ITS 2.0 Validation Techniques

Jirka Kosek
University of Economics, Prague
Prague, Czech Republic
jirka@kosek.cz

Abstract

ITS 2.0 (Internationalization Tag Set) is a new W3C Recommendation, which defines a set of universal elements and attributes that can be used in host vocabularies like HTML or XML to improve localization and translation processing. The fact that ITS markup can be combined with almost any other markup makes validation of ITS content more challenging than usual. This paper discusses various approaches to validation of ITS markup both in XML and HTML documents. Advantages and disadvantages of various approaches are discussed. Special attention is given also to validation of HTML5 content.

Keywords: *XML, HTML, XML schema, validation, NVDL, ITS*

1. Introduction

ITS 2.0 (Internationalization Tag Set) is a new W3C Recommendation which defines set of universal elements and attributes that can be used in a host vocabularies like HTML or XML to improve localization and translation processing (Filip, D., McCance, S., Lewis, D., Lieske, C., Lommel, A., Kosek, J., Sasaki, F., Savourel, Y.; Eds. 2013). The most common way to use ITS is to attach special attributes from the ITS namespace to elements containing content that can benefit from additional language related metadata.

```
<para>It would certainly be quite a <phrase its:translate="no">faux
pas</phrase> to start a dissertation in a pub…</para>
```

Example 1: Local ITS markup in an XML document expressed as an attribute

In example 1, you can see `its:translate` attribute in action. This attribute indicates that content of the `<phrase>` element should not be translated.

There are dozens of other attributes similar to `its:translate` available in ITS. Using this so called "local markup" is arguably the most common way of using ITS.

Another option is to define global rules. This is done by using dedicated rules elements. Example 2 shows a rule that forbids translation of labels in user interface in a DocBook document. Please note that rules are usually placed in some metadata wrapper element, such as `<info>` or `<head>`.

```
<article xmlns="http://docbook.org/ns/docbook"
         xmlns:db="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>
    <its:rules>
      <its:translateRule selector="//db:guilabel" translate="no"/>
    </its:rules>
  </info>
  <para>This is a short article. Title of article is shown in
        <guilabel>Title</guilabel> field.</para>
</article>
```

Example 2: Global ITS Rules

From examples 1 and 2, it is apparent that attributes for local ITS markup need to be allowed on almost

```
<!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>
```

Example 3: Local ITS markup inside HTML document

any element, while special ITS elements are better to be allowed only inside of specific elements that already serve as metadata containers in the host format when ITS markup is integrated.

In HTML, the situation is similar. The only difference is that namespaces cannot be used. So instead of using a namespace prefix followed by a ":" (colon), such as `its:`, HTML has to use the hardcoded prefix `its-` as shown in example 3.
Let us now see various validation options for ITS content.

2. Schema Languages

In the XML world, validation is done using schema languages. A schema describes constraints on a document structure (elements and attributes you can use), datatypes (values allowed inside elements and attributes) and sometimes it can also express more advanced checks.

Over the time, several schema languages emerged. Currently, the two most common schema languages are W3C XML Schema (Fallside, D.C., Walmsley, P.;Eds., 2004) (Thompson, H.S., Beech, D., Maloney, M., Mendelsohn, N.; Eds., 2004) (Biron, P.V., Malhotra, A.; Eds., 2004) and RELAX NG (Clark, J., Murata, M.; Eds., 2001). Both of them are grammar based, which means that they can precisely list all possible element/attribute combinations in a very concise way. However, this approach has some limitations, especially when more complex relationships in documents need to be described. In such cases, the Schematron language (*Document Schema Definition Languages (DSDL) — Part 3: Rule-Based Validation — Schematron.* 2006) is very popular, as it can describe complex constraints over XML documents using XPath expressions.

There are also special schema languages that are useful in particular cases. One of them is NVDL (*Document Schema Definition Languages (DSDL) — Part 4: Namespace-Based Validation Dispatching Language — NVDL.* 2006), which can be very useful if you use several namespaces in your document and there is no single schema for such compound document available.

3. Validating ITS markup alone

In case you do not have any schema for a document and just want to validate ITS markup used inside it, you can use the NVDL schema available as a part of

```
<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>
```

Example 4: NVDL schema for ITS

the ITS specification.

This schema finds all elements and attributes from the ITS namespace in a document and sends them separately for validation against the RELAX NG schema for ITS elements and attributes. Everything else that is not ITS markup is ignored during this validation.

validation does not detect misplaced elements with ITS markup, usually rules. For attributes, this is not such an issue, as ITS attributes are usually available on most elements of a host language.

4. Validating host vocabulary together with ITS markup

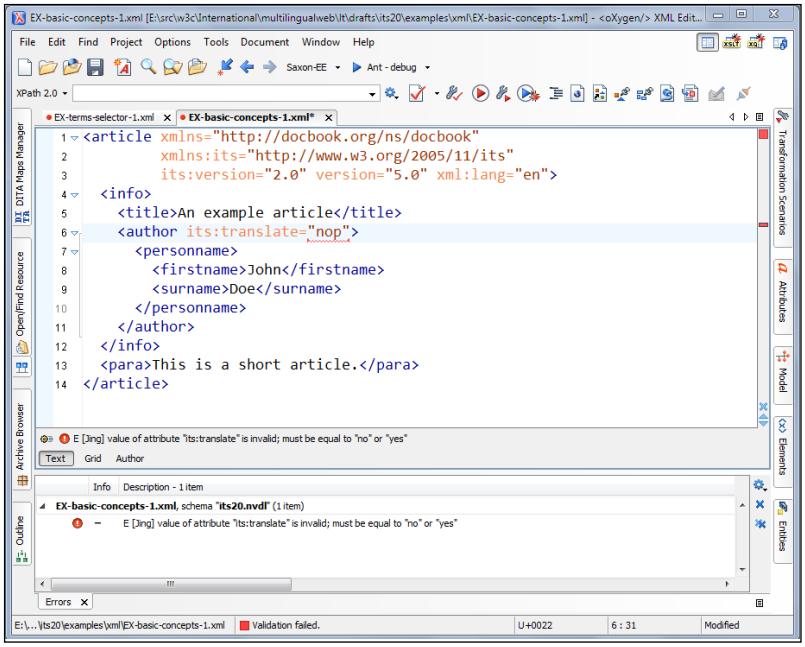

Figure 1: Validation of invalid ITS markup inside oXygen XML editor

The main advantage of this approach is that any file containing ITS markup can be validated without an additional effort. The main disadvantage is that such

If you need to have precise control over where in your existing vocabulary ITS markup can appear, you need to create a new schema that combines the

```
# Include base DocBook schema
include "docbook.rnc"

# Include base ITS schema
include "its20.rnc"
{
    # Disable ITS directionality as DocBook has its own attribute
    its-attribute.dir = empty
}

# Add local ITS attributes to all DocBook elements
db.common.base.attributes &= its-local.attributes & its-
attribute.version?

# Allow its:rules inside info element
db.info.extension |= its-rules
```

Example 5: DocBook + ITS schema

schema of the host vocabulary with the ITS schema. In order to make this easy, the ITS specification contains highly modular schemas in RELAX NG and W3C XML Schema languages. It is rather easy to take ITS building blocks from these schemas and combine them with the host vocabulary.

Example 5 shows how to integrate ITS schema into the schema for DocBook. ITS rules are added into the `<info>` element and ITS attributes are allowed to appear on any element. Because DocBook already contains its own attribute `dir` for specifying directionality, the corresponding attribute `its:dir` is removed from the ITS schema.

Please note that the schema in example 5 had to be simplified for the purposes of this publication. The complete schema can be found at https://github.com/docbook/docbook/tree/master/relaxng/schemas/dbits.

This approach to creating a combined schema of the ITS and a host vocabulary has many advantages and is thus preferable. The resulting schema will catch both, errors in the host markup and in the injected ITS markup, and shall also identify any misplaced ITS markup. For an even more reliable check, the documents to be validated can be additionally checked against the Schematron schema that is also included with the ITS specification.

There is one obvious disadvantage, this approach requires that your document has a schema and this schema needs to be extendable with ITS support. Sometimes, this is easy – for example schema for DocBook has many hooks that make extending it very easy. Unfortunately this is not the case with all of the potential host vocabularies.

Validation of the ITS markup within HTML5 documents is rather easy because support for ITS was added as an option to some of the online validation services, such as http://validator.w3.org and http://validator.nu.

Internally, validation of HTML+ITS is driven by RELAX NG schemas. That basically means that an approach similar to the one described in Section 4, *Validating host vocabulary together with ITS markup* has been used. The underlying schemas are available from https://bitbucket.org/validator/validator/src/.

Elements based ITS rules or standoff markup must be placed inside the `<script>` element because the HTML language lacks extensibility. Unfortunately,

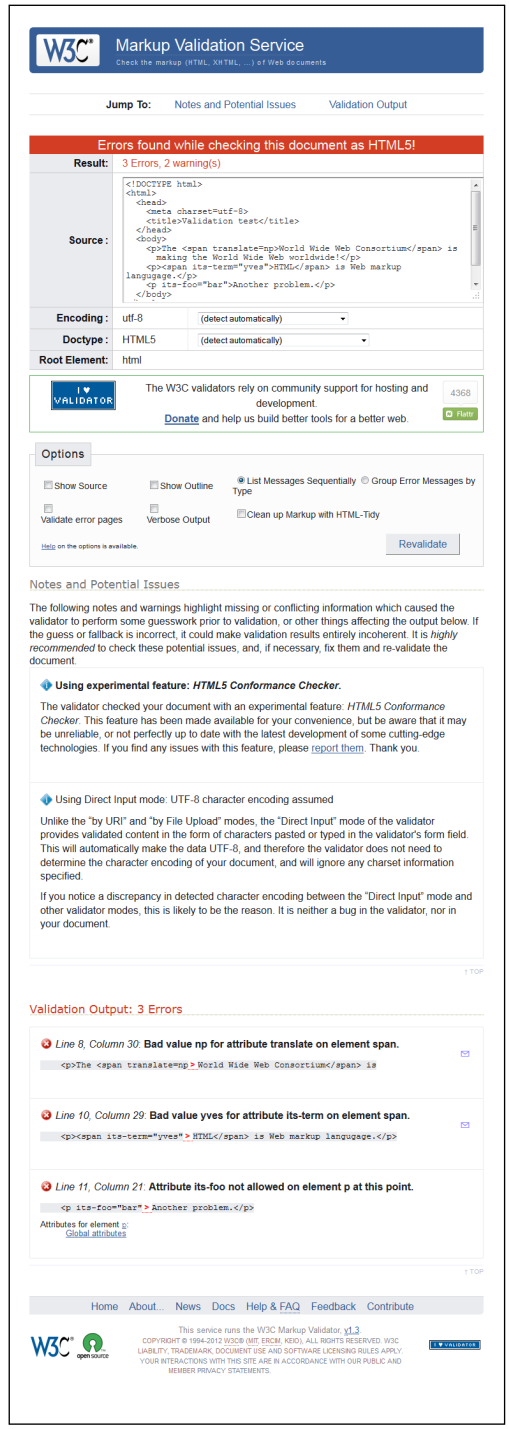

Figure 2. Result of HTML+ITS validation in W3C validator

from the validation point of view the content of this element is just an opaque string and cannot be reasonably validated. Because of this issue, it is recommended not to use any ITS elements inside an HTML page, and restrict the use to just ITS attributes. The ITS elements holding rules can then be stored in separate XML files and linked from the HTML page using the `<link>` element.

6. Tools

There are many implementations of validators. From the user perspective, the easiest option is to use a validator integrated in a popular XML editor such as the oXygen XML editor. If a commercial tool cannot be acquired, an open-source tool, such as the Jing tool (available from http://code.google.com/p/jing-trang/) is an option.

7. Conclusions

We have shown and discussed several approaches to validation of documents containing ITS markup. Potential ITS implementers should definitively include validation as one of the initial steps in procuring their ITS tool chain. This is critical to make sure that any manually or automatically produced ITS markup is conformant and thus can be successfully processed by a variety of ITS ready tools.

References

Biron, P.V., Malhotra, A. (Eds.) (2004) *XML Schema Part 2: Datatypes Second Edition* [online], Recommendation. ed, Recommendation, W3C, available: http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/ [accessed 10 Dec 2013].

Clark, J., Murata, M. (Eds.) (2001) *RELAX NG Specification* [online], Committee Specification. ed, Standard, OASIS, available: https://www.oasis-open.org/committees/relax-ng/spec-20011203.html [accessed 10 Dec 2013].

Document Schema Definition Languages (DSDL) — Part 3: Rule-Based Validation — Schematron. [online] (2006) International Standard, ISO/IEC, available: http://standards.iso.org/ittf/PubliclyAvailableStandards/c040833_ISO_IEC_19757-3_2006%28E%29.zip [accessed 10 Dec 2013].

Document Schema Definition Languages (DSDL) — Part 4: Namespace-Based Validation Dispatching Language — NVDL. [online] (2006) International Standard, ISO/IEC, available: http://standards.iso.org/ittf/PubliclyAvailableStandards/c038615_ISO_IEC_19757-4_2006%28E%29.zip [accessed 10 Dec 2013].

Fallside, D.C., Walmsley, P. (Eds.) (2004) *XML Schema Part 0: Primer Second Edition* [online], Recommendation. ed, Recommendation, W3C, available: http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/ [accessed 10 Dec 2013].

Filip, D., McCance, S., Lewis, D., Lieske, C., Lommel, A., Kosek, J., Sasaki, F., Savourel, Y. (Eds.) (2013) *Internationalization Tag Set (ITS) Version 2.0* [online], Recommendation. ed, Recommendation, W3C, available: http://www.w3.org/TR/its20/ [accessed 11 Nov 2013].

Thompson, H.S., Beech, D., Maloney, M., Mendelsohn, N. (Eds.) (2004) *XML Schema Part 1: Structures Second Edition* [online], Recommendation. ed, Recommendation, W3C, available: http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/ [accessed 10 Dec 2013].