


<oo> → <dh> Digital humanities

Maintained by: David J. Birnbaum (djbitt@gmail.com) 

Last modified: 2012-10-27T14:34:01+0000

Schematron assignment #2

Preamble

This assignment is situated in the context of Real Life linguistic documentation project in which we were asked to provide some XML assistance. Your assignment involves a bit of Schematron in the middle, but we describe below both the linguistic project itself and the eventual XML conversion that the Schematron was ultimately used to facilitate.

The problem

Here's a quote from <http://dh.obdurodon.org/schematron-class-01.html> (simplified slightly):

Linguistic corpora often record transcriptions in multiple tiers, such as a transcription of the original utterance, a word-by-word gloss with grammatical information, and a more fluid, natural-language translation. The set of notational conventions most commonly used for this purpose by corpus linguists have been codified in the Leipzig Glossing Rules (<http://www.eva.mpg.de/lingua/resources/glossing-rules.php>). Here is a Russian example based on that document:

Orth	Мы	с	Марко	поеха-л-и	автобус-ом	в	Переделкино
Translit	My	s	Marko	poexa-l-i	avtobus-om	v	Peredelkino.
ILG	we	with	Marko	go-PST-PL	bus-by	to	Peredelkino.
Free	'Marko and I went to Peredelkino by bus.'						

Other tiers might include International Phonetic Alphabet (IPA) and interlinear glossing or free translation into other languages.

Each of the computationally tractable tiers should have the same number of words, and each word should have the same number of hyphens.

From field notes to markup

Field linguists often type up this information in plain text, so that their starting point is something like:

Orth: Мы с Марко поеха-л-и автобус-ом в Переделкино
Translit: My s Marko поеха-l-i avtobus-om v Peredelkino.
ILG: we with Marko go-PST-P bus-by to Peredelkino.
Free: Marko and I went to Perdelkino by bus.

Assume that you can get the raw text into the following XML easily:

```
<sentence>
  <orth>Мы с Марко поеха-л-и автобус-ом в Переделкино</orth>
  <translit>My s Marko поеха-l-i avtobus-om v Peredelkino.</translit>
  <ilg>we with Marko go-PST-P bus-by to Peredelkino.</ilg>
  <free>Marko and I went to Perdelkino by bus.</free>
</sentence>
```

You don't have to do the following for this assignment, but now that you've learned a bit about regular expressions, **<xsl:analyze-string>**, and the XPath **tokenize()** function, you would be able to write XSLT to convert this XML to a different XML structure, one where the pieces are aligned properly, that is, so that every word and morpheme on the Orth, Translit, and ILG (interlinear gloss) tier is associated with the corresponding word or morpheme on the other tiers (except the Free tier, which isn't expected to match up; it's a free translation, after all). But that works only if the person who entered the data originally got the spaces and hyphens right! If the number of spaces and hyphens doesn't match up in the Orth, Translit, Gram, and ILG tiers, you can't automate the alignment.

The task: using Schematron to get your data ready for XML-to-XML conversion

When we had to perform this type of plain-text-to-XML conversion for a real linguistic documentation project, the linguists' initial, raw field notes had lots of error: spaces instead of hyphens and vice versa, as well as other punctuation (periods, hash marks, etc.) in place of both spaces and hyphens. This is typical field data; it's very hard for a human to pay attention to counting spaces and punctuation marks, which is why we use markup languages in projects of this sort in the first place. Before we even tried to transform the data with XSLT to something that formalized the word-by-word and morpheme-by-morpheme alignment, we used Schematron to verify that the number of spaces and hyphens matched where it needed to. That doesn't mean that we can't still have a mistake, of course, but it greatly reduces the opportunity that we won't notice an error, since only if we were to make the same error (or the same type of error) in every associated tier would we fool the counter.

Your assignment, then, is to *write a Schematron schema* that will take input like:

```
<sentence>
  <orth>Мы с Марко поеха-л-и автобус-ом в Переделкино</orth>
  <translit>My s Marko поеха-l-i avtobus-om v Peredelkino.</translit>
  <ilg>we with Marko go-PST-P bus-by to Peredelkino.</ilg>
  <free>Marko and I went to Perdelkino by bus.</free>
</sentence>
```

and verify that the first three lines (Orth, Translit, and ILG) all have the same number of spaces and the same number of hyphens. *You do not have to convert this XML to word-aligned or morpheme-aligned XML;*

all you have to do is write the Schematron that will verify whether the spaces and hyphens match. The verification is a prerequisite for the transformation, which would be the next step in Real Life, but for a Schematron assignment all you have to do is ... well ... write the Schematron.

To test your Schematron rules, create your own small sample XML document, with a handful of sentences formatted like the example above, with each tier in its own element but no internal markup separating words or morphemes. You can make up your own examples in a language of your choice or copy examples from <http://www.eva.mpg.de/lingua/resources/glossing-rules.php>. If you make up your own examples, don't worry about the precision of your linguistic annotations; this is an exercise in Schematron, and not in field linguistics. It doesn't matter what tiers you use, as long as you have at least two that have spaces and hyphens in them that are supposed to correspond. You should also make copies of some of your examples, muck up the spaces and hyphenation, and use that bad data to test whether your Schematron schema can catch the errors.

If that's too easy

The following isn't required, but if you feel like exercising your XSLT and XPath skills, you're welcome to transform the input XML, which you've verified with Schematron, to a different XML structure, one that formalizes the word-by-word or morpheme-by-morpheme associations. There's no single right output XML structure (schema) for this purpose, so should you choose to try it, you should first decide what the XML output of the transformation should look like, and then write the XSLT to produce it.