

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)

Liam Quinn

Doug Schepers

Yasuhiro Anan
Masayuki Nakano

Yasuyuki Hirakawa

Tony Graham

Anders Berglund

Hiroyuki Chiba

Klaas Bals

Edward Jiang

Junzaburo Edamoto

Ishino-san

Richard Ishida

Toshi Kobayashi

Elika Eternad

Kunio Ohno

Tatsuo Kobayashi

Shinyu Murakami



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



㊟Closing brackets following full stops

…『』。』』。…

㊟Opening brackets following opening brackets

…『』。「」。…

㊟Full stops following closing brackets

…『』。 『』。…

㊟Closing brackets following closing brackets

…『』) 』』。…

㊟Opening brackets following commas

…『』、 『』。…

㊟Continuation of brackets and middle dots

… 『編集』・ 『校正』

㊟Opening brackets following closing brackets

…『』』 『』。…

Experts in
Japanese Layout



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

Communication
mostly in Japanese

Felix Sasaki

Masayuki Nakano

Yasuhiro Anan

Doug Schepers

Hiroyuki Chiba

Klaas Bals

Edward Jiang

Yasuyuki Hirakawa

Tony Graham

Anders Berglund

JLTF – a

Ishino-san

cross-technology / cross-cultural meeting point

Junzaburo Edamoto

Richard Ishida

Kunio Ohno

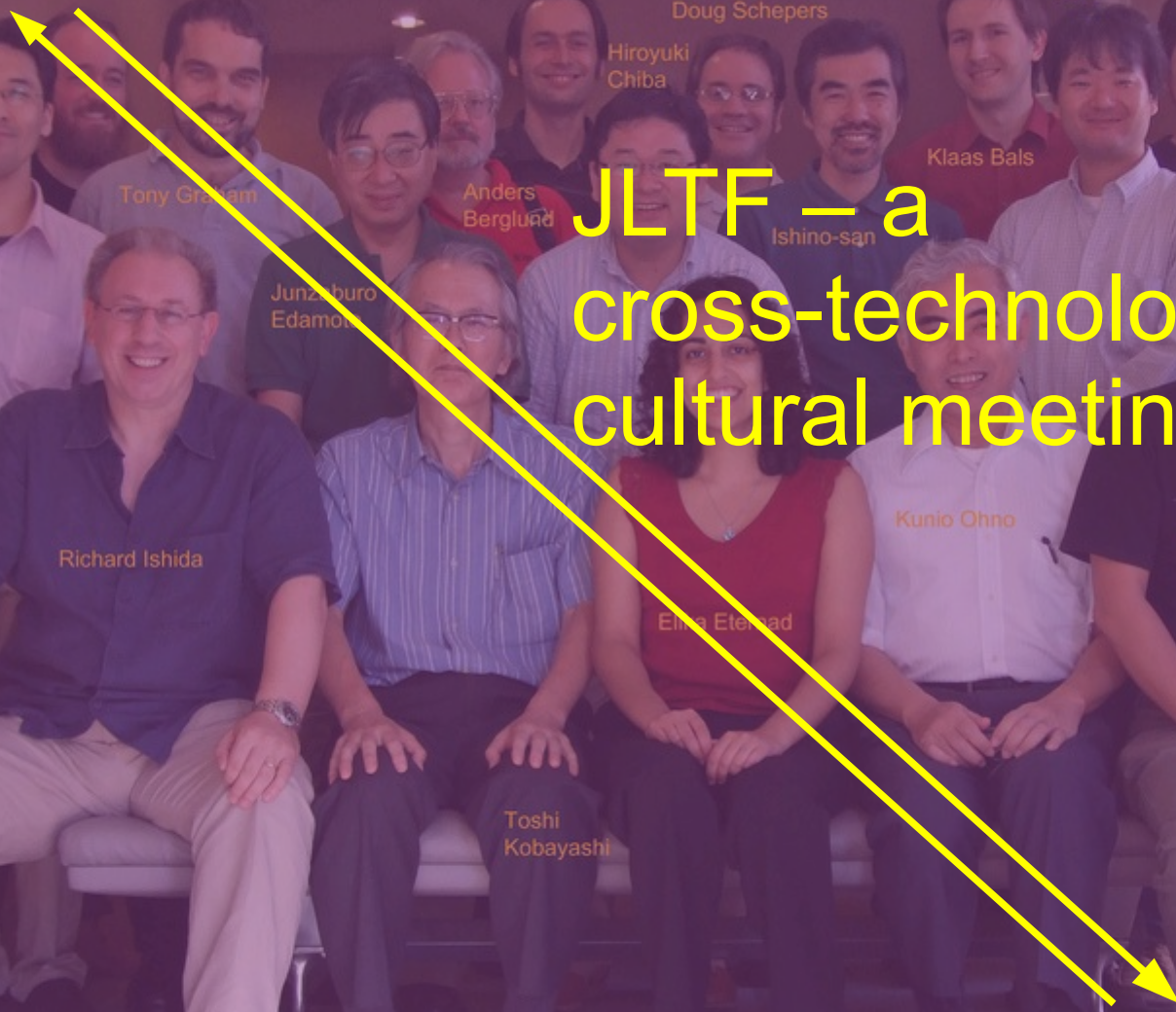
Tatsuo Kobayashi

Elena Eterlad

Shinyu Murakami

Toshi Kobayashi

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 阿伊宇江於
hiragana	あいうえお
katakana	アイウエオ

[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	— ~ —

完了

平仮名	あいうえお
片仮名	アイウエオ

[図1]: 漢字・平仮名・片仮名

注1) 日本語組版には、[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 \(cl-27\)](#) を混用する場面がある。このドキュメントでは、日本語組版で使用する文字について組版上の振る舞いから文字クラスとして分類し、解説する。

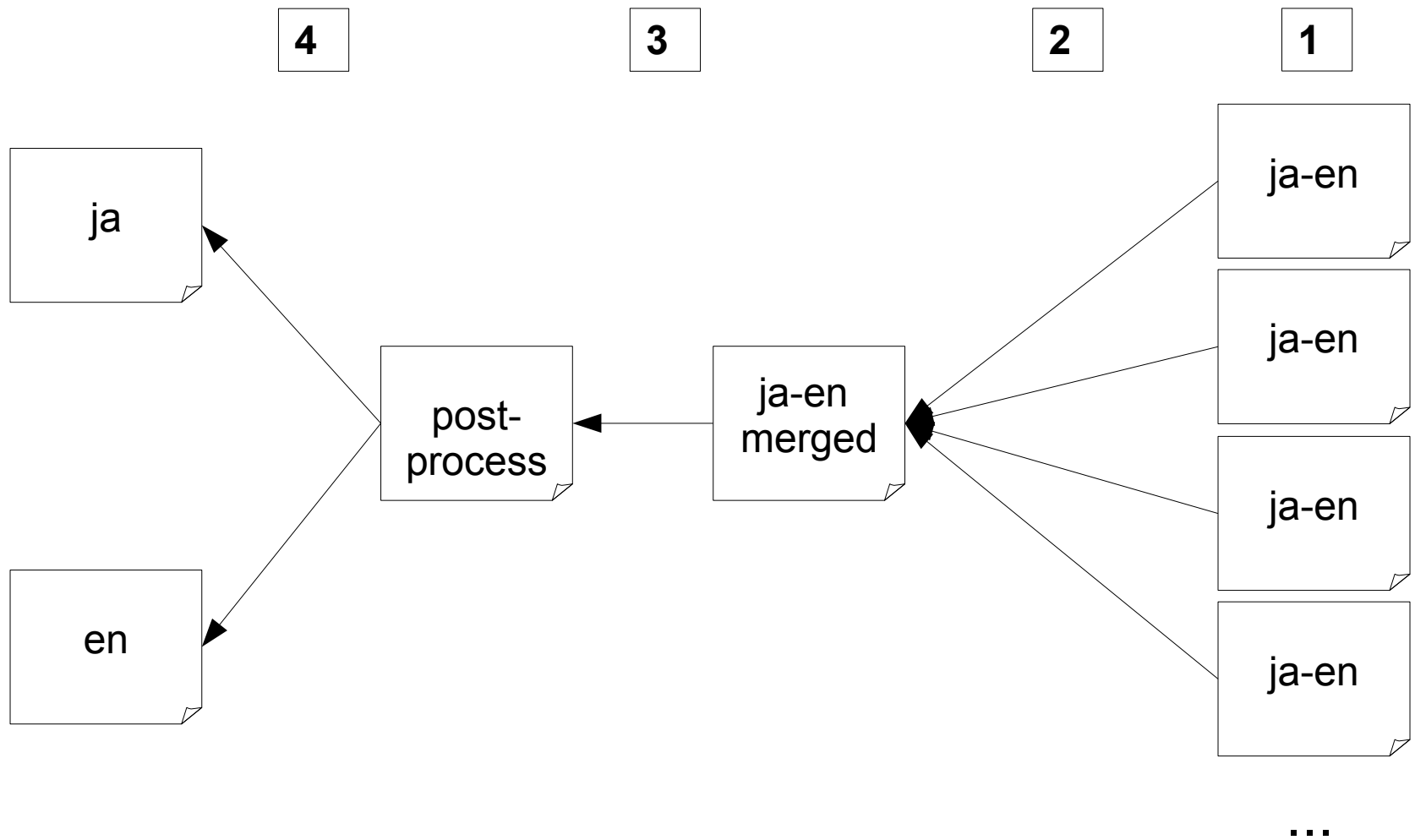
始め括弧類	‘ “ ([[{ < 《 「 『 【
終わり括弧類	’ ”)]] } > 》 」 』 】
ハイフン類	— ~ —

完了

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

```
<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

```
<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

```
<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

```
<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>
```

2

```
<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>
```

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>...  
<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>
```

Going through an example: 3. postproc.

Example: consulting the Unicode character data base for standardized character names

In JIS X 4051, [、]
and ...

database lookup



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 `` element

```
element span {  
  attribute class { "character" },  
  xsd:string { pattern = "\[.*\].*" } }
```

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

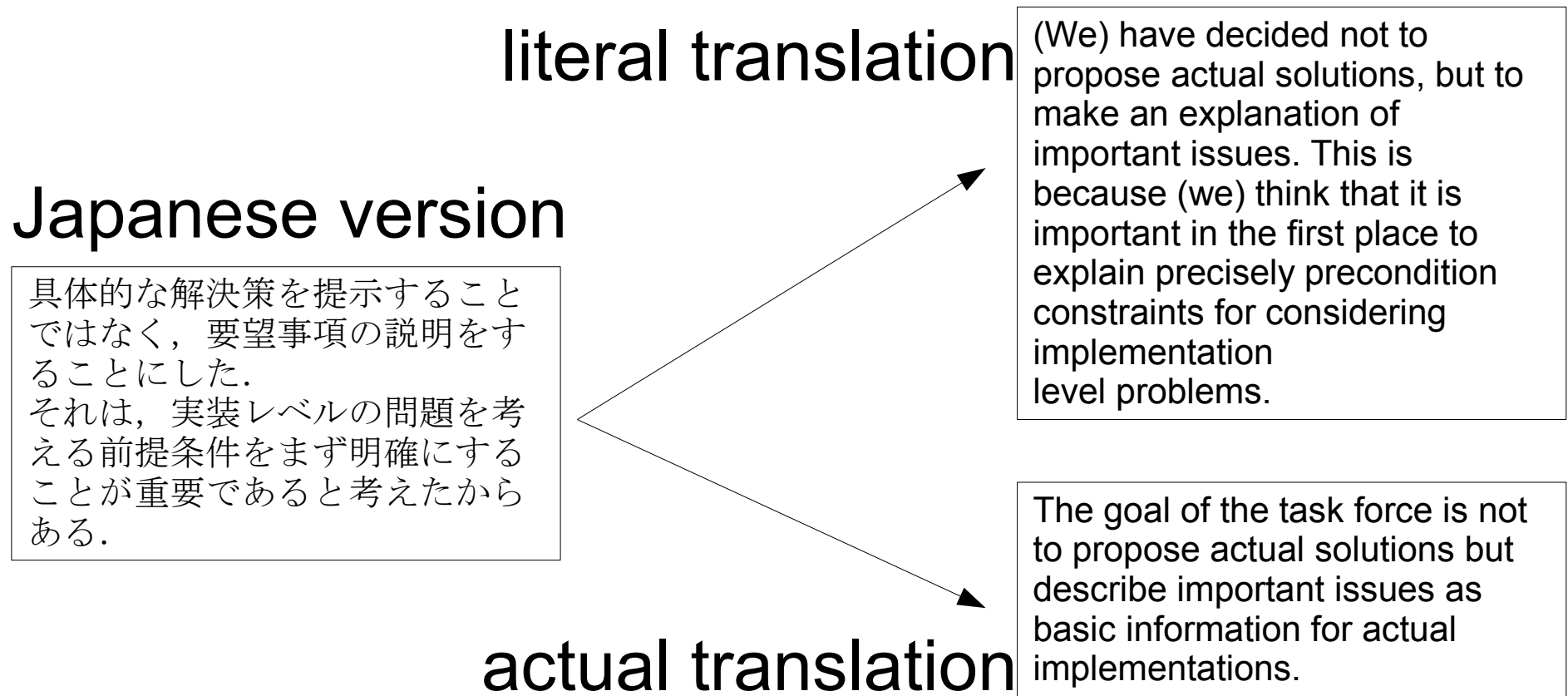
Some history III: Background of participants

- 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?



Translation sentence-by sentence does not work here

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

ありがとうございました。

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!