XML Namespaces

URI: Names and Locators

Addressing of resources in the WWW

URL	:=	Uniform Resource Locator
URN	:=	Uniform Resource Name
URI	:=	URN ∪ URL
		Uniform Resource Identifier
Resource	:=	an object that is addressable by a URI

Examples for URIs

- **http://www.darmstadt.gmd.de/IPSI/index.html**
- **http://rdbm.com/sql.cgi?query="select * from foo"**
- **urn:ISBN:0-395-36341-6**

Note:

- **Even if a URI looks like an address, it does not necessarily specify a location!**

Idea

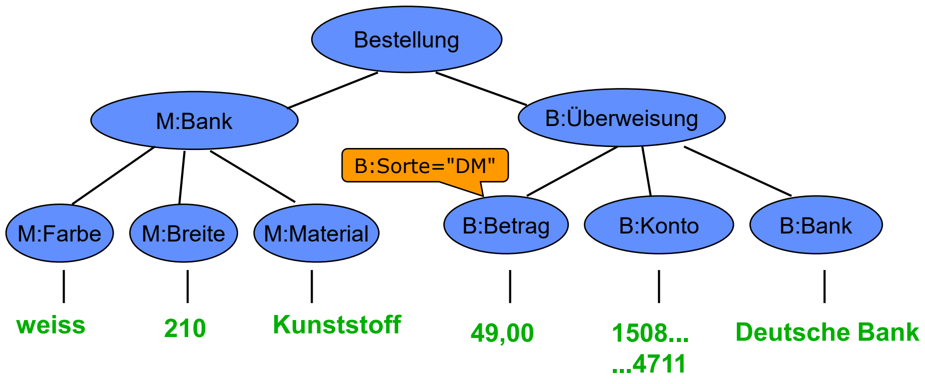
Problem

- Often collections of names that have been created independent from each other have to be combined. In doing so we have to avoid name collisions
- We want to group names to allow an application to quickly find out whether a name belongs to a group

How to create globally unique names?

- Central control of names
  - Problem: difficult implementation
- Hierarchical names
  - used in XML
  - a name is composed of a URI and a local name

Example Namenspaces



```
<Bestellung   xmlns:B     =     "http://bank.de/schema/"
              xmlns:M     =     "http://moebel.de/schema/">
```

Implementation

Problems

- The full URI is often very long
- The XML syntax does not allow the character '/' in names

Solution

- We define an abbreviation for the URI
  this abbreviation is used as a prefix for the name
- Example: we can declare that "xsl" is an abbreviation for the namespace' “http://www.w3.org/TR/WD-xsl”
  Afterwards we can use for example the tag "xsl:template‘‘
- The prefix "xsl" has no magic meaning
  We could have defined instead "xyz" as abbreviation for “http://www.w3.org/TR/WD-xsl” and then write the tag as "xyz:template‘‘
- Applications like an XSLT-processor always use the URI and the local name to recognize elements.
  On the other hand, the string "xsl:template" is useless if we have not assigned the correct URI to the prefix "xsl‘‘

Remind: The URI just has to unique (and stable). It need not refer to location in which a DTD or a schema is stored

Names

The XML-specification allow to use a colon in names (for element-types, attributes, entities, etc.)

However, a colon should only be used in accordance with the namespace-standard

- Names of element-types and attributes must contain at most one colon
  The part before the colon is the namespace abbreviation
  The part after the colon is the local name
  The namespace abbreviation has to be declared
- If the name of an element-type does not contain a colon, it belongs to the default namespace if such a default namespace is declared; otherwise, it has no namespace
- Attributes without namespace abbreviation are a special case (see below) ;
  they are not part of the default namespace

Declaration of Namespaces (1)

A namespace is declared in a start-tag by using an attribute with a name that is composed of the string "xmlns", a colon and the namespace abbreviation;
the value of the attribute is the URI

- Example: <Beispiel xmlns:xsl="http://www.w3.org/TR/WD-xsl" a="...">

The declared namespace abbreviation is visible for the element itself as well as for all descendants (if it is not redefined in a descendant element)

- As the declaration is visible for the element itself it is common to use it immediately:
  <xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl" version="1.0">

As namespace declaration are inherited to descendants, namespaces are often declared in the ''document element" (the root element) to be used in the whole document

Declaration of Namespaces (2)

We can declare a default-namespace that is used for all elements that are not explicitly assigned to a namespace (i.e., where the name does not contain a colon)

- We use the attribute "xmlns" with the namespace-URL as value
- Example:

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"
                xmlns="http://www.w3.org/TR/REC-html40"
                version="1.0">
```

Namespaces of Attributes

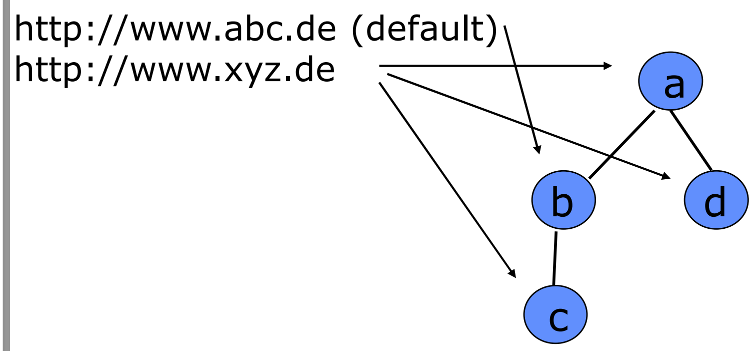
A declaration of a default-namespace is not valid for attributes

Attributes without a namespace-prefix (“local attributes”) are identified by their name and the element to which they belong (and the element is identified by its local name an namespace)

- In this example the namespace of "version" is not the HTML-URL!

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"
                xmlns="http://www.w3.org/TR/REC-html40"
                version="1.0“>
```

Exercise



http://www.abc.de (default)
http://www.xyz.de