Ch. 13: Using XML Namespaces

Internet Database Laboratory, Seoul National University

July, 2015

Contents

- Populating an XML Namespace
- XML Schemas, XML Documents, and Namespaces
- Referencing XML Schema Components in Namespaces
- Namespaces and Validating XML
- Adding All Locally Defined Elements
- XML Schemas in Multiple Files
- XML Schemas with Multiple Namespaces
- The Schema of Schemas as the Default
- Namespaces and DTDs
- XSLT and Namespaces

Populating an XML Namespace

- To populate a namespace: Type targetNamespace="URI" in the root element

xsd

```
<?xml version="1.0"?>

<xs:schema
 xmlns:xs = "http://www.w3.org/2001/XMLSchema"
 targetNamespace = "http://www.kehogo.com/ns/wow/1.0">

<xs:element name="ancient_wonders">
   <xs:complexType>
    <xs:sequence>
    <xs:element name="wonder"
            type="wonderType"
            maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

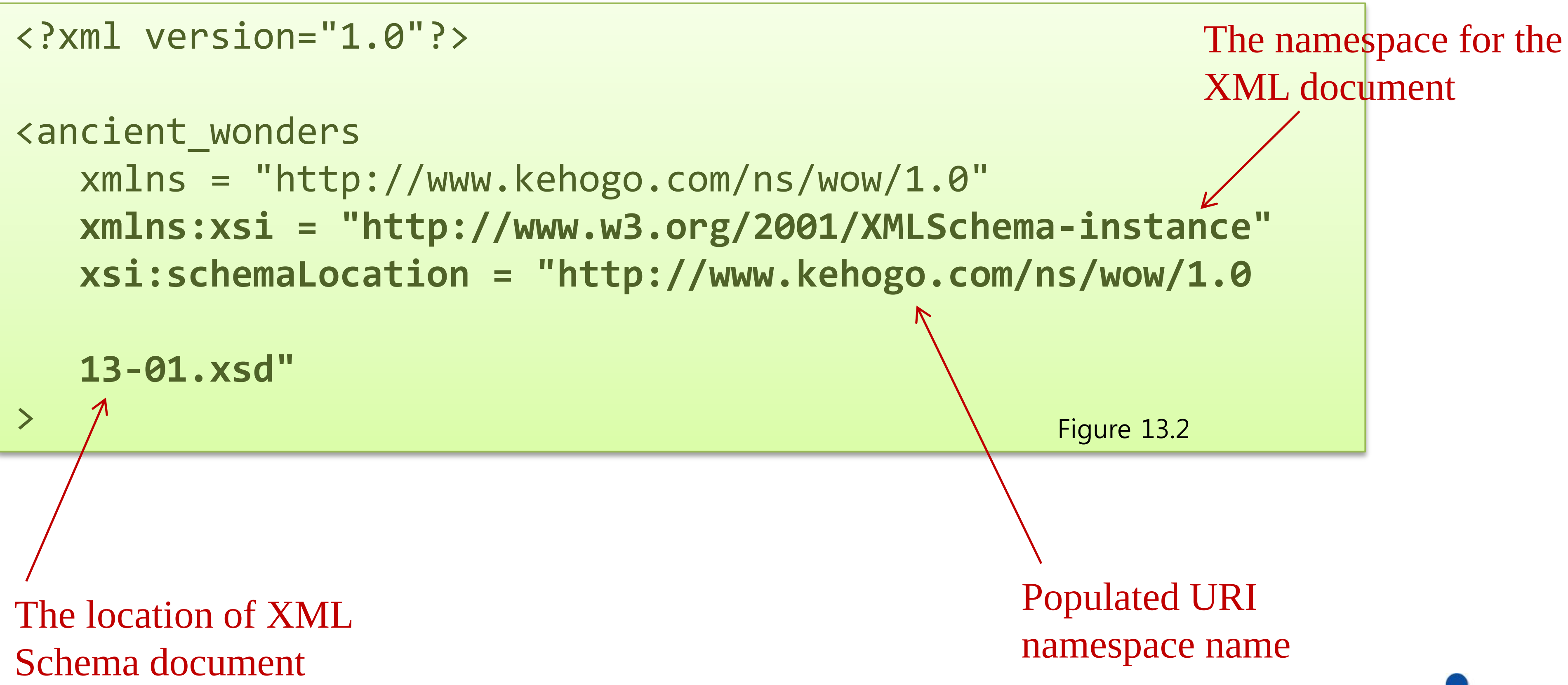
Namespace URI

Figure 13.1

XML Schemas, XML Documents, and Namespaces

- To indicate the location of an XML Schema and the namespace it populated:
  Type xmlns:xsi=“http://www.w3.org/2001/XMLSchema-instance”

xml



```
<?xml version="1.0"?>

<ancient_wonders
   xmlns = "http://www.kehogo.com/ns/wow/1.0"
   xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation = "http://www.kehogo.com/ns/wow/1.0

   13-01.xsd"
>
```

Figure 13.2

The namespace for the XML document

Populated URI namespace name

The location of XML Schema document

Referencing XML Schema Components in Namespaces

- To declare a default namespace for XML Schema components
  - Type xmlns="URI" in the root element
- To declare a namespace with a prefix for XML Schema components
  - Type xmlns:prefix ="URI" in the root element
- To reference the components in the XML Schema
  - Type prefix:component_name

xsd

```
<?xml version="1.0"?>

<xs:schema xmlns:xs = "http://www.w3.org/2001/XMLSchema"
           xmlns = "http://www.kehogo.com/ns/wow/1.0"
           targetNamespace="http://www.kehogo.com/ns/wow/1.0">

<xs:element name="ancient_wonders">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="wonder“ type="wonderType“ maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Namespace URI

Figure 13.3

Namespaces and Validating XML

- To write XML documents with qualified elements
  - Type xmlns:prefix="URI" in the root element

xml

```
<?xml version="1.0"?>
<wow:ancient_wonders
   xmlns:wow="http://www.kehogo.com/ns/wow/1.0"

   xmlns:xsi = "http://www.w3.org/2001/XMLSchemainstance"
   xsi:schemaLocation = "http://www.kehogo.com/ns/wow/1.0

   13-03.xsd“
>
```

Figure 13.4

Namespace URI

Adding All Locally Defined Elements

- To add all locally defined elements to the target namespace
  - Type elementFormDefault="qualified"

xsd

```
<?xml version="1.0"?>

<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"

xmlns="http://www.kehogo.com/ns/wow/1.0"

targetNamespace="http://www.kehogo.com/ns/wow/1.0"

elementFormDefault="qualified">

...>
```

Figure 13.5

The locally declared elements will be associated with the target namespace

Adding All Locally Defined Elements

- It is considered best practices to add all locally defined elements when populating a namespace

xml

```
<wow:ancient_wonders
xmlns:wow="http://www.kehogo.com/ns/wow/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.kehogo.com/ns/wow/1.0
13-05.xsd">

<wow:wonder>
  <wow:name language="English">Colossus of Rhodes</wow:name>
  <wow:name language="Greek">Κολοσσός της Ρόδου</wow:name>
  <wow:location>Rhodes, Greece</wow:location>
  <wow:height units="feet">107</wow:height>
  <wow:history>
    <wow:year_built era="BC">282</wow:year_built>

…>
```

Figure 13.6

Adding Particular Locally Defined Elements

- To add a particular locally defined element to the target namespace
  - Type form=“qualified”

xsd

```
<?xml version="1.0"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.kehogo.com/ns/wow/1.0"
targetNamespace="http://www.kehogo.com/ns/wow/1.0"
elementFormDefault="qualified">

<xs:element name="ancient_wonders">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="wonder"
      type="wonderType“ maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Figure 13.7

XML Schemas in Multiple Files

- Divide the XML Schema components among files
  - Here, there is new XML Schema with the civType complex type definition

xsd

```
<?xml version="1.0"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.kehogo.com/ns/wow/1.0">

<xs:complexType name="civType">
  <xs:sequence>
    <xs:element name="name“ type="xs:string"/>
    <xs:element name="locale“ type="xs:string"/>
    <xs:element name="period“ type="xs:string"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

Figure 13.8

XML Schemas in Multiple Files

- To include XML Schema components
  - Type <xs:include schemaLocation=“includefile.uri” /> after xs:schema

xsd

```
<?xml version="1.0"?>

<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.kehogo.com/ns/wow/1.0"
targetNamespace="http://www.kehogo.com/ns/wow/1.0">

<xs:include schemaLocation="13-08.xsd"/>
```

Figure 13.9

XML Schema document

XML Schemas with Multiple Namespaces

- To import components from XML Schemas with different target namespaces
  - <xs:import namespace=“URI” schemaLocation=“schema.uri” />

xsd

```
<?xml version="1.0"?>

<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.kehogo.com/ns/wow/1.0"
targetNamespace="http://www.kehogo.com/ns/wow/1.0"
xmlns:anc_civ="http://www.kehogo.com/ns/ancient_civ/2.3">

<xs:import
namespace="http://www.kehogo.com/ns/ancient_civ/2.3"
schemaLocation="13-10-anc_civ.xsd"/>
```

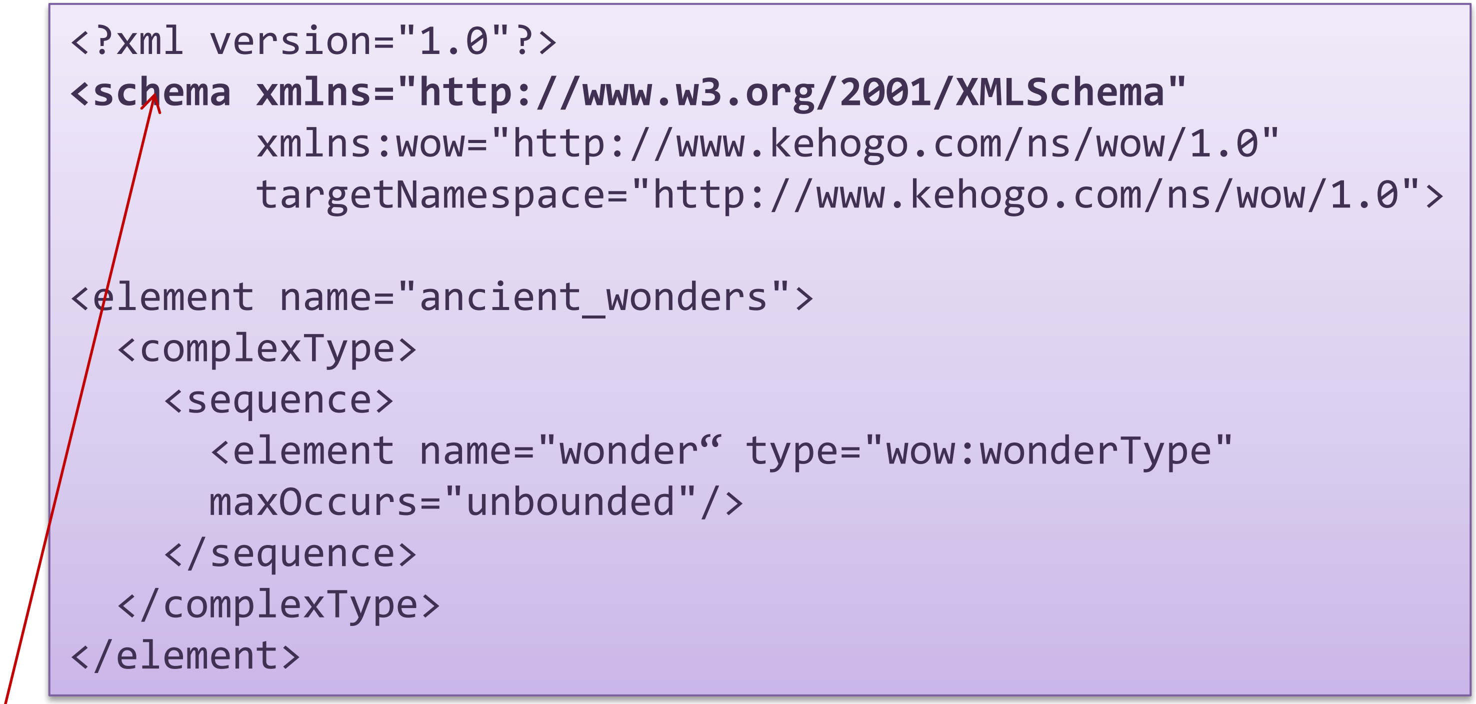
Figure 13.10

Namespace name of the XML Schema

XML Schema document

The Schema of Schemas as the Default

- To declare the Schema of Schemas as the default namespace
  - <schema xmlns=http://www.w3.org/2001/XMLSchema >



```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:wow="http://www.kehogo.com/ns/wow/1.0"
        targetNamespace="http://www.kehogo.com/ns/wow/1.0">

<element name="ancient_wonders">
  <complexType>
    <sequence>
      <element name="wonder“ type="wow:wonderType"
      maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```

<schema> element

xsd

Figure 13.11

Namespaces and DTDs

- DTDs do not support XML namespace declarations

- For example, the element **`wow:height`**,
  - Does not mean the **`height`** element in the **`wow:`** namespace,
  - It just means the **`wow:height`** element

XSLT and Namespaces

- To use an XML namespace in an XSLT Style Sheet
  - Within the opening tag of root element, type xmlns:prefix
  - Then type =“URI”, where URI identifies the name of the XML namespace

```
<?xml version="1.0"?>

<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0"
xmlns:wow="http://www.kehogo.com/ns/wow/1.0">

<xsl:template match="/">
<html><head><title>Wonders of the World</title></head>
<body>
<h1 align="center">Seven Wonders of the Ancient World</h1>

…>
```

xslt

Figure 13.12

The name of the XML namespace