Semi-structured Data

3 - Namespaces
Outline

• The Need for Namespaces
• Namespace Syntax
• Default Namespace
• Multiple Namespaces
A Common Problem

- Merging of XML documents often leads to conflicts

<!-- Students’ Evaluation -->
<course>
  <title>SSD</title>
  <assessment>Fair</assessment>
</course>

<!-- University’s Evaluation -->
<course>
  <title>SSD</title>
  <assessment>Elective</assessment>
</course>

- The two assessment elements are semantically different

- How we distinguish these two elements?
Solution 1 - Renaming

• Simply *rename* the assessment elements

  <studassessment> Fair </studassessment>

  <univassessment> Elective </univassessment>

• … but there are some weaknesses:
  o The new element names are *not transparent*
  o *We may get* new *conflicts* in the future
Solution 2 - Refined Renaming

• Rename the elements, but use a separator

  <stud:assessment> Fair </stud:assessment>

  <univ:assessment> Elective </univ:assessment>

• … but still:
  o Although the new element names are transparent
  o We may get new conflicts in the future
Solution 3 - Unique Names

- We can exploit URIs (Uniform Resource Identifier)
  - http://www.oeh.ac.at - Austrian Students’ Union
  - http://www.tuwien.ac.at - TU Wien


- Transparent and unique element names

- But, the new document is not well-formed - not valid XML names
Final Solution - Namespaces

- Combination of solutions 2 and 3 - Namespaces
- Mechanism to associate the prefixes stud and univ with the URIs

```xml
<!-- Students’ and University’s Evaluation -->
<course
    xmlns:stud="http://www.oeh.ac.at"
    xmlns:univ="http://www.tuwien.ac.at">
    <title>SSD</title>
    <stud:assessment>Fair</stud:assessment>
    <univ:assessment>Elective</univ:assessment>
</course>
```

**ATTENTION:** Namespace URIs are simply identifiers, they are not followed as links
The Need for Namespaces

Namespaces have two purposes in XML:

- **Disambiguating elements and attributes**
  Distinguish between elements and attributes from different vocabularies that share the same name but are semantically different

- **Grouping elements**
  Group related elements and attributes together so that programs can easily recognize them
Namespace Syntax

• A namespace declaration is of the form:

   xmlns:prefix="name"

where prefix is an XML name, and name is a URI

• It appears as an attribute in an element

   <course
     xmlns:stud="http://www.oeh.ac.at"
     xmlns:univ="http://www.tuwien.ac.at">
Namespace Syntax

• For elements and attributes qualified names are used of the form
  
  prefix:local-name

  where both prefix and local-name are XML names

  
  <stud:assessment> Fair </stud:assessment>

  <univ:assessment> Elective </univ:assessment>
Default Namespace

- We can have a default namespace declared as xmlns="name"
- We simply remove the prefix

```xml
<!-- Students’ and University’s Evaluation -->
<course xmlns="http://www.oeh.ac.at"
        xmlns:univ="http://www.tuwien.ac.at">
    <title>SSD</title>
    <assessment>Fair</assessment>
    <univ:assessment>Elective</univ:assessment>
</course>
```

**ATTENTION:** Default namespace applies only to unprefixed elements, not attributes
Multiple Namespaces

- We can redefine a prefix or the default namespace

<!-- Students’ and University’s Evaluation -->
<course xmlns= "http://www.tuwien.ac.at">
  <title>SSD</title>
  <assessment xmlns="http://www.oeh.ac.at">
    Fair
  </assessment>
  <assessment>Elective</assessment>
</course>
Multiple Namespaces

• The closest ancestor with a namespace declaration takes precedence

• If there is no declaration among the ancestors:
  o For the default namespace the empty namespace is used
  o For a prefix we get an error (when the prefix us used)