XML Entities

Author: Mihai Rotaru

Date: 1 Dec 2011

XML entities can be used to store pre-defined text fragments, which can then be used throughout the XML document. The entities are replaced with the text they represent; such entities are known as 'General Entities'.

Another type of entities are the 'Parameter Entities' - which are to be used only inside the DTD. It is important to distinguish between them because they occupy different namespaces, and are recognized only in their respective context. (http://www.w3.org/TR/REC-xml/REC-xml-20081126.xml#sec-physical-struct)

Not distinguishing between the two types of entities can be a source of errors; for example, we might have the following XML document:

```
<?xml version="1.0"?>
<!DOCTYPE my_doc [
<!ENTITY % ParEntity "This is a parameter entity" >
<!ENTITY GeneralEntity "This is a general entity" >
<!ELEMENT my_doc (#PCDATA | item )*>
]>
<my_doc>
&ParEntity;
</my_doc>
```

This document is not valid, because ParEntity is a parameter entity. Because general and parameter entities are in different namespaces, we can simply declare another general entity, with the name ParEntity:

```
<!ENTITY ParEntity "This is another general entity" >
```

The document would then be valid. Obviously, having two entities with the same name is generally not a good idea, since it can easily lead to confusion, and is best avoided when possible.

General entities can hold data other than text, for example images or sounds. But these powerful capabilities open the door for a range of mistakes, normally of no concern to XML.

One example is the location of the resource, specified after SYSTEM or PUBLIC inside the external entity declaration; it is known as the system identifier, and is meant to be converted to a URI reference. Relative paths are normally relative to the document within which the entity is declared, but the final URI might be affected by the requirements of particular DTD's, or might be interpreted in a non-conventional manner by applications. This can lead to the inability to correctly resolve the URI. Additionally, the incorrect NDATA might be specified, which will lead to the file being interpreted as the wrong type of file.