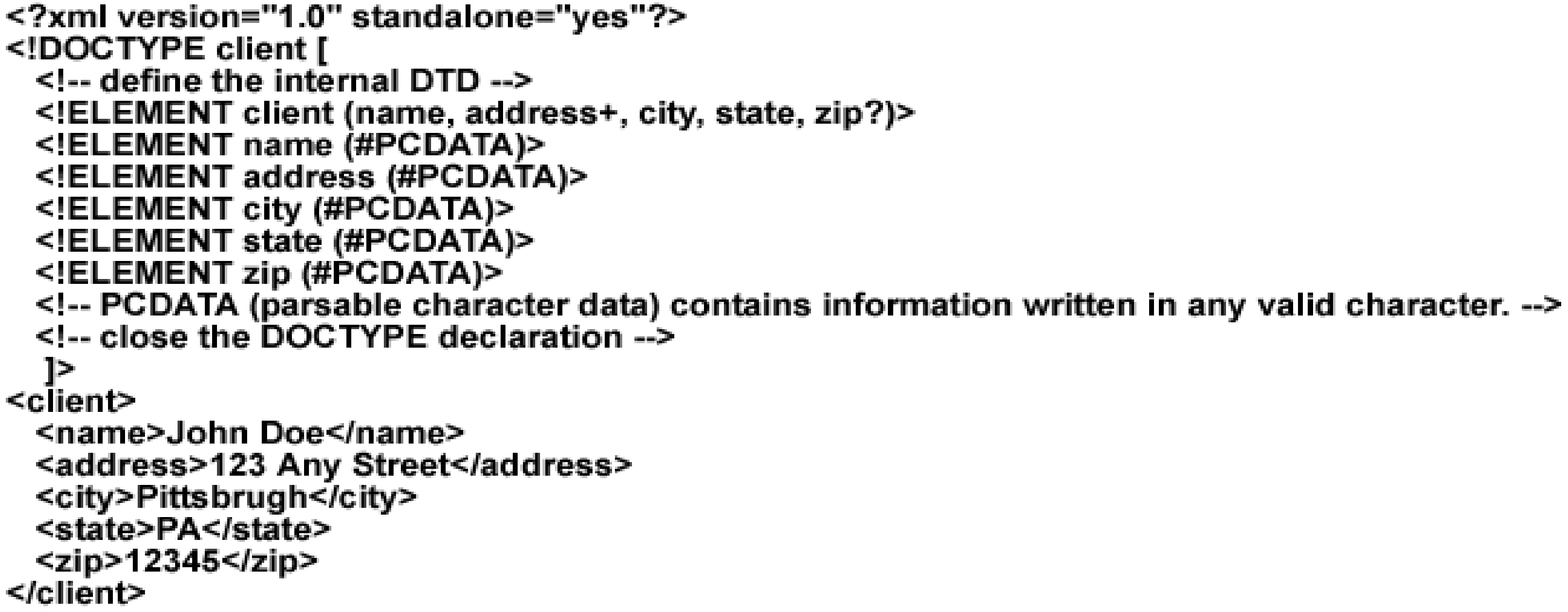
DTD, Namespaces

T. Pitner, L. Bártek, A. Rambousek. L Grolig
FI MU Brno 2020

DTD – Document Type Definition

- **Document Type Definition** (usage/reference to this definition is then a **Document Type Declaration**).
- Specified in the (core) XML standard 1.0.
- Describes allowed element content, attribute presence and content, their default values, defines used entities.
- DTD might be either internal or external DTD (*internal* and *external subset*) or "mixed" – both.
- A document conformant with a DTD is denoted as valid ("platný" in Czech).
- DTD and languages for similar purpose are denoted as modeling languages – they model/define concrete markups.
- Syntax of DTD *IS NOT XML* (in contrast to *XML Schema* and many others modeling languages).

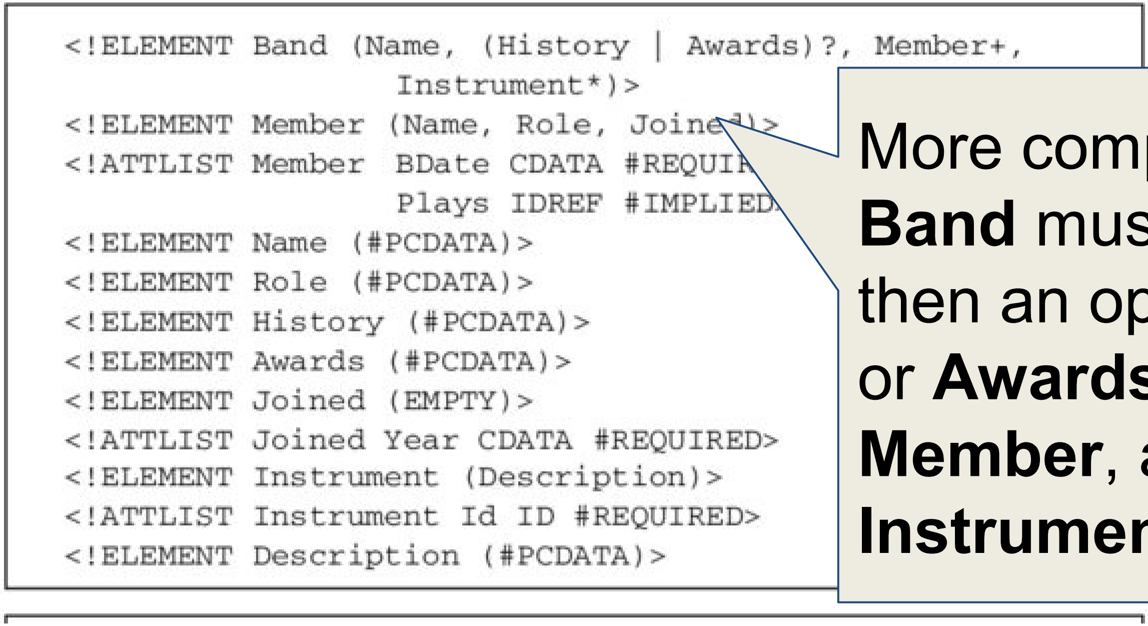
DTD - Sample 1 - Internal DTD



```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE client [
  <!-- define the internal DTD -->
  <!ELEMENT client (name, address+, city, state, zip?)>
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT address (#PCDATA)>
  <!ELEMENT city (#PCDATA)>
  <!ELEMENT state (#PCDATA)>
  <!ELEMENT zip (#PCDATA)>
  <!-- PCDATA (parsable character data) contains information written in any valid character. -->
  <!-- close the DOCTYPE declaration -->
   ]>
<client>
  <name>John Doe</name>
  <address>123 Any Street</address>
  <city>Pittsbrugh</city>
  <state>PA</state>
  <zip>12345</zip>
</client>
```

From Imed Bouchrika, https://www.learndb.com/xml/practice-and-learn-document-type-definition-to-validate-xml

DTD - Sample 2



```
<!ELEMENT Band (Name, (History | Awards)?, Member+,
                   Instrument*)>
<!ELEMENT Member (Name, Role, Joined)>
<!ATTLIST Member  BDate CDATA #REQUIR
                   Plays IDREF #IMPLIED
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Role (#PCDATA)>
<!ELEMENT History (#PCDATA)>
<!ELEMENT Awards (#PCDATA)>
<!ELEMENT Joined (EMPTY)>
<!ATTLIST Joined Year CDATA #REQUIRED>
<!ELEMENT Instrument (Description)>
<!ATTLIST Instrument Id ID #REQUIRED>
<!ELEMENT Description (#PCDATA)>
```

More complex content model - **Band** must include **Name**, then an optional either **History** or **Awards**, then at least one **Member**, and maybe some **Instrument**(s)

https://www.researchgate.net/publication/220919584_Once_Upon_a_Time_a_DTD_Evolved_into_Antother_DTD

DTD - Element type definition

- *Describes allowed content of the element:*
- **`<!ELEMENT element-name ... >`**, where … can be
- **`EMPTY`**
  - for empty element which may be represented as **<element/>** or **<element></element>** with the same logical meaning
- **`ANY`**
  - any content
- **`<!ELEMENT element-name`** *`(specification of child elements)`***`>`**
  - containing both text and child elements given by enumeration
- **`<!ELEMENT element-name (#PCDATA |`** *`specification of child elements`***`)*>`**
  - For MIXED, the order or cardinality of concrete child elements cannot be specified. The star (*) is required and any number of occurencies is always allowed

DTD - Attribute(s) definition

- *Describes (data) type and/or implicit attribute values for the respective element.*

```
<!ATTLIST element-name attribute-name attribute-value-type implicit-value>
```

- or a list of attributes:

```
<!ATTLIST ename attr-name1 attribute-value-type1 implicit-value1
attr-name2 attribute-value-type2 implicit-value2>
```

DTD - Issues and Limitations

- DTD still in (heavy) use
- Sufficient for modeling of simpler markups
- Replace with **XMLSchema** (or **RelaxNG** alternatively)
- No advanced modeling - no more primitive data types (integers, floats, date/time, boolean), no inheritance, no detailed element content modeling, reuse not very handy

XML Namespaces

- They define **logical spaces for names of elements, attributes** in XML document.
- Therefore, they give the elements and attributes the "**third dimension**".
- To each NS in XML, there is exactly one ("globally") **unique identifier**, given by **URI** (URI is a superset of URL).
- NS corresponding to an URI does not anyhow relate to content that would potentially be available under the URL ("nothing is downloaded when processing NSs").
- Instead of URIs for denoting a namespace in document, one uses prefixes for these NS mapped to the respective URI using **`xmlns:prefix="URI"`**.

XML Namespaces

```
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en" lang="en">
  <body>
     <h1>Huraaaa</
  </body>
</html>
```

Declaration of **implicit namespace** (for **entire document**)

XML Namespaces

```
<xhtml:html
xmlns:xhtml="http://ww    w3.org/1999/xhtml"
xml:lang="en" lang

  <xhtml:body>

     <xhtml:h1>Hur

  </xhtml:body>

</xhtml:html>
```

Declaration of **prefixed namespace** for **entire document** wherever prefix is used

XML Namespaces

- Element- or attribute-name containing colon (:) is denoted as *Qualified Name*, *QName*.
- Two NS are equal iff their URIs are *one-to-one-character* the same (in UNICODE).
- Namespaces *do not* apply to *text nodes*.
- Element/attribute *need not* be in a namespace.
- NS prefix declaration or declaration or the implicit NS recursively applies to all descendants (child elements, their children etc.), unless another declaration "remaps" the given prefix.
- One NS is co-called **implicit** (default) NS, declared by attribute xmlns=
- Default NSs *do not apply to attributes*!!!, thus attributes without an explicit prefix do not belong to any NS.
- NS *are not compatible* with DTD.