How to avoid suffering from markup: A project report about the virtue of hiding XML

Felix Sasaki
University of Appl. Sciences Potsdam / W3C German-Austrian Office
felix.sasaki@fh-potsdam.de
Let's start with acknowledgements: Thank You, Japanese Layout Taskforce!
Some participants of the Japanese Layout Taskforce (JLTF)
What are (were) these guys doing?

- Describing requirements for Japanese layout - essentially
  - What looks good for a Japanese reader
  - What looks bad
  - And: why?
- Providing input for Web technologies
Bad example:
unadjusted spaces
Good example:

✓ space adjustment
Experts in Japanese Layout

Experts in Web technology (CSS, SVG, XSL-FO)
Communication mostly in Japanese

JLTF – a cross-technology / cross-cultural meeting point

Communication mostly in English
What are they producing:
an aligned Japanese-English document
describing requirements for Japanese
Layout

http://www.w3.org/TR/jlreq/
http://www.w3.org/TR/jlreq/ja/
The following screenshots are from editor's copies
of the two documents
ideographic 安以字衣於

hiragana 阿伊字衣於

katakana あいうえお

[Fig. 1]: Kanji, hiragana and katakana.

(note 1) In addition to ideographic (cl-19), hiragana (cl-15) and katakana (cl-16) characters, various punctuation marks (see [Fig 2]) as well as Western characters (cl-27), such as European numerals, Latin letters and/or Greek letters, may be used in Japanese text. In this document these characters are classified into character classes, for which explanations are given describing their behavior in type-setting.

opening brackets ‘ “ ( [ { 〈 《 「 『

closing brackets ’ ” ) ] } 〉 》 ’ ]

hyphens 〜

平仮名 あいうえお

片仮名 アイウエオ

[Fig 1]: 漢字・平仮名・片仮名

注1）日本語経験には、漢字等（cl-19）、平仮名（cl-15）及び片仮名（cl-16）以外に、多くの約物類を使用する（[図2]参照）。そのほかに、アラビア数字、ラテン文字、ギリシャ文字などの欧文用文字（cl-27）を使用する場合がある。このドキュメントでは、日本語経験で使用する文字について経験上の振り分けから文字クラスとして分類し、解説する。
How do they produce it: the document processing-chain – hiding XML to the authors, but taking benefit of XML during processing
The document processing-chain

0 Editing small XHTML-files containing the content in two languages
1 Validating the multilingual alignment
2 Merging into one large XHTML-file
3 Doing some post-processing
4 Output of two XHTML-files: one in Japanese, one in English
● Steps XSLT processing chain: 2, 3, 4
● Steps RELAX NG validation: 1, 2
● Editing: before 1 (and checked by 1)
Going through an example: 0. editing

<div class="Taiyaku">

<p lang="ja" xml:lang="ja">これはインラインのタグも使用できる例文です。</p>

<p lang="en" xml:lang="en">This is an example sentence which may also contain inline markup.</p>

</div>
Going through an example: 0. editing

```html
<div class="Taiyaku">
  <p lang="ja" xml:lang="ja">これはインラインのタグも使用できる例文です。</p>
  <p lang="en" xml:lang="en">This is an example sentence which may also contain inline markup.</p>
</div>
```

Aligned paragraphs in English and Japanese
Wrapper for aligned units
Going through an example: 1. validation

Definition of various „Taiyaku“ patterns (paragraphs, figures, headings, list items, ...) in RELAX NG:

```
p-taiyaku =
element div {
    attribute class { "Taiyaku" },
    attribute id { xsd:NCName }?,
    p-ja,
    p-en
}
```

```
p-ja =
element p { langAttrJa, commonAtts, p-mix }
p-en =
element p { langAttrEn, commonAtts, p-mix }
```
Explicit naming (not used) versus "Taiyaku" pattern for a paragraph

```html
<p-taiyaku>
  <p-ja>これはインラインのタグも使用できる例文です。 </p-ja>
  <p-en>This is an example sentence which may also contain inline markup. </p-en>
</p-taiyaku>
```

identical „meaning“ in terms of role(s) for processing after editing

```html
<div class="Taiyaku">
  <p lang="ja" xml:lang="ja">これはインラインのタグも使用できる例文です。 </p>
  <p lang="en" xml:lang="en">This is an example sentence which may also contain inline markup. </p>
</div>
```
Going through an example: 2. merging

1 and 2 and 3 and ... become

<html ...><div class="Taiyaku">
  <p lang="ja" xml:lang="ja">これはインラインのタグも使用できる例文です。</p>
  <p lang="en" xml:lang="en">This is an example sentence which may also contain inline markup.</p>
</div>...<html>

<html ...><div class="Taiyaku">
  <p lang="ja" xml:lang="ja">これはまた違う文章です。</p>
  <p lang="en" xml:lang="en">This is a different text.</p>
</div>...</html>
Example: consulting the Unicode character database for standardized character names

**In JIS X 4051, [、] and ...**

**In JIS X 4051, IDEOGRAPHIC COMMA "、"**...
Going through an example: 3. postproc.

This processing also relies on the validation made previously, using the following declaration for the <span> element:

```xml
<element name="span">
    <attribute name="class" value="character"/>
    <xsd:string pattern="\[.\]\.*"/>
</element>
```
Step 4: output of two monolingual documents (seen before)
So much about the "What" and "How" – now about the "Why?"
Some history I: How the project evolved

1) Starting in one language
2) Translating sentence by sentence
3) Translating aligned
4) Giving up translation, working in both languages
5) "We need more features!"
Some history II: How the project evolved & **technical decisions**

1) Starting in one language: *XHTML template with usage instructions*

2) Translating sentence by sentence: *Copy of the template*

3) Translating aligned: *Re-engineering the template, creating validation and transformation chain*

4) Giving up translation, working in both languages: *no technical changes*

5) "We need more features!": *minor transformation and validation tweeks*
Some history III: Background of participants

- Letting people use their own editing environment was crucial.
- (X)HTML as an editing format was the "lowest common denominator" for everybody.
- Hiding the XML machinery (all steps 1-4) from most of them was essential. One should not disturb their main task: explaining Japanese Layout requirements.
Why not explicit naming or XLIFF?

- XLIFF is a vocabulary to align source and translated text
- The social circumstances of the project prevented using XLIFF
- The same reason prevented the use of any other special-purpose XML-vocabulary
Why not sentence-alignment?

The goal of the task force is not to propose actual solutions but describe important issues as basic information for actual implementations.

(We) have decided not to propose actual solutions, but to make an explanation of important issues. This is because (we) think that it is important in the first place to explain precisely precondition constraints for considering implementation level problems.

Why not sentence-alignment?

Japanese version

具体的な解決策を提示することではなく，要望事項の説明することにした．それは，実装レベルの問題を考える前提条件をまず明確にすることが重要であると考えたからある．
Why RELAX NG?

- An arbitrary choice
- Other means (e.g. XSD 1.1. conditional type assignment or Schematron) could do the job as well
Conclusion – was it worth it?

- Of course! See the great results at http://www.w3.org/TR/jlreq/

- Outlook: adding constraints to XHTML via "hidden" XML-validation and processing can help with
  - Getting more adoption for XML in communities like microformats
  - Spreading the word without speaking it out loudly
Conclusion – was it worth it?

- Of course! See the great results at http://www.w3.org/TR/jlreq/

- Outlook: adding constraints to XHTML via "hidden" XML-validation and processing can help with
  - Getting more adoption for XML in communities like microformats
  - Spreading the word without speaking it out loudly

Thank you for your attention!