provides more information on how to validate XML files and validate HTML files. For the conformance testing, each test file in the test suite requires at least two implementations in order to be allowed into the ITS 2.0 standard. This is in line with the conformance clauses which can be found at the following address http://www.w3.org/TR/qaframe-spec/. The requirement of two implementations per test suite file helps in catching errors in the standard along with making sure implementers are using the standard correctly.

2.3. Outcome of conformance tests

The detailed breakdown of the conformance testing results and of systems which are certified in complying with the ITS 2.0 standard can be found in the test suite implementation report. This report is located at the following address http://www.w3.org/International/multilingualweb/lt/drafts/its20/its20-implementation-report.html.

2.4. How to use the test suite?

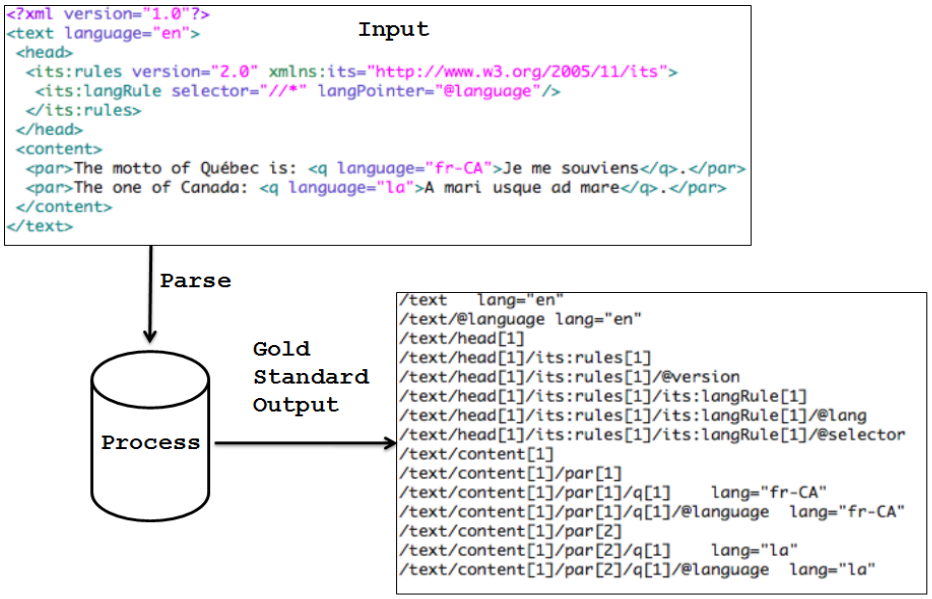

Figure 1: Test Suite processing Input to Output

The validation of the input files is done through processing the input file so that it outputs a file to match a corresponding gold standard output file. The gold standard was developed to be easy to understand and implement for conformance testers. The gold standard output format was developed by the ITS 2.0 working group. Figure 1 above describes the test suite files process. The ITS 2.0 test suite input files are located at the following address https://github.com/w3c/web-platform-tests/tree/master/conformance-checkers/html-its.

2.5. Gold Standard output specifications details:

The gold standard output files have the following characteristics:

- Every element and attribute path from the XML and HTML file are listed (apart from the content within script elements in HTML)
- The output has to be tab-delimited format:

  /html/body[1]/p[1]/span[2] annotatorsRef="text-analysis|http://enrycher.ijs.si" taConfidence="0.5" taIdent="301467919" taSource="Wordnet3.0"

- The attributes within elements have to be in alphabetical order:

  /doc/header[1]/its:rules[1]/its:locQualityIssueRule[2]

  /doc/header[1]/its:rules[1]/its:locQualityIssueRule[2]/@locQualityIssueComment

  /doc/header[1]/its:rules[1]/its:locQualityIssueRule[2]/@locQualityIssueProfileRef

  /doc/header[1]/its:rules[1]/its:locQualityIssueRule[2]/@locQualityIssueSeverity

/doc/header[1]/its:rules[1]/its:locQualityIssueRule[2]/@locQualityIssueType

/doc/header[1]/its:rules[1]/its:locQualityIssueRule[2]/@locQualityIssuesRef

/doc/header[1]/its:rules[1]/its:locQualityIssueRule[2]/@selector

- The rules output also have to be in alphabetical order:

  //html/body[1]/p[1]/span[2] annotatorsRef="text-analysis|http://enrycher.ijs.si" taConfidence="0.5" taIdent="301467919" taSource="Wordnet3.0"

- The rules output does not contain Pointer style attribute values but it has to have its equivalent as results from ITS processing (unless it is in the target pointer data category then it must display targetpointer="...." and the pointer details):

  Incorrect:

  /doc/para[1]/issue[2] locQualityIssueTypePointer="misspelling" locQualityIssuesRefPointer="#l1234"

  Correct:

  /doc/para[1]/issue[2] locQualityIssueType="misspelling" locQualityIssuesRef="#l1235"

- The rules output for local html rules have to be like their global counterparts:

  Incorrect:

  /doc/p[1] its-loc-quality-issue-type="misspelling" locQualityIssuesRef="#l1235"

  Correct:

  /doc/p[1] locQualityIssueType="misspelling" locQualityIssuesRef="#l1235"

2.6. How gold standard output is compared to implementers output?

The conformance test output of a proposed ITS2.0 implementation is located in the folder **its-2.0-testsuite/its2.0/outputimplementors**. This can then be tested against the gold standard output files located in the **its-2.0-testsuite/its2.0/expected** folder. This can be done simply through performing a diff of the implementation's output files and the corresponding gold standard output files. To automate this process for implementers a test suite dashboard was created which supports the following tasks:

- Help to track the process of implementers in relation to the tests in which they were committed to complete ***(indicated on the dashboard via 'N/A' which means the implementer did not commit to run the test)***.
- Help to track if the implementer's output file matches the corresponding gold standard output file ***(indicated on the dashboard via fnf: the output file from the implementer has not been found OK = the output file is identical to the reference output file)***.

- Help to track if the output file that is in the output folder for a particular tests doesn't match the corresponding gold standard output file ***(indicated on the dashboard via 'error = an error occurred', e.g. the output file is not available or it is not identical to the reference output file. Move the mouse over error message to see details)***.
- Help to track whether the implementer has committed an output file or not for a corresponding test ***(indicated on the dashboard via fnf: the output file from the implementer has not been found)***.
- The dashboard can also track how many tests a particular implementer has left to run.

2.7. Validating Output Test Files

To validate the implementer's output files the test suite dashboard has to be compiled so that a diff across all files in the **its-2.0-testsuite/its2.0/outputimplementors** folder can be performed against the corresponding files in the gold standard output **its-2.0-testsuite/its2.0/expected** folder. The test suite dashboard can be compiled by doing the following:

- Download saxon.jar from here: http://saxon.sourceforge.net
- Then use this command (Linux/Mac/Windows): java -jar /path/of/file/saxon.jar testsuiteMaster.xml testsuiteDashboard.xsl -o:testSuiteDashboard.html
- Upload newly compiled testsuiteDashboard.html to the git hub
- Check the state of your files in the related data categories on this web page: http://htmlpreview.github.io/?https://raw.github.com/w3c/its-2.0-testsuite/blob/master/its2.0/testSuiteDashboard.html

The files for the test suite dashboard are as follows:

- **testsuiteMaster.xml -** has a list of the implementers who committed tests and the tests that they are involved in and thereby aids in the creation of the testSuiteDashboard.html
- **testsuiteDashboard.xsl -** does the diffs between the implementer's output and the gold standard output.
- **testSuiteDashboard.xml -** gives information on the errors between the diffs for the implementer's output and the gold standard output
- **testSuiteDashboard.html -** the live html test suite dashboard which is located http://htmlpreview.github.io/?https://raw.github.com/w3c/its-2.0-testsuite/blob/master/its2.0/testSuiteDashboard.html

3. TESTING FOR NIF

This part of the test suite is used to determine whether ITS 2.0 implementations of NIF 2.0 meet the ITS 2.0 specification standard for NIF usage. The mapping between an XML and HTML document annotated with ITS to and from NIF is not a normative part of the ITS 2.0 specification. The NIF test suite has a set of test documents for HTML for the **Localisation Quality Issue** data category which is then used to validate the different aspects of ITS 2.0 and NIF against various constructs available to the Localisation Quality Issue data category. There are a total of 11 files only for HTML in the Localisation Quality Issue data category. The input files are located in the **its-2.0-testsuite/its2.0/nif-conversion/input** folder.

The validation of the input files is done through processing the input file so that it outputs a file to match a gold standard output file. The gold standard for NIF is RDF output using the ITS 2.0 ontology. The gold standard output for NIF can be reached by following the NIF conversion algorithm discussed in the ITS 2.0 specification located http://www.w3.org/TR/its20/#conversion-to-nif .

3.1. How gold standard NIF output is compared to implementers output?

The NIF output files are compared via the use of SPARQL queries done over the implementers RDF/NIF output files (.ttl). If the SPARQL queries are successful then the NIF output files are correct and meet the gold standard. The implementer's NIF test output files are located in the **its-2.0-testsuite/its2.0/nif-conversion/expected** folder.

3.2. Validating NIF output files

Prerequisites: Java and UNIX Shell

- create a temporary folder for output files (henceforth referred to as $datafolder)
- read ITS files from "its2.0/nif-conversion/input/" one by one, convert to NIF and write output files in turtle to $datafolder
- go to directory cd its2.0/nif-conversion/sparqltest
- run : ./executeAllTests.sh ../relative/pathTo/$datafolder

Explanations of output:

- If no message appears between "Running: test1.sparql" and "Done: test1.sparql" the test was successful.
- Otherwise the output filename and additional debug output is shown.

4. INPUT FILE VALIDATION

This part of the test suite is important to ensure that any test input files, including any used by implementers in addition to the test suite, represent valid use of ITS 2.0 annotation in HTML and XML. More information about this validator can be found http://validator.nu/ and http://about.validator.nu/.

4.1. Validating Input Test Files

The following sections detail how to validate the test suite input files both HTML and XML.

4.1.1 Validating XML test files

- Download and install Ant from http://ant.apache.org/
- Run 'ant validate-xml' command in **its2.0** directory

4.1.2 Validating HTML test files

- Download and install Ant from http://ant.apache.org/
- Download html5-its-tools from https://github.com/kosek/html5-its-tools
- Modify **its2.0/build.properties** to point to your local copy of html5-its-tools
- Run 'ant validate-html' command in **its2.0** directory

4.1.3 Validating all test files

- Make sure that XML and HTML validation described above works for you
- Run 'ant' command in **its2.0** directory
- Please note that HTML schema doesn't supports RDFa so RDFa attributes are reported as errors
- Please note that currently Schematron validation is not performed so some errors are not detected.

5. XLIFF SAMPLES

The test suite also contains some sample XLIFF files. These are not used in conformance testing, but demonstrate the representation of ITS 2.0 metadata in XLIFF. These XLIFF files are located in **its-2.0-testsuite/its2.0/xliffsamples** folder.