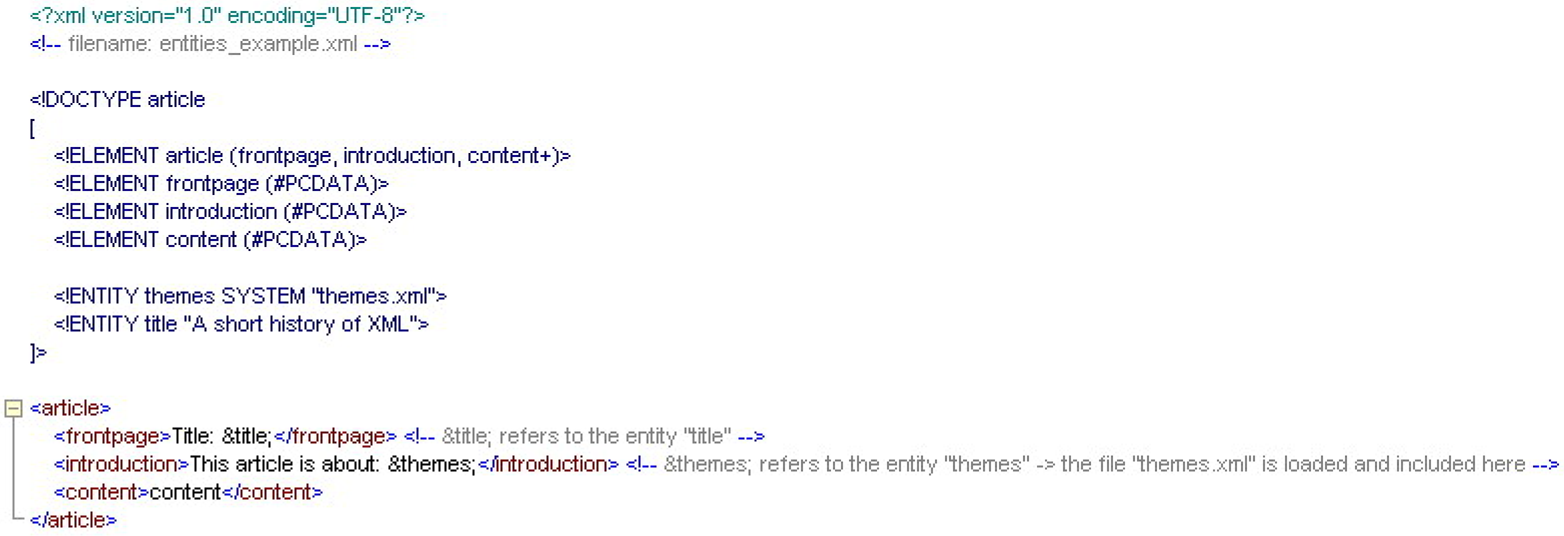
Entities

- according to the XML specification, the term "entity" is used for the following circumstances:
  - the whole XML document itself (a so-called document entity)
  - an external DTD
  - an external file which is declared as an external entity within the DTD; the XML document contains a reference to the external file
  - a string, surrounded by quotation marks which is declared as an internal entity within the DTD; the XML document contains a reference to the string
- to reduce the size of XML documents
- to modularize XML documents
- to integrate different types of data

Using Entities

- files_03/entities_example.xml



```
<?xml version="1.0" encoding="UTF-8"?>
<!-- filename: entities_example.xml -->

<!DOCTYPE article
[
    <!ELEMENT article (frontpage, introduction, content+)>
    <!ELEMENT frontpage (#PCDATA)>
    <!ELEMENT introduction (#PCDATA)>
    <!ELEMENT content (#PCDATA)>

    <!ENTITY themes SYSTEM "themes.xml">
    <!ENTITY title "A short history of XML">
]>

<article>
    <frontpage>Title: &title;</frontpage> <!-- &title; refers to the entity "title" -->
    <introduction>This article is about: &themes;</introduction> <!-- &themes; refers to the entity "themes" -> the file "themes.xml" is loaded and included here -->
    <content>content</content>
</article>
```

Types Of Entities (I)

- general vs. parameter entities

- general entities
  - contain XML text or other text used in the XML document (within the root element)
- parameter entities
  - contain XML text (markup code) used within the DTD

Types Of Entities (II)

- internal vs. external entities

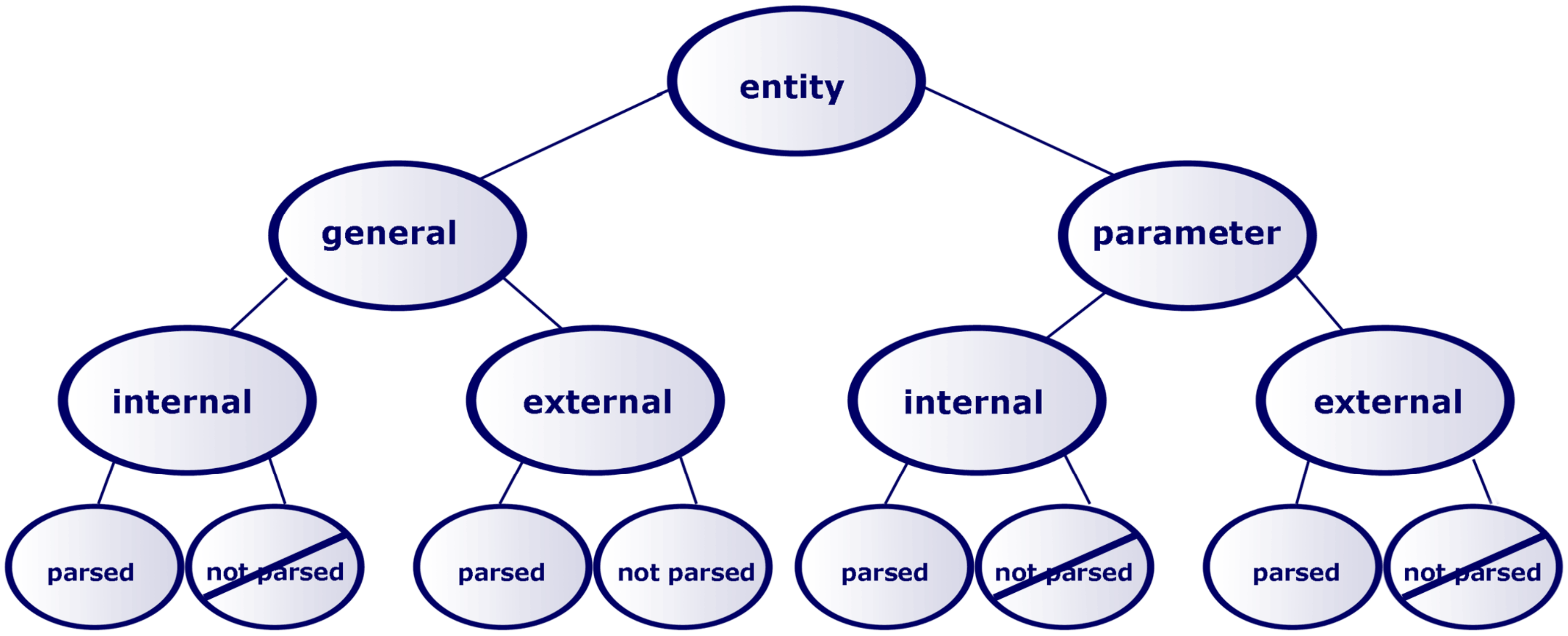
- internal entities
  - strings, surrounded in quotation marks
- external entities
  - are saved in separate files

Types Of Entities (III)

- parsed vs. unparsed entities

- parsed entities
  - contain XML text; the reference is replaced by the entity's content which then is parsed by the XML processor
- unparsed entities
  - contain XML data, text or other data (e.g. images); the XML processor does not access the external file but passes it on to the application that processes the XML document

Types Of Entities (IV)



Entity Names And Values

- names
  - names may only start with letters or an underline, not with a number
  - names may not start with "xml"
  - the XML processor is case sensitive
- values
  - values must be surrounded by apostrophes or quotation marks
  - values may not contain the surrounding character
  - values may not contain the characters & and % (exceptions can be found in chapter 4 of the XML specification, see http://www.w3.org/tr/rec-xml)
  - values must be valid in the context where they are to be inserted

Internal Parsed General Entities

- declaration
  - `<!ENTITY` *`name`* `"`*`value`*`">`
- example
  - `<!ENTITY abb "(This is an abbreviation)">`

```
<content>PHP &abb;</content>

<content>HTML &abb;</content>
```

External Parsed General Entities

- declaration
  - `<!ENTITY` *name* `SYSTEM "`*filename*`"`
- example

  -
    ```
    <!ENTITY themes SYSTEM "http://www.myhomepage.com/themes.xml">


    <description>You'll find all themes here: &themes;</description>
    ```

External Unparsed General Entities

- declaration
  - `<!ENTITY` *`name`* `SYSTEM "`*`filename`*`" NDATA` *`notation_name`*`>`
- example

  -
    ```
    <!ENTITY cover SYSTEM "cover.gif" NDATA GIF>
    <!ELEMENT cover_image EMPTY>
    <!ATTLIST cover_image source ENTITY #REQUIRED>

    <cover_image source="cover" />
    ```
- the keyword `NDATA` indicates that the file contains unparsed data

Internal Parsed Parameter Entities

- declaration
  - `<!ENTITY % name "value">`
- example

  -
    ```
    <!ENTITY author "
      <!-- the author's information -->
      <!ELEMENT author (name, birthdate)>
      <!ELEMENT name (#PCDATA)>
      <!ELEMENT birthdate (#PCDATA)>">
    ```
- the value must include at least one complete markup declaration, corresponding to the DTD where it is used

External Parsed Parameter Entities

- declaration
  - `<!ENTITY % name SYSTEM "filename">`
- example

  -
    ```
    <!DOCTYPE library [
      <!ELEMENT library (book | cd)*>
      <!ENTITY % book SYSTEM "book.dtd">
      <!ENTITY % cd SYSTEM "cd.dtd">

      %book;
      %cd;
    ]>
    ```

Inserting Entity References

- general entities:
  - `&name;`
- parameter entities:
  - `%name;`
- each entity must be declared before it is referred

Notations

- declaration
  - `<!NOTATION` *`name`* `SYSTEM "`*`system_literal`*`">`
- example
  - `<!NOTATION html SYSTEM "c:\program_files\mozilla\firefox.exe">`

- notations are always used with external unparsed general entities
- the system literal is not accessed by the XML processor, it is only forwarded to the application that processes the XML document (i.e. a webpage script)
- the system literal describes the format (normally either by using a URI referring to more information or by specifying a program to process the format)

Inserting Character References

- to insert characters or letters that can not be found on the keyboard, you can use decimal or hexadecimal codes corresponding to the ISO/IEC-10646 code (see http://www.iso.org/iso/en/ISOOnline.frontpage for details)
- decimal for "A": &#65;
- hexadecimal for "A": &#x41;
- more:
  - http://unicode.e-workers.de/entities.php
  - http://www.htmlhelp.com/reference/html40/entities/

Predefined Entities (excerpt)

&	&#38;	&amp;
<	&#60;	&lt;
>	&#62;	&gt;
'	&#39;	&apos;
"	&#34;	&quot;