Work within the W3C Internationalization Activity and its Benefit for the Creation and Manipulation of Language Resources

Felix Sasaki*

*World Wide Web Consortium
Keio Research Institute at SFC
5322 Endo Fujisawa, Kanagawa, 252-8520 Japan
{fsasaki}@w3.org

Abstract

This paper introduces ongoing and current work within Internationalization (i18n) Activity, in the World Wide Web Consortium (W3C). The focus is on aspects of the W3C i18n Activity which are of benefit for the creation and manipulation of multilingual language resources. In particular, the paper deals with ongoing work concerning encoding, visualization and processing of characters; current work on language and locale identification; and current work on internationalization of markup. The main usage scenarios is the design of multilingual corpora. This includes issues of corpus creation and manipulation.

1. Background: Internationalization and W3C

1.1. What is Internationalization?

According to (Ishida and Miller, 2006), *internationalization* is the process of making a product or its underlying technology ready for applications in various languages, cultures and regions. The acronym of Internationalization is used as “i18n” because there are 18 characters between the first character “i” and the last character “n”.

Closely related to internationalization is *localization*, which is the process of adapting a product and technology to a *locale*, that is, a specific language, region or market. The concept “locale” will be described in more detail below. The acronym “l10n” is used because there are ten characters between the first character “l” and the last character “n”.

(Sasaki, 2005) and (Phillips, 2006) demonstrate that internationalization is not a specific feature, but a requirement for software design in general. “Software” can be a text processor, a web service - or a linguistic corpus and its processing tools. For each design target, there are different internationalization requirements.

1.2. Internationalization within the W3C

One task of the Internationalization Activity within the *World Wide Web Consortium* (W3C) is to review a great variety of emerging W3C technologies with respect to internationalization issues[1]. During this work, ongoing topics like character encoding, visualization and processing have to be taken into account for many technologies. This article will describe the relevance of these issues for the design target “multilingual, textual corpus”.

Besides reviewing emerging technologies, the Internationalization Activity is developing technologies itself. This article focuses on two work items, which are of direct relevance for the creation and processing of language resources: A standard for the identification of languages and locales, and markup for internationalization and localization purposes.

2. Ongoing Topics of Internationalization

2.1. Creating a Corpus: Character Encoding Issues

It will be assumed that the design target is a “multilingual, textual corpus”. The corpus should contain existing and yet to be created data in various languages.

In the past, multilingual corpora like the Japanese, English and German data in Verbmobil (Wahlster, 2000) have been created relying only on the ASCII character repertoire. Today the usage of the *Unicode* (Aliprand et al., 2005) character repertoire is common sense, for corpus and many other kinds of textual data. The Basic Multilingual Plane of Unicode encompasses characters from many widely used scripts, which solves basic problems of multilingual corpus design.

However, using Unicode does not solve all problems. Still various decisions have to be made: what encoding form is suitable, how characters not in Unicode are handled, or how to deal with “glyph” variants (see below).

The encoding form is the serialization of characters in a given base data type. The Unicode standard provides three encoding forms for its character repertoire: UTF-8, UTF-16 and UTF-32[2]. If the multilingual corpus contains only Latin based textual data, UTF-8 will lead to a small corpus size, since this data can be represented mostly with one byte sequences. If corpus size and bandwidth are no issues, UTF-32 can be used. However, especially for web based corpora, UTF-32 will slow down data access. UTF-16 is for environments which need both efficient access to characters and economical use of storage.

Unicode encodes widely used scripts and unifies regional and historic differences. Such differences are described as *glyphs*. Unicode unifies many glyphs into singular characters. The most prominent example for the unification of

[1] An overview of past reviews can be found at http://www.w3.org/International/reviews/ .

[2] UTF-8 encodes characters as sequences of a variable length: one, two, three or four bytes. UTF-16 uses variable sequences of one or two double bytes. UTF-32 is a character serialization with a fixed length of four bytes.

glyphs is the *Han unification*, which maps multiple glyph variants of Korean, Chinese and Japanese into a single character repertoire.

As for a multilingual corpus, glyphs characteristics might be quite important. Diachronic glyph variants and rarely used scripts have nearly no chance of becoming a part of the Unicode character repertoire. As one solution to the problem, Unicode provides “variation selectors” which follow a graphic character to identify a (glyph related) restriction on the graphic character. However, it is impossible to have one collection of variation sequences which satisfies all user communities needs.

A different solution would be the encoding of variants as characters. However, since Unicode is an industrial consortium, minority scripts and historical scripts have a small lobby. This makes it difficult for corpus data in such scripts to be represented in Unicode. Fortunately, the *Script Encoding Initiative*[3] has been founded to support proposals to the Unicode consortium for the encoding of rarely used scripts and script elements.

2.2. Visualising a Corpus: Bidirecional Text

In this section, the scope of multilingual corpus design is narrowed to corpora with scripts written from right to left, like Arabic and Hebrew. Unicode has for each character a property “directionality”. This property is used by the bidirectional algorithm (Davis, 2005) to support appropriate order in text visualization. If a corpus contains only one script, the bidirectional algorithm assures proper visualization. HTML(Raggett et al., 1999) uses Unicode as the document character encoding. Hence, the HTML visualization order of the source code “HEBREW” (if written in the Hebrew script) will be “WERBEH”.

However, the Unicode bidirectional algorithm needs help in certain cases of mixed scripts sequences:

```
Source code:
 engl1 ‘‘HEBREW2 engl3 HEBREW4’’ engl5
Visualization a):
 engl1 ‘‘2WERBEH’’
 engl3 ‘‘4WERBEH’’ engl5
Visualization b):
 engl1 ‘‘4WERBEH engl3 2WERBEH’’ engl5
```

In the example, the source code can be visualized as a), that is an English text with two Hebrew citations, or b), that is an English text with a Hebrew citation, which itself contains a English citation. In plain text, visualization b) can be achieved by Unicode control characters. They are inserted to indicate the directional embedding. In text with markup, an attribute like @dir (for “directionality”) in HTML can produce the same effect:

```
Source code plain text:
 engl1 ‘‘*U+202B*HEBREW2 engl3
 HEBREW4*U+202C*’’ engl5
Source code with markup:
 engl1 ‘‘<span dir=’’RTL’’>HEBREW2
 engl3 HEBREW4</span>’’ engl5
```

A query for the length of visualization b) will lead to 25 characters for the source code with markup. As for source code in plain text with control characters, the query will lead to 27 characters. To avoid such influence of directionality indicators, using markup is highly recommended.

Such directionality issues are only one example of the relation between Unicode and markup languages. Further information on the topic is provided by (Dürst and Freytag, 2003).

2.3. Processing Textual Corpus Data

As the corpus is created and can be visualized, the next step is processing of character data. The following processes will be discussed in this section: Counting, normalization and collation sensitive ordering.

A basic process is counting characters. In the Java programming language, regular expressions in Java count character borders: Java takes the beginning of an input sequence into account, even if it is empty. Other technologies count “only” characters. Hence, given the empty input sequence “” and the regular expression “a?”, there will be a match in Java, but not in every other technology.

As for comparison of character sequences, there are two prerequisites. First, the strings have to be in the same encoding. This is not a trivial requirement if massive corpus data is gathered from the Web. (Emerson, 2006) describes character encoding detection issues.

Second, characters have to be in the same *normalization form*. Normalization is the process of bringing two strings to a canonical encoding before they are processed. This is necessary because some character encodings allow multiple representations for the same string. An example: The character “LATIN SMALL LETTER C WITH CEDILLA” can be represented in Unicode as a single character with the code point “U+00E7” or as a sequence “U+0063 U+0327”. Normalization of text and string identify matching is described in detail in the “Character Model for the World Wide Web 1.0: Normalization”(Yergeau et al., 2005). As with character encoding, normalization is of high importance for the creation of mass corpora relying on web data.

The last process discussed for characters is ordering based on *collations*. A collation is a specification of the manner in which character strings are compared and ordered. A simple collation is a code point based collation. It is used for example as the default collation in XPath 2.0 functions (Berglund et al., 2005), which is also used in the XML Query language XQuery (Boag et al., 2005). “Code point based” means that strings are compared relying on the order given by the numeric identifiers of code points. More enhanced collations take specific information into account. For example, a collation might identify the two strings “Strasse” and “Straße” as identical or different. An example of such differences is the order in a German phone book, versus a German lexicon.

3. Current Topic: Language and Locale Identification

The topics of comparisons and collations lead naturally to language and locale identification. In the example above,

[3] See http://www.linguistics.berkeley.edu/sei/ for further information.

there is the same language (German), but two different collations (phone book versus lexicon).

It is crucial to have proper language identification within corpus meta data standards like IMDI (Wittenburg et al., 2000) or OLAC (Simons and Bird, 2003). But language identification is also useful for glyph identification mentioned above, i.e. to separate language and region specific differences for a HAN character.

Corpora being created with XML can make use of the attribute *xml:lang*. It supplies language values in the format described by RFC 3066 (Alvestrand, 2001)[4]. RFC 3066 defines a language tag as follows:

```
Language-Tag =
 Primary-subtag *( "-" Subtag )
Primary-subtag =
 1*8ALPHA
Subtag =
 1*8(ALPHA / DIGIT)
```

A language tag consists of a primary subtag, which can contain 1 to 8 alphabet characters, and a subtag which can contain 1 to 8 alphabet or numeric characters. All values are case insensitive. Two letter primary subtags are interpreted as ISO 639 part 1 language codes (ISO-639, with various publication dates). Three letter primary subtags are ISO 639 part 2 language alpha-3 codes. The second subtag is interpreted ISO 3166 alpha-2 country codes (ISO-3166, with various publication dates). An example of a language tag compliant to this interpretation is "en-US", that is English in US America.

There are some shortcomings of RFC 3066:

- The RFC 3066 grammar is too general for a validation of values.
- There is no stability in the relation between RFC 3066 values and the underlying ISO standards.
- There is no subtag to encode scripts, hence it is impossible to differentiate between e.g. Chinese in the Chinese script and a romanized transliteration.
- The notion of country is different from a region e.g. the political entity "country" may change relatively fast, compared to the region of a speaker community.

All these shortcommings have an impact on the usefulness of language information in a great variety of applications, including metadata about multilingual corpora. Recently a revision of RFC 3066 called *RFC3066bis*[5] was undertaken. It will be introduced in the following section.

3.1. Structure of the Language Tags in RFC3066bis

RFC 3066bis is compatible to RFC 3066: an RFC 3066bis language tag is valid against the grammar of RFC 3066. There are additional constraints defined in RFC 3066bis to differentiate between various types of subtags: script, region, variant, extension and privateuse.

```
langtag = (language
 ["-" script]
 ["-" region]
 *("-" variant)
 *("-" extension)
 ["-" privateuse])
```

The various subtags have the following meaning.

The primary language subtag (2 or 3 letters, 4 letters or 5-8 letters) indicates the language in accordance with ISO639-1 (two letter) or ISO639-2 (three letter). Three letter subtags immediately following the primary subtag are called "extlang". These are reserved for ongoing revisions of ISO 639. An example of a subtag is "de", meaning "German".

The script subtag (4 letters) indicates script or writing system variations, in accordance with /citeISO15924. An example is "de-Latn", meaning "German written with the Latin scrip".

The region subtag (2 letters or 3 digits): 2 letters indicate country, territory, or region, in accordance with ISO3166, part 1. This subtag fulfills the same role as the 2-letter second subtags in rfc3066. 3 digits indicate region information in accordance to UN "Standard Country or Area Codes for Statistical Use" within the region subtag. An example of a subtag is "de-DE", meaning "German in Germany".

The variant subtag (starting with a letter: at least 5 characters; starting with a digit: at least 4 character) indicate variations not associated with an external standard. These must be registered in the IANA subtag registry (see below). An example is "de-Latn-DE-1996", meaning "German written with the latin script in Germany, in the year of 1996)".

The extension subtag (introduced by a single character subtag) indicates an extension to RFC3066bis. RFC3066bis defines mechanisms how such extensions must be registered, which encompasses e.g. the creation of an RFC about the extension. An example is an extension introduced by "r": "en-Latn-GB-r-extended-sequence-x-private".

The privateuse subtag (introduced by "x") indicate a private agreement.

RFC3066bis defines also a new registry called "IANA language subtag registry". It contains not whole language tags, but subtags. All subtags defined already for RFC3066, and all subtags currently (and in the future) available in the underlying ISO standards are part of this registry. RFC306bis introduces two conformance criteria for language tags: "well formed" versus "valid". The former checks the syntax defined above, the latter checks in addition conformance to the language subtag registry.

In addition to the structure of language tags and a registry for subtags, RFC3066bis defines mechanisms for matching values. These encompass matching schemes for filtering (the least specific language tags match) versus lookup (the most specific language tags match).

3.2. Language Tags and Language Resources

RFC3066bis has been designed carefully to fulfill both the needs of the language resource community and of other application areas. The main means for language identification

[4] There is no means in XML for validating that this attributes contains RFC 3066 compliant values. However, many processes mentioned above rely on them.

[5] If finally approved, this revision will have a different RFC number.

of language resources is ISO 639 part 3[6]. Its aim are identifiers for all human languages. This is in contrast to ISO 639 part 1, which focuces on terminology and lexicography, and part 2, which focuces on terminology and bibliography. Part 3 allows for distinguishing extinct, ancient, historic and constructed languages. It lists a very large number of not well-known, yet research relevant languages. ISO 639 part 3 is not approved as a standard yet, but is expected to be an ISO standard at the end of this year. RFC3066bis then will be updated to take ISO 639 part 3 into account.

The W3C Internationalization Activity is working on a document about language and locale identifiers in internet based scenarios[7]. It has two purposes. First, it will provide a common set of identifier (values), which is necessary for any reliable processing of distributed (language) resources. Here the document will mainly rely on RFC3066bis.

Second, the draft will provide a distinction mechanism for separating language and locale. The concept of a locale is important for processing of dates, times, numbers, or currencies. But it is also relevant for linguistic related processing, like the mentioned sort-order (collation). An example of the difference was given above for German ordering conventions (telephone book versus lexicon): Both are conventions for the German language, but with different ordering preferences. Linguistic processing may also rely on the script. It is for example necessary to differentiate Romanized, transliterated Japanese from Japanese in its mainly used version which combines four scripts. Locale definitions can also effect text boundaries (character, word, line, and sentence), or text transformation definitions (including transliterations).

4. Current Topic: Internationalization Tag Set

The purpose of the "Internationalization Tag Set" (ITS) is to provide a set of elements and attributes for common needs of XML internationalization and localization. Various examples of such needs have been described in the previous sections: For example, HTML defines an attribute for directionality, or the working draft for language and locale identifiers defines locale specific information. ITS gathers these state of the art definitions, to enable their application in existing or emerging XML vocabularies.

4.1. ITS: General Approach

ITS encompasses various *data categories* for internationalization and localization and their implementation in XML. The separation of data categories versus their implementation is made to allow for a great variety of usage scenarios, which will be described below. ITS is currently a working draft. A first version of ITS will be finalized within this year.

The following data categories are covered in the current ITS working draft:

- "Translatability" conveys information about whether a piece of textual content in a document should be translated or not.
- "Directionality" conveys the directionality information which is beneficial for visualization of text width mixed directionality.
- "Terminology" indicates terms and is used to add reference information to external resources like terminology data bases.
- "Localization Information" provides a means to add information necessary for the localization process.
- "Ruby" is used to provide pronunciation or further information, in compliance with the W3C Ruby specification (Sawicki et al., 2001).
- "Language Information" is used to specify that a piece of content is of a language, as defined by RFC3066bis.

More and more textual data is being created in XML based formats. Hence, many of these data categories have direct relevance for the language resource community which deals with such formats. For example, (Senellart and Senellart, 2005) describe a methodology to pass information to machine translation tools on the translatability of textual content in XML document. ITS can be used to define such information.

4.2. Simple, Local Implementation of ITS Data Categories

The simple, so-called "local" implementation of the data category "translatability" is an attribute "its:translate" with the values "yes" or "no". It can be attached to any element in an XML document:

```
<text its:translate="no">
... <p its:translate="yes">..."</p>
</text>
```

The attribute expresses information about elements, including child elements and textual content, but excluding attributes. The value of this definition is to have an common agreement on the scope and the values of the data category "translatability", and a unique attribute in a unique XML namespace.

4.3. Global Implementation of ITS Data Categories

The "local" implementation of ITS data categories is used locally in XML documents. In contrast, there is a "global" usage of data categories, which is independent of a specific position:

```
<its:documentRules>
<its:translateRule its:select="//p"
 its:translate="yes"/>
<its:termRule its:select="//qterm"/>
</its:documentRules>
```

[6] See http://www.sil.org/iso639-3/default.asp for further information.

[7] A draft document can be found at http://www.w3.org/International/core/langtags/ .

In the documentRules element, where is an element "translateRule" for the "translatability" data category. It contains an attribute "its:selector". Its value is an XPath expression which selects in the example all "p" elements. The second attribute "its:translate='yes"' expresses that these attributes are translatable.

As for language resources, global usage of ITS is important for the preparation of a variety of processes. For example, the application of a term data base can make use of global rules which define that a specific name in an XML vocabulary is used to mark up terms. Or the translatability data category can be used to differentiate translatable and non-translatable text as input for (semi-)automatic machine translation, as described above.

There is an additional usage of global rules which is important for the combined reuse of a variety of markup schemes.

```
<its:documentRules>
<its:langRule its:select="//*"
 its:langMap="@someAttribute"/>
<its:termRule its:select="//*"
 its:localeMap="@anotherAttribute"/>
</its:documentRules>
```

In the example, it is assumed that a document contains attributes with language or locale information. The "its:select" attributes again select nodes, as in the examples before. The "map" attributes at the "langRule" and "localeRule" elements are used to specify the element or attribute on which the information is available.

```
<text someAttribute="en-US">
...<value
    anotherAttribute="locale-x"/>
...
</text>
```

Given these global rules and the example document above, the value of the "someAttribute" attribute are interpreted as language values. The value of ITS global rules here is that they specify a common semantics for markup: In the ITS tagset working draft, it is specified that the "someAttribute" attribute value has the meaning of RFC3066bis, and that the "anotherAttribute" value is used for locale identification. The benefit is that there is no need to change existing markup to specify this semantics. This is identical to the aim of (Simons et al., 2004). The difference is that (Simons et al., 2004) rely on an RDF representation of markup semantics, while the ITS approach uses an XML representation.

4.4. Relation to other Standardization Efforts and Prospectives for Language Resources

ITS is related to various existing standards and standardiziation efforts. This concerns especially (Savourel, 2005) and XLIFF (Savourel and Reid, 2003). TMX is used to allow easier exchange of translation memory data. The goal of XLIFF is to align data from source and target language(s).

Both of these formats are important *during* the localization process. In contrast, ITS is necessary to assure "localizability" itself: Its aim is to provide proper internationalization, as a requirement for successful localization. From the perspective of language resources, ITS is not meant as a part of a language resource processing scenario. It is rather a means to prepare (large sets of) documents, e.g. to be able to use the same processes of language and locale values for heterogeneous markup schemes.

A perspective for a future version of ITS could be a data category for linguistic markup, e.g. for part of speech units or sentential units. Many efforts have been taken to standardize such markup. ITS could allow for merging such efforts, by declaring levels of linguistic analysis, without forcing people to agree on specific values, e.g. for parts of speech.

5. Conclusion: W3C and Language Resources: Prospective

This article discussed current and ongoing work within the W3C Internationalization Activity and its benefit for the creation and manipulation of language resources. The focus of ongoing work was on issues related to character encoding, order in visualization and character processing. As for current work, the topics of language and locale identification and markup for internationalization and localization purposes were discussed.

Many examples in the sections on current work showed that the work within the i18n Activity and W3C *in general* is driven by its member companies and organizations. However, the work on ITS, previously manly driven by localization aspects, is now being enhanced by aspects which are of direct relevance for the development of multilingual language resources. In this sense, the i18n Activity is a place there two communities meet, i.e. the language resource and the localization community. Both of these communities are dealing with massive linguistic data, but currently there are only a few places of exchange. Hence, one underlying motivation of this paper is also to bring the value of W3C work to the language resource community. W3C develops technologies, which can benefit from the requirements of (multilingual) linguistic applications; and it is a place where communities with interest on language resources can meet to create new technologies together, to their mutual benefit.

6. References

J. Aliprand, Julie Allen, et al., editors. 2005. *The Unicode Standard. Version 4.0.* Addison-Wesley, Boston.

H. Alvestrand. 2001. Tags for the Identification of Languages. Technical report, IETF. http://www.ietf.org/rfc/rfc3066.txt.

A. Berglund, S. Boag, et al. 2005. XML Path Language (XPath) 2.0. W3C Canidate Recommendation. Technical report, W3C. http://www.w3.org/TR/xpath20/.

S. Boag, D. Chamberlin, et al. 2005. XQuery 1.0: An XML Query Language. W3C Candidate Recommendation. Technical report, W3C. http://www.w3.org/TR/xquery/.

M. Davis. 2005. The Bidirectional Algorithm. Unicode Standard Annex #9. Technical report, Unicode Consortium. http://www.unicode.org/reports/tr9/.

M. Dürst and A. Freytag. 2003. Unicode in XML and other Markup Languages. Technical report, W3C and Unicode Consortium. http://www.w3.org/TR/unicode-xml/.

Thomas Emerson. 2006. Large Corpus Construction for Chinese Lexicon Development. In *Proceedings of the 29th Internationalization and Unicode Conference*, San Francisco.

R. Ishida and S. Miller. 2006. Localization vs. Internationalization. Article of the W3C i18n Activity. http://www.w3.org/International/questions/qa-i18n.

ISO-3166. with various publication dates. Codes for the Representation of Names of Countries and their Subdivisions. Technical report, International Organization for Standardization.

ISO-639. with various publication dates. Codes for the Representation of Names of Languages. Technical report, International Organization for Standardization.

A. Phillips. 2006. Internationalization: An Introduction. In *Proceedings of the 29th Internationalization and Unicode Conference*, San Francisco.

D. Raggett, A. Le Hors, and I. Jacobs. 1999. HTML 4.01 Specification. W3C Recommendation. Technical report, W3C. http://www.w3.org/TR/html401.

F. Sasaki. 2005. Internationalization is everywhere. *ACM Ubiquity*, 6. http://www.acm.org/ubiquity/issues6.html.

Y. Savourel and J. Reid. 2003. XLIFF 1.1 Specification. Technical report, OASIS. http://www.oasis-open.org/committees/xliff/documents/xliff-specification.htm.

Y. Savourel. 2005. TMX 1.4b Specification. Technical report, Localisation Industry Standards Association (LISA). TMX 1.4b Specification.

M. Sawicki, M. Suignard, et al. 2001. Ruby Annotation. W3C Recommendation. Technical report, W3C. http://www.w3.org/TR/ruby/.

P. Senellart and J. Senellart. 2005. SYSTRAN Translation Stylesheets: Machine Translation driven by XSLT. In *Proceedings of XML Conference 2005*, Atlanta.

G. Simons and S. Bird. 2003. OLAC Metadata. Technical report, SIL International and others. http://www.language-archives.org/OLAC/metadata.html.

G. F. Simons, W. D. Lewis, et al. 2004. The Semantics of Markup: Mapping Legacy Markup Schemas to a Common Semantics. In *Proceedings of Coling 2004*, Geneva.

W. Wahlster, editor. 2000. *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer, Berlin.

P. Wittenburg, D. Broeder, and B. Sloman. 2000. EAGLES/ISLE: A Proposal for a Meta Description Standard for Language Resources. In *Proceedings of LREC 2000*, Athens.

F. Yergeau, M. Dürst, et al. 2005. Character Model for the World Wide Web 1.0: Normalization. W3C Working Draft. Technical report, W3C. http://www.w3.org/TR/charmod-norm/.