XML and Semantic Web Technologies

II. XML / 1. Unicode, URIs, and XML Syntax

1. Unicode

2. Uniform Resource Identifiers (URIs)

3. XML Syntax
Coded Character Sets

<table>
<thead>
<tr>
<th>name</th>
<th>codes</th>
<th>examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII code</td>
<td>0–127</td>
<td>64 \rightarrow A</td>
</tr>
<tr>
<td>ISO-8859-1, ISO-LATIN-1</td>
<td>0–255</td>
<td>0–127 as ASCII, 196 \rightarrow α</td>
</tr>
<tr>
<td>ISO-8859-7</td>
<td>0–255</td>
<td>0–127 as ASCII, 225 \rightarrow α</td>
</tr>
<tr>
<td>Unicode</td>
<td>0–(2^{32} − 1)</td>
<td>0–255 as ISO-8859-1</td>
</tr>
</tbody>
</table>

Unicode is organized in 256 groups à 256 planes à 256 rows à 256 cells.

Plane 0 (codes 0–65535) is called **basis multilingual plane (BMP)**.

Non ISO-8859-1 characters are mapped to higher codes, e.g., 945 \rightarrow α.
Unicode

Assigned characters of the Unicode standard (v6.0.0, 2011) can be found at http://www.unicode.org/charts/.

Unicode also specifies character classes for each character, as

- letters (capital and small),
- digits,
- punctuation,
- control characters.
The Unicode Character Code Charts

Character Encoding Schemata are trivial for 1-byte coded character sets.

Direct representations of Unicode:

**UCS-2**: direct representation of codes 0–65535 with 2 bytes.

**UCS-4**: direct representation of all codes with 4 bytes.
Drawbacks of direct representations:

- **bytecode 0x00 occurs** (that marks string endings in C), e.g., in UCS-4:

  \[ A \mapsto 65 \mapsto (0, 0, 0, 65) \]

- uniform blow-up of storage space, but most texts mostly use ASCII or ISO-8859-1.

- error-prone, as if one byte is lost, all following data will be decoded incorrectly.

Unicode Transformation Formats (UTF)

Unicode Transformation Formats (UTF) use a variable number of bytes for coding a character.

UTF-8:

- **0x00–0x7f** (bit sequences 0........) code ASCII characters directly,

- **0xc0–0xfd** (bit sequences 11.......) mark the start of a multi-byte character representation (and code its length and leading bits of its code),

- **0x80–0xbf** (bit sequences 10.......) code continuations of multi-byte character representations,

- **0xfe, 0xff** (bit sequences 111111.) are not used.

<table>
<thead>
<tr>
<th>bit sequence</th>
<th>bytes</th>
<th>free bits</th>
<th>character codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0....</td>
<td>1</td>
<td>7</td>
<td>0x00–0x7f</td>
</tr>
<tr>
<td>110....</td>
<td>2</td>
<td>5 + 6 = 11</td>
<td>0x80–0x7ff</td>
</tr>
<tr>
<td>1110....</td>
<td>3</td>
<td>4 + 2 \cdot 6 = 16</td>
<td>0x800–0xffff</td>
</tr>
<tr>
<td>11110...</td>
<td>4</td>
<td>3 + 3 \cdot 6 = 21</td>
<td>0x10000–0x1fffff</td>
</tr>
<tr>
<td>111110..</td>
<td>5</td>
<td>2 + 4 \cdot 6 = 26</td>
<td>0x200000–0x3fffffff</td>
</tr>
<tr>
<td>1111110.</td>
<td>6</td>
<td>1 + 5 \cdot 6 = 31</td>
<td>0x4000000–0xffffffff</td>
</tr>
</tbody>
</table>
II. XML / 1. Unicode, URIs, and XML Syntax

1. Unicode

2. Uniform Resource Identifiers (URIs)

3. XML Syntax

Uniform Resource Identifiers (URIs)

URIs are used to identify resources.

Example:

http://www.ismll.uni-hildesheim.de/lehre/xml-09s/index.html

URIs are defined in RFC 3986 (01/2005).
Generic URI syntax

![Generic URI syntax](http://www.informatik.uni-freiburg.de:8080/secret/top.jsp?id=20&from=1)

```
scheme host port path query
authority

scheme-specific part

mailto: lst@informatik.uni-freiburg.de

scheme scheme-specific part
```

Figure 6: Typical parts of URIs.

Generic URI syntax:

\[
\langle URI \rangle := \langle \text{scheme} \rangle : \langle \text{scheme-specific-part} \rangle
\]

Hierarchical URIs

An URI is called **hierarchical** iff

\[
\langle \text{scheme-specific-part} \rangle := ( \! \! / \! / \langle \text{authority} \rangle \! \! [ \langle \text{path} \rangle ] \\
\quad [ \langle \text{path} \rangle ] \! \! \{ ? \langle \text{query} \rangle \} \! \! \# \langle \text{fragment} \rangle
\]

\[
\langle \text{path} \rangle := ( / \langle \text{path-segment} \rangle )^+
\]

otherwise its called **opaque**.

The path-segments . and .. have special meaning: context path and parent path.

A hierarchical URI is called **server-based** iff

\[
\langle \text{authority} \rangle := [ \langle \text{userinfo} \rangle @ ] \langle \text{host} \rangle [ : \langle \text{port} \rangle ]
\]

otherwise it is called **registry-based**.
Fragment identifiers are used to identify parts of the resource identified by an URI.

Example:

http://www.informatik.uni-freiburg.de/xml/books.html#R03

```
<html>
<body>
<li><a name="EE04">Rainer Eckstein, Silke Eckstein: <em>XML und Datenmodellierung</em>, 2004.</a></li>
</body>
</html>
```

Figure 7: HTML document at http://www.informatik.uni-freiburg.de/xml/books.html.

A relative URI is defined as:

\[
\text{⟨relativeURI⟩ ::= ( // ⟨authority⟩ [ ⟨path⟩ ]
| ⟨path⟩
| ⟨relativePath⟩ ) [ ? ⟨query⟩ ]}
\]

\[
\text{⟨relativePath⟩ ::= ⟨path-segment⟩ (/ ⟨path-segment⟩ )*}
\]

Figure 8: A Base URI is the context for resolving relative URIs [RFC 2396].
URI schemes

URI schemes are managed by Internet Assigned Numbers Authority (IANA).

<table>
<thead>
<tr>
<th>Scheme Name</th>
<th>Description</th>
<th>Reference</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ftp</td>
<td>File Transfer Protocol</td>
<td>RFC 1738</td>
<td>server-based</td>
</tr>
<tr>
<td>http</td>
<td>Hypertext Transfer Protocol</td>
<td>RFC 2616</td>
<td>server-based</td>
</tr>
<tr>
<td>mailto</td>
<td>Electronic mail address</td>
<td>RFC 2368</td>
<td>server-based</td>
</tr>
<tr>
<td>file</td>
<td>Host-specific file names</td>
<td>RFC 1738</td>
<td>server-based</td>
</tr>
<tr>
<td>pop</td>
<td>Post Office Protocol v3</td>
<td>RFC 2384</td>
<td>server-based</td>
</tr>
<tr>
<td>dav</td>
<td>dav</td>
<td>RFC 2518</td>
<td>server-based</td>
</tr>
<tr>
<td>tel</td>
<td>telephone</td>
<td>RFC 2806</td>
<td>opaque</td>
</tr>
<tr>
<td>https</td>
<td>Hypertext Transfer Protocol Secure</td>
<td>RFC 2818</td>
<td>server-based</td>
</tr>
<tr>
<td>urn</td>
<td>Uniform Resource Names</td>
<td>RFC 2141</td>
<td>opaque</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>


Example:

tel:+(49)-761-203-8164

Figure 9: URI types.
URNs are special kinds of URIs that

- **map other namespaces** into URN-space,
- are required to remain **globally unique and persistent** (even when the resource ceases to exist or becomes unavailable).
- have scheme `urn`.

\[
\langle \text{URN} \rangle := \text{urn}: \langle \text{namespace} \rangle : \langle \text{namespace-specific-part} \rangle
\]

Examples:

urn:isbn:0-395-36341-1
urn:newsml:reuters.com:20000206:IIMFFH05643_2000-02-06_17-54-01_L0615

A book or a news item (identified by an URN) may be retrieved from different locations (URLs).

<table>
<thead>
<tr>
<th>URN Namespaces</th>
<th>Value</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>ietf</td>
<td>1</td>
<td>RFC 2648</td>
</tr>
<tr>
<td>pin</td>
<td>2</td>
<td>RFC 3043</td>
</tr>
<tr>
<td>issn</td>
<td>3</td>
<td>RFC 3044</td>
</tr>
<tr>
<td>oid</td>
<td>4</td>
<td>RFC 3061</td>
</tr>
<tr>
<td>newsml</td>
<td>5</td>
<td>RFC 3085</td>
</tr>
<tr>
<td>oasis</td>
<td>6</td>
<td>RFC 3121</td>
</tr>
<tr>
<td>xmlorg</td>
<td>7</td>
<td>RFC 3120</td>
</tr>
<tr>
<td>publicid</td>
<td>8</td>
<td>RFC 3151</td>
</tr>
<tr>
<td>isbn</td>
<td>9</td>
<td>RFC 3187</td>
</tr>
<tr>
<td>nbn</td>
<td>10</td>
<td>RFC 3188</td>
</tr>
<tr>
<td>web3d</td>
<td>11</td>
<td>RFC 3541</td>
</tr>
<tr>
<td>mpeg</td>
<td>12</td>
<td>RFC 3614</td>
</tr>
<tr>
<td>mace</td>
<td>13</td>
<td>RFC 3613</td>
</tr>
<tr>
<td>fipa</td>
<td>14</td>
<td>RFC 3616</td>
</tr>
<tr>
<td>swift</td>
<td>15</td>
<td>RFC 3615</td>
</tr>
</tbody>
</table>

40 URN namespaces (as of 2008-12-09; http://www.iana.org/assignments/urn-namespaces)
Characters Allowed in URIs

In URIs only some characters may be used literally in non-syntactic parts ("data"). All others have to be escaped using their code (in some character encoding):

\[
\langle \text{dataChars} \rangle := \langle \text{alphanum} \rangle | - | _ | . | ! | ~ | * | ' | ( | )
\]

\[
\langle \text{escapedChar} \rangle := \% \langle \text{hexDigit} \rangle \langle \text{hexDigit} \rangle
\]

Codes have been interpreted as codes in different character encodings, depending on the URI scheme.

UTF-8 is recommended by RFC 2718 and already used by some schemes (e.g., urn, imap, pop).

Example:

\[
\text{http://www.informatik.uni-freiburg.de/login.jsp?name=Hans\%20Meyer}
\]

Internationalized Resource Identifiers (IRIs)

IRIs allow more characters to be used literally (RFC 3987; 01/2005).

In IRIs only

- data characters that can be misinterpreted as syntactic characters and
- some bidirectional formatting characters

have to be escaped.

All other data characters are used literally (in some character encoding, e.g., UTF-8).

Example:

\[
\text{http://www.informatik.uni-freiburg.de/login.jsp?name=Hans\%20Müller}
\]

Schemes still are restricted to US ASCII characters.
II. XML / 1. Unicode, URIs, and XML Syntax

1. Unicode

2. Uniform Resource Identifiers (URIs)

3. XML Syntax

W3C development process

W3C specifications are called **Recommendations**.

Stages of W3C recommendations:

<table>
<thead>
<tr>
<th>stage</th>
<th>completion date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>XML 1.0</td>
</tr>
<tr>
<td>Working Draft</td>
<td>1996/11/14</td>
</tr>
<tr>
<td></td>
<td>1997/11/17</td>
</tr>
<tr>
<td>Last Call Working Draft</td>
<td></td>
</tr>
<tr>
<td>Candidate Recommendation</td>
<td></td>
</tr>
<tr>
<td>Proposed Recommendation</td>
<td>1997/12/08</td>
</tr>
<tr>
<td>Recommendation</td>
<td>1998/02/10</td>
</tr>
<tr>
<td>Working Draft (2nd edition)</td>
<td>2000/08/14</td>
</tr>
<tr>
<td>Recommendation (2nd edition)</td>
<td>2000/10/06</td>
</tr>
<tr>
<td>Proposed Edited Recommendation (3rd edition)</td>
<td>2003/10/30</td>
</tr>
<tr>
<td>Recommendation (3rd edition)</td>
<td>2004/02/04</td>
</tr>
<tr>
<td>Recommendation (4th edition)</td>
<td>2006/08/16</td>
</tr>
<tr>
<td>Recommendation (5th edition)</td>
<td>2008/11/26</td>
</tr>
</tbody>
</table>
Every XML document consists of a **prolog** and a single element, called **root element**.

\[
\langle \text{document} \rangle := \langle \text{prolog} \rangle \langle \text{element} \rangle \ (\ (\langle \text{Comment} \rangle \mid \langle \text{PI} \rangle \mid \langle \text{S} \rangle \ )^* \\
\langle \text{prolog} \rangle := \langle ?\text{xml} \langle \text{S} \rangle \text{version} = "1.1" \ \\
\ (\langle \text{S} \rangle \text{encoding} = \langle \text{encoding} \rangle \ )? \ \\
\ (\langle \text{S} \rangle \text{standalone} = ("yes" \mid "no"))? \ \\
\langle \text{S} \rangle?\ ?> \ \\
(\ (\langle \text{Comment} \rangle \mid \langle \text{PI} \rangle \mid \langle \text{S} \rangle \ )^* \\
(\langle \text{DoctypeDecl} \rangle (\ (\langle \text{Comment} \rangle \mid \langle \text{PI} \rangle \mid \langle \text{S} \rangle \ )^* )?)
\]

In all productions

- matching " can be replaced by ’.
- = may be surrounded by spaces (i.e., match \langle S\rangle? = \langle S\rangle?).

\[
\langle \text{S} \rangle := (\#x20 \mid \#x9 \mid \#xD \mid \#xA)+
\]

Figure 10: A minimal XML document with root element "page".

In XML 1.1 the version attribute is mandatory.

If the version attribute is missing, version 1.0 is assumed.
Elements and Attributes

\[
\langle \text{element} \rangle := \langle \text{emptyElementTag} \rangle \\
| \langle \text{STag} \rangle \langle \text{content} \rangle \langle \text{ETag} \rangle
\]

\[
\langle \text{emptyElementTag} \rangle := < \langle \text{Name} \rangle ( \langle S \rangle \langle \text{Name} \rangle = " \langle \text{AttValue} \rangle " )^* \langle S \rangle ? / >
\]

\[
\langle \text{STag} \rangle := < \langle \text{Name} \rangle ( \langle S \rangle \langle \text{Name} \rangle = " \langle \text{AttValue} \rangle " )^* \langle S \rangle ? >
\]

\[
\langle \text{ETag} \rangle := </ \langle \text{Name} \rangle \langle S \rangle ? >
\]

\(\langle \text{Name} \rangle s\)

- start with a unicode letter or _
  ( _ is also allowed, but used for namespaces).
- may contain unicode letters, uncode digits, -, ., or ..

A wellformed document requires,

- that start and end tag of each element match,
- that for each tag the same attribute never occurs twice.

Figure 11: More than one root element.
Not-wellformed Documents (2/2)

The contents of an element can be made up from 6 different things:

1. other elements,
2. Character data,
3. References,
4. CDATA sections,
5. Processing instructions, and
6. comments.

\[
\langle \text{content} \rangle := \langle \text{CharData} \rangle ?
\quad ( ( \langle \text{element} \rangle | \langle \text{Reference} \rangle | \langle \text{CDSect} \rangle | \langle \text{PI} \rangle | \langle \text{Comment} \rangle )
\quad \langle \text{CharData} \rangle ? )^* 
\]

Lars Schmidt-Thieme, Information Systems and Machine Learning Lab (ISMLL), University of Hildesheim, Germany, Course on XML and Semantic Web Technologies, summer term 2012
Character data

(CCharData) may contain any characters except
	<, &, or the sequence > ] ]

Attribute values may not contain

• "", if delimited by "",
• ’’, if delimited by ’’,

These characters can be expressed by references.

<?xml version="1.1"?>
<abstract>
  x^2 = y has no real solution for y < 0.
  But there are solutions for y = 0 & for y > 0.
</abstract>

Figure 14: Forbidden characters in character data.

<?xml version="1.1"?>
<abstract>
  x^2 = y has no real solution for y < 0.
  But there are solutions for y = 0 & for y > 0.
</abstract>

Figure 15: Using references in character data.
References

\[ \langle \text{Reference} \rangle := \langle \text{EntityRef} \rangle \mid \langle \text{CharRef} \rangle \]
\[ \langle \text{CharRef} \rangle := \& \# [0-9]+ ; \]
\[ | \& \#x [0-9a-fA-F]+ ; \]
\[ \langle \text{EntityRef} \rangle := \& \langle \text{Name} \rangle ; \]

There are five predefined entity references:

\[ \&lt; ; \&gt; \&amp; \&apos; \&quot; \]
\[ < > & , " \]

All other entities known from HTML (as \&auml;) are not predefined in XML.

Custom entities can be defined in the document type declaration.

CDATA sections

CDATA sections allow the literal usage of all characters (except the sequence ]]>).

\[ \langle \text{CDSect} \rangle := <! \[ \text{CDATA} \[ \text{\langle \text{Data} \rangle} \] ]> \]

CDATA sections are typically used for longer text containing < or &.

CDATA sections are flat, i.e., there is no possibility to structure them with elements (as < or & are interpreted literally).
Character data and CDATA sections

```xml
<?xml version="1.1"?>
<abstract>
  x^2 = y has no real solution for y &lt; 0.
  But there are solutions for y = 0 & x26; for y & 0.
</abstract>
```

Figure 16: Using numeric character references.

```xml
<?xml version="1.1"?>
<abstract><![CDATA[
  x^2 = y has no real solution for y < 0.
  But there are solutions for y = 0 & for y > 0.
]]></abstract>
```

Figure 17: Using a CDATA-section.

```xml
<?xml version="1.1"?>
<book abstract="Discusses meaning of &quot;wellformed&quot;">  
  <author>John Doe</author>  
  <title>About wellformedness</title>  
</book>
```

Figure 18: Literal usage of attribute delimiter.

```xml
<?xml version="1.1"?>
<book abstract='Discusses meaning of "wellformed"'>  
  <author>John Doe</author>  
  <title>About wellformedness</title>  
</book>
```

Figure 19: Using different attribute delimiters.

```xml
<?xml version="1.1"?>
<book abstract="Discusses meaning of &quot;wellformed&quot;">  
  <author>John Doe</author>  
  <title>About wellformedness</title>  
</book>
```

Figure 20: Using references in attribute values.
Comments can occur in the prolog and in the contents of elements.

Comments are not allowed to contain the character sequence `--`. 

\[
\langle \text{Comment}\rangle := \langle ! -- \langle \text{Char}\rangle^* -- \rangle
\]

```
<?xml version="1.1"?>
<!-- list is not complete yet ! -->
<books>
<!-- yet to be ordered -->
<book>
  <author><fn>Rainer</fn><sn>Eckstein</sn></author>
  <author><fn>Silke</fn><sn>Eckstein</sn></author>
  <title>XML und Datenmodellierung</title>
  <year><!-- look up year of publication --></year>
</book>
</books>
```

Figure 21: Comments in the prolog and in the contents of elements.
<?xml version="1.1"?>
<book>
  <author><fn>Rainer</fn><sn>Eckstein</sn></author>
  <author><fn>Silke</fn><sn>Eckstein</sn></author>
  <title>XML und Datenmodellierung</title>
  <year><!-- edition="1" -->>2004</year>
</book>

Figure 22: Comments in tags are not allowed.

<?xml version="1.1"?>
<books>
  <!-- 2004 ---------------------------------------- -->
  <book>
    <author><fn>Rainer</fn><sn>Eckstein</sn></author>
    <author><fn>Silke</fn><sn>Eckstein</sn></author>
    <title>XML und Datenmodellierung</title>
    <year>2004</year>
  </book>
</books>

Figure 23: -- -- is not allowed in comments.

Processing Instructions

Processing instructions (PIs) allow documents to contain instructions for applications.

\[
\langle PI \rangle := \langle? \langle Name \rangle ( \langle S \rangle \langle Char \rangle^* )? \rangle?
\]

The name of a PI must be different from xml.
Character encoding schemata

Character encoding schemata are specified by the name they are registered with at IANA (http://www.iana.org/assignments/character-sets), e.g.,

- **US-ASCII**
- **ISO-8859-1**
- **ISO-10646-UCS-2** or **csUnicode** (UCS2)
- **ISO-10646-UCS-4** or **csUCS4** (UCS4)
- **UTF-8**
- **UTF-16**
- ... 

If no encoding is specified in the XML declaration, UTF-8 is assumed.

```xml
<?xml version="1.1" encoding="ISO-8859-1" ?>
<page>
  Grüß Gott !
</page>
```

Figure 24: Non-wellformed document (assumed to be ISO-8859-1 coded).

```xml
<?xml version="1.1"?>
<page>
  Grüß Gott !
</page>
```

Figure 25: XML document coded in ISO-8859-1.
There are two predefined attributes,

- `xml:lang`

and

- `xml:space`,

that can be used with any element.

`xml:lang` specifies the language of the character contents of elements and attributes with (RFC 3066)

- an ISO language code
  (http://www.loc.gov/standards/iso639-2/langcodes.html)

  or

- an IANA language code
  (http://www.iana.org/assignments/language-tags).

Example ISO and IANA language codes:

<table>
<thead>
<tr>
<th>language code</th>
<th>meaning</th>
<th>source</th>
</tr>
</thead>
<tbody>
<tr>
<td>de</td>
<td>ISO</td>
<td>German</td>
</tr>
<tr>
<td>de-CH</td>
<td>ISO</td>
<td>German, Swiss variant</td>
</tr>
<tr>
<td>de-DE</td>
<td>ISO</td>
<td>German, German variant</td>
</tr>
<tr>
<td>en</td>
<td>ISO</td>
<td>English</td>
</tr>
<tr>
<td>en-US</td>
<td>ISO</td>
<td>US English</td>
</tr>
<tr>
<td>en-GB</td>
<td>ISO</td>
<td>Britain English</td>
</tr>
<tr>
<td>tlh</td>
<td>ISO</td>
<td>Klingon</td>
</tr>
<tr>
<td>de-1901</td>
<td>IANA</td>
<td>German, traditional orthography</td>
</tr>
<tr>
<td>de-1996</td>
<td>IANA</td>
<td>German, orthography of 1996</td>
</tr>
</tbody>
</table>
<?xml version="1.1"?>
<page>
  <p xml:lang="de">Guten <s>Morgen</s>!</p>
  <p xml:lang="en">Good <s>morning</s>!</p>
  <table>
    <tr><td>USD</td><td>0</td><td>1</td><td>...</td></tr>
    <tr><td>EUR</td><td>0</td><td>0.839818</td><td>...</td></tr>
  </table>
</page>

Figure 26: Language attribute.