Unicode and XML

François Yergeau
Alis Technologies

20th International Unicode Conference, Washington, DC, USA,
28-31 January 2002

Overview

- Some history
- XML characters and encodings
- Non-ASCII URIs
- Normalization
- Language tagging
- XML 1.1
- XML influencing Unicode

History: SGML

- SGML invented early 1980’s
- Standardized 1986: ISO 8879:1986
- SGML has notions of “Document Character Set” and “Syntax Character Set” to deal with i18n

History: HTML

- HTML invented early 1990’s
- First standardized 1995 by IETF
  - RFC 1866: HTML 2.0
  - RFC 2070: Internationalization of HTML, introduces Unicode as Document Character Set
- W3C takes over
  - HTML 3.2 (1997)
  - HTML 4.0 (1998, includes RFC 2070)

History: XML

- XML 1.0 published 10 February 1998
- Namespaces added early 1999
- XML Base, Xpath, XSL, XML Infoset, XML Schema, etc. added since
- XML 1.0 2nd edition October 2000
- XML 1.1 now in development

XML definitions

- “The document is composed of units called entities.”
- “A parsed entity contains text, a sequence of characters…”
- “A character is an atomic unit of text as specified by ISO/IEC 10646…”

Production [2]

- The whole formal grammar of XML is built on top of Unicode characters.
- A very important production is production [2]:

```
[2] Char::= #x9 | #xA | #xD | [#x20-#xD7FF] |
            [#xE000-#xFFFD] |
            [#x10000-#x10FFFF]


/* any Unicode character, excluding the
   surrogate blocks, FFFE, and FFFF. */
```

Encoding XML documents

- In SGML terms, XML has Unicode as Document Character Set
- Any character encoding compatible with Unicode
- Parsers *must* support both UTF-8 and UTF-16, *may* support any others

Encoding recognition

- Encoding must be known before parsing can start
- Recognition uses first few bytes + the *encoding declaration*

```
<?xml version=‘1.0’ encoding=‘foobar’?>
```

- If absent, encoding defaults to either UTF-8 or UTF-16 (with BOM)

Encoding and well-formedness

- Most encoding errors (not recognized, not supported, illegal byte sequence) are *fatal errors*
- As such, character encoding can be considered part of XML well-formedness

Character references

- `&#xE9;` or `&#233;`
- In XML, always refer to the character number in Unicode (a.k.a. code point, a.k.a. Unicode scalar value)
- Independent of encoding
- No surrogates, `&#x233B4;` not `&#xD84C;&#xDFB4;`

Non-ASCII URIs

- XML 1.0 allows non-ASCII chars in the *system identifier*, an URI
- It specified how to deal with them:
  - Express as UTF-8, encode as %HH
- This has made its way in other W3C specs, also an IETF Internet-Draft

Character Normalization

- Multiple representations for “the same thing”:
  - Multiple character encodings (in different entities)
  - Canonically equivalent representations (precomposed-decomposed) in Unicode
- Big problem for string matching
- String matching is everywhere

Normalization: solution

- Solution is normalization
- Early or late?
- Early:
  - Must choose a canonical form
  - Recipients must check
- Late:
  - Larger burden on *all* recipients
  - Slightly safer

[Charmod]: early, NFC from UTR#15

Language tagging

- XML 1.0 defines **`xml:lang`** attribute, much like HTML’s **`LANG`**
- Values drawn from RFC 3066
  - 1st edition specified RFC 1766, exact syntax
- Hint applies to all attributes and content, until overridden
- Must be declared for validity

XML 1.1

- Work in progress
  - http://www.w3.org/TR/xml11/
- Minor upgrade to deal with character issues:
  - Controls and the NEL character
  - Unicode upgrades vs Names
  - Normalization

Controls

- Most controls excluded from [2] Char in XML 1.0, problem with automatic generation
- XML 1.1 working draft proposes:

```
[2] Char::= [#x1-#xD7FF] | [#xE000-xFFFD]|
            [#x10000-#x10FFFF]
```

Names

- XML 1.0 was based on Unicode 2.0, names (identifiers) restricted to 2.0 characters
- XML 1.1 takes open approach:
  - Almost anything except delimiters, non-characters, U+037E ? (norm. Problem)
  - NameStart further excludes ASCII digits, combining characters
  - Unassigned chars allowed!

NEL

- NEL (U+0085) is used in IBM mainframes as plain text newline
- XML 1.0 docs not plain text

```
S ::= (#x9 | #x20 | #xA | #xD | #x85 |
        #x2028)+
```

- Unicode line separator also added
- Allows straightforward interop for sharing b/w mainframe and others

Normalization

- “XML processors *must/should/may* check whether their input documents are in W3C normalized form, as defined by [Charmod].”
- “It is a *fatal error/error/not an error* for the document not to be in normalized form.”

XML influences on Unicode

- Unicode Technical Report #20
- MathML
- UTF-8 tightening

UTR #20

- *“Unicode in XML and other Markup Languages”*, UTR and W3C Note
  - http://www.unicode.org/unicode/reports/tr20/
  - http://www.w3.org/TR/unicode-xml/
- Unicode is primarily for plain text
  - Needs some control-like functions
  - Compatibility characters
- May conflict or overlap with markup

UTR #20

- Format characters:
  - – Often stateful
  - – Scope often matches markup
  - ➢ Use markup in those cases
  - – Others are *point* functions (nbsp, shy, zwj,…)
  - – Markup may be harmful to searching and sorting
  - ➢ Use Unicode characters

UTR #20

- Compatibility characters:
  - A very mixed bag
  - 3 possible actions depending on case:
    - Retain: when semantic distinction is needed, e.g. math, variant letter forms used as symbol, etc.)
    - Normalize away (KC): presentation forms, fractions, parenthesized letters
    - Markup (+style): list markers, super/subscripts,…
- Read UTR #20 for all the gory details!

MathML

- MathML 1.0 used entities and default styling for math characters
- MathML WG (and others) lobbied UTC for inclusion of a large set of math alphanumerics + other math symbols
- MathML 2.0 now relies on these
  - – http://www.w3.org/TR/MathML2/

UTF-8 tightening

- UTF-8 definition prohibited generation but allowed interpretation of overlong sequences
- Serious security issue, brought to light in part by XML (but IIS incident and others were more spectacular!)
- Resulted in tightening of UTF-8 conformance in Unicode

Q&A

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns:xi=http://www.w3.org/2001/XInclude
           targetNamespace="http://www.w3.org/2001/XInclude">
  <xs:element name="include">
    <xs:complexType mixed="true">
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="xi:fallback"/>
        <xs:any namespace="#other" processContents="lax"/>
      </xs:choice>
      <xs:attribute name="href" type="xs:anyURI" use="required"/>
      <xs:attribute name="parse" use="optional" default="xml">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="xml"/>
            <xs:enumeration value="text"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="encoding" type="xs:string" use="optional"/>
      <xs:anyAttribute namespace="##other" processContents="lax" />
    </xs:complexType>
  </xs:element>

  <xs:element name="fallback">
    <xs:complexType mixed="true">
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="xi:include"/>
        <xs:any namespace="##other" processContents="lax"/>
      </xs:choice>
      <xs:anyAttribute namespace="##other" processContents="l
```