avans hogeschool

Module XML

Week 2: Schema's (vervolg)/XSLT

Vandaag

- Schema's – deel 2
- XML met CSS
- XML met XSL(T)

Schema: Groeperen

- **xsd:all**
  - *volgorde van elementen niet belangrijk*
  - *ieder element mag hooguit 1x voorkomen in de groep*
- **xsd:choice**
  - *er moet 'precies 1' element uit de groep voorkomen*
  - *aantal voorkomens via minOccurs en maxOccurs te regelen*
- **xsd:sequence**
  - *elementen exact volgens opgegeven volgorde*
  - *aantal voorkomens via minOccurs en maxOccurs te regelen*
- voorbeelden:
  - *wk2\leden_all.xsd*
  - *wk2\brief.xsd*
  - *wk2\leden_choice.xsd*

Schema: afleiden van SimpleType

- **3 types afleiding mogelijk:**
  - ***restriction***: *selecteert subset van de waardes van het basistype*

```
<xsd:simpleType name="phonoYearType">
  <xsd:restriction base="xsd:gYear">
    <xsd:minInclusive value="1877"/>
  </xsd:restriction>
</xsd:simpleType>
```

  - ***enumeration***: *specificeert een lijst van waardes*

```
<xsd:simpleType name="muziekdragerType">
   <xsd:restriction base="xsd:string">
    <xsd:enumeration value="LP"/>
    <xsd:enumeration value="CD"/>
    <xsd:enumeration value="tape"/>
   </xsd:restriction>
</xsd:simpleType>
```

  - ***union***: *combineert meerdere types*

Schema: afleiden van SimpleType

Facetten:

- *xsd:minInclusive: alle instanties moeten >= deze waarde zijn*
- *xsd:minExclusive: alle instanties moeten > deze waarde zijn*
- *xsd:maxInclusive: alle instanties moeten <= deze waarde zijn*
- *xsd:maxExclusive: alle instanties moeten < deze waarde zijn*
- *xsd:enumeration: lijst van toegestane waardes*
- *xsd:whiteSpace: hoe wordt white space binnen het element behandeld*
- *xsd:pattern: reguliere expressie waaraan instantie moet voldoen*
- *xsd:length: exacte aantal characters in een string, items in een list of bytes in binary data*
- *xsd:minLength: minimum lengte*
- *xsd:maxLength: maximum lengte*
- *xsd:totalDigits: maximum aantal toegestane digits in het element*
- *xsd:fractionDigits: maximum aantal toegestane digits in fractionele deel van het element*

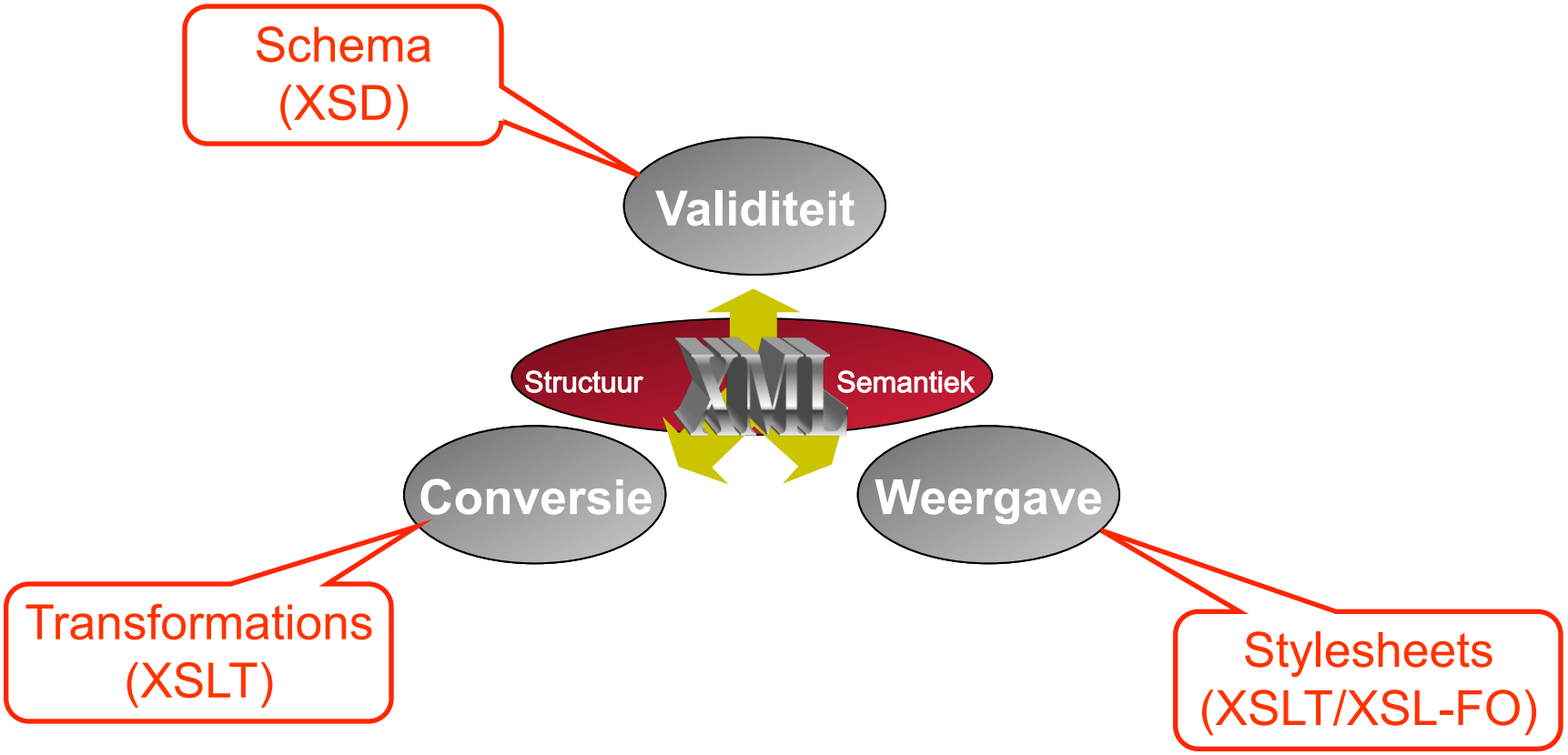
Schema: afleiden van ComplexType

- extension (net als in OO):

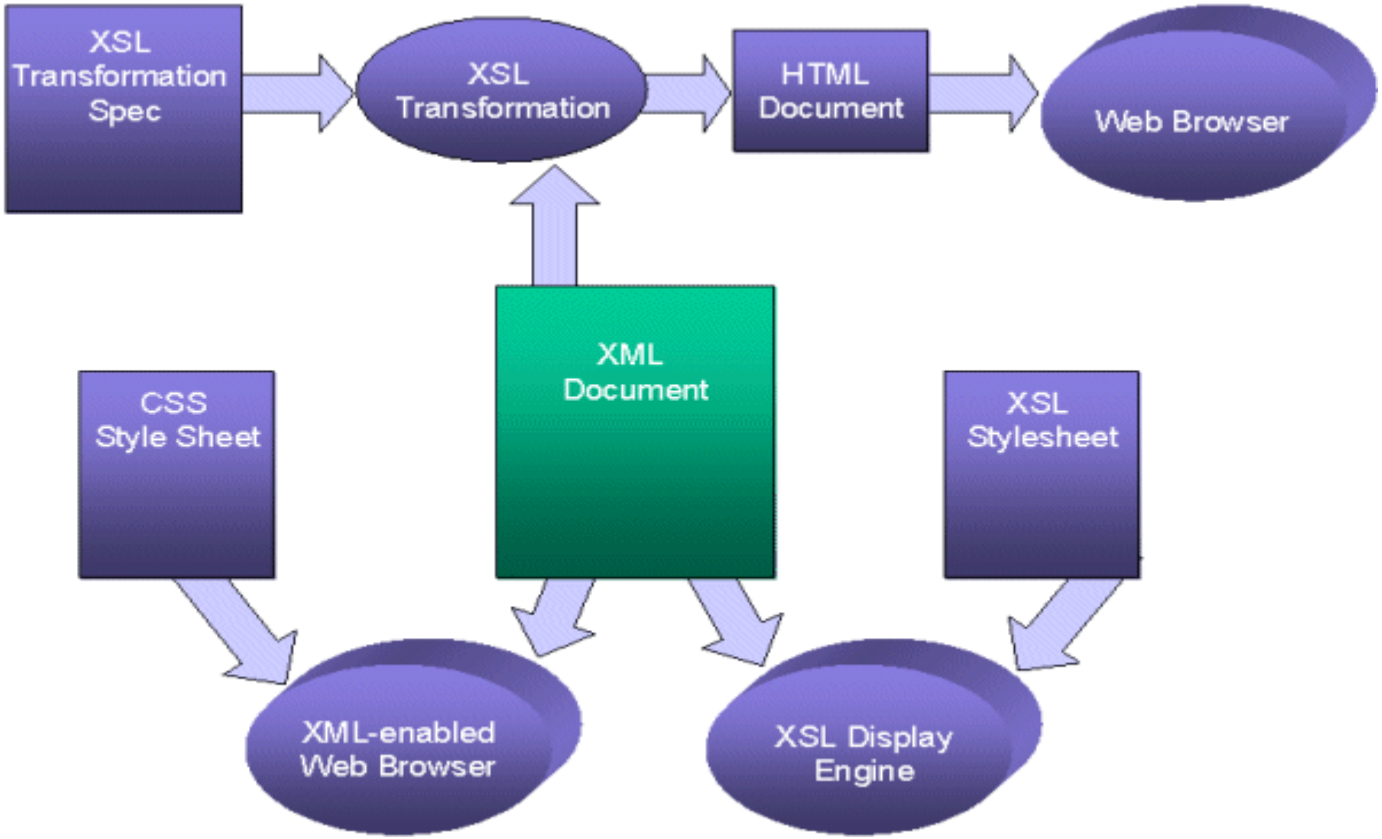
```
 <xsd:complexType name="starWarsType">
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string"/>
      . . .
      <xsd:element name="home" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>

<xsd:complexType name="wookieType">
  <xsd:complexContent>
    <xsd:extension base="starwarsType">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

De grenzen van XML



Hoe toon je XML?



XML weergeven met CSS

Waarvoor dient CSS?

- Opmaak-definitie van één of meer documenten
- Opmaak scheiden van structuur, bedoeld voor HTML
- Definities voor fonts, kleuren, achtergronden, teksten, boxjes, classificaties, en plaatsing van elementen.
- Cascading: stijldefinitie parent-element geldt automatisch voor child-element tenzij child-element eigen stijldefinitie krijgt.

XML weergeven met CSS

Hoe gebruik je een CSS?

- Koppeling naar de stylesheet:
  ```
  <?xml-stylesheet type="text/css" href="aii.css">
  ```
- Stijl-definitie voor de elementen in het CSS-file:
  element {eigenschap: waarde; eigenschap: waarde; …}
  ```
  kennisgebied {
      display: block;
      font-size: 16pt;
      font-weight: bold;
      color: blue
  }
  ```
- Voorbeeld: wk2\aii_css.xml

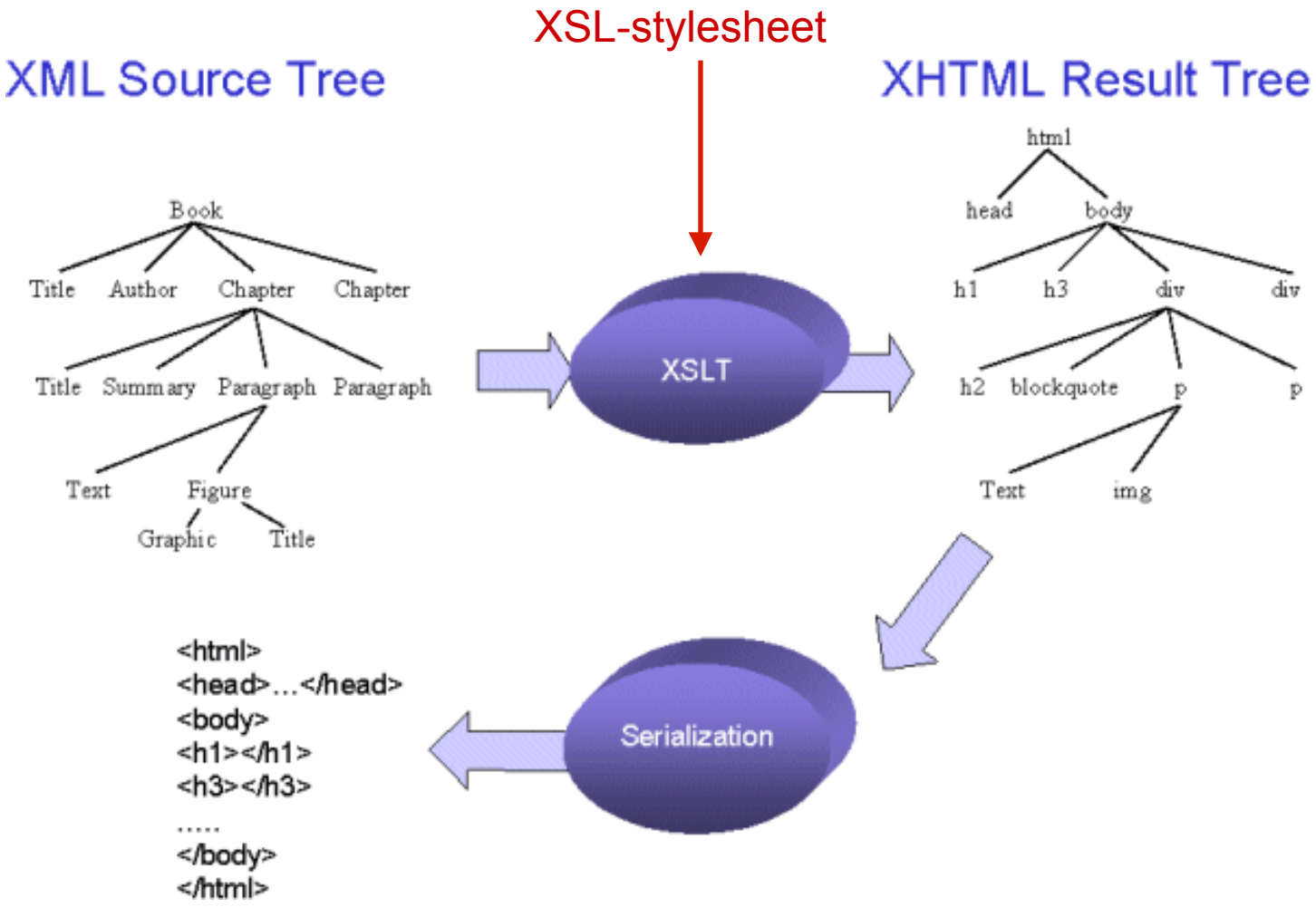
CSS of XSL(T)?

	CSS	XSL
Kan XML transformeren (bijv. naar HTML)	Nee	Ja
Elementen selecteren en sorteren	Nee	Ja
Nieuwe elementen toevoegen	Nee	Ja
Elementen verwijderen	Nee	Ja
Gespecificeerd in XML	Nee	Ja
Eenvoudig	Ja	Nee
Onderhoudsvriendelijk	Ja	Ja

Wat is XSL?

- Extensible Stylesheet Language is een op XML gebaseerde taal om stylesheets mee te maken
  - *Is dus een XML document: well-formedness!*
  - *Is gekoppeld aan eigen namespaces*
- Deze stylesheets worden door een XSL-engine gebruikt om XML-documenten om te zetten in andere document-types
- In XML wordt het invoer document wel source tree genoemd en het uitvoer document result tree.
- XSL is geschreven in XML

Source tree & Result tree



XSL talen

- XSL = XSLT + XSL-FO

- XSLT = XSL for Transformations
  - *ene 'helft' van XSL*
  - *een taal die XML-bronbestanden kan transformeren in andere formaten*
- XSL-FO= XSL Formatting Objects
  - *andere 'helft' van XSL*
  - *een taal die gebruikt wordt om XML documenten te beschrijven voor het tonen ervan*
  - *veel meer mogelijkheden dan CSS (bv PDF's)*
  - *lastiger om te gebruiken*

XSLT

- Om XSLT transformaties toe te passen zijn drie dingen nodig:
  - *een XML document om te transformeren*
  - *een XSLT stylesheet: bevat instructies voor de transformatie die plaats moet vinden*
  - *en een XSLT engine: stukje software dat instructies gaat uitvoeren (meeste gratis: AltovaXML, XT, MSXML, …)*

XSLT – wat kun je zoal?

- Transformaties:
  - *Het genereren/toevoegen van tekst (bv HTML-tags)*
  - *Weglaten van content (deel van XML gebruiken)*
  - *Verplaatsen van data (v.b. het wisselen van de voor- en achternaam)*
  - *Dupliceren van data (v.b. kopiëren van de tekst voor het maken van een inhoudsopgave)*
  - *Selecteren/Sorteren/Filteren*
  - *…*
- M.a.w.: Complexe transformaties die nieuwe informatie genereren uit bestaande informatie

Tussenopdracht 1+2

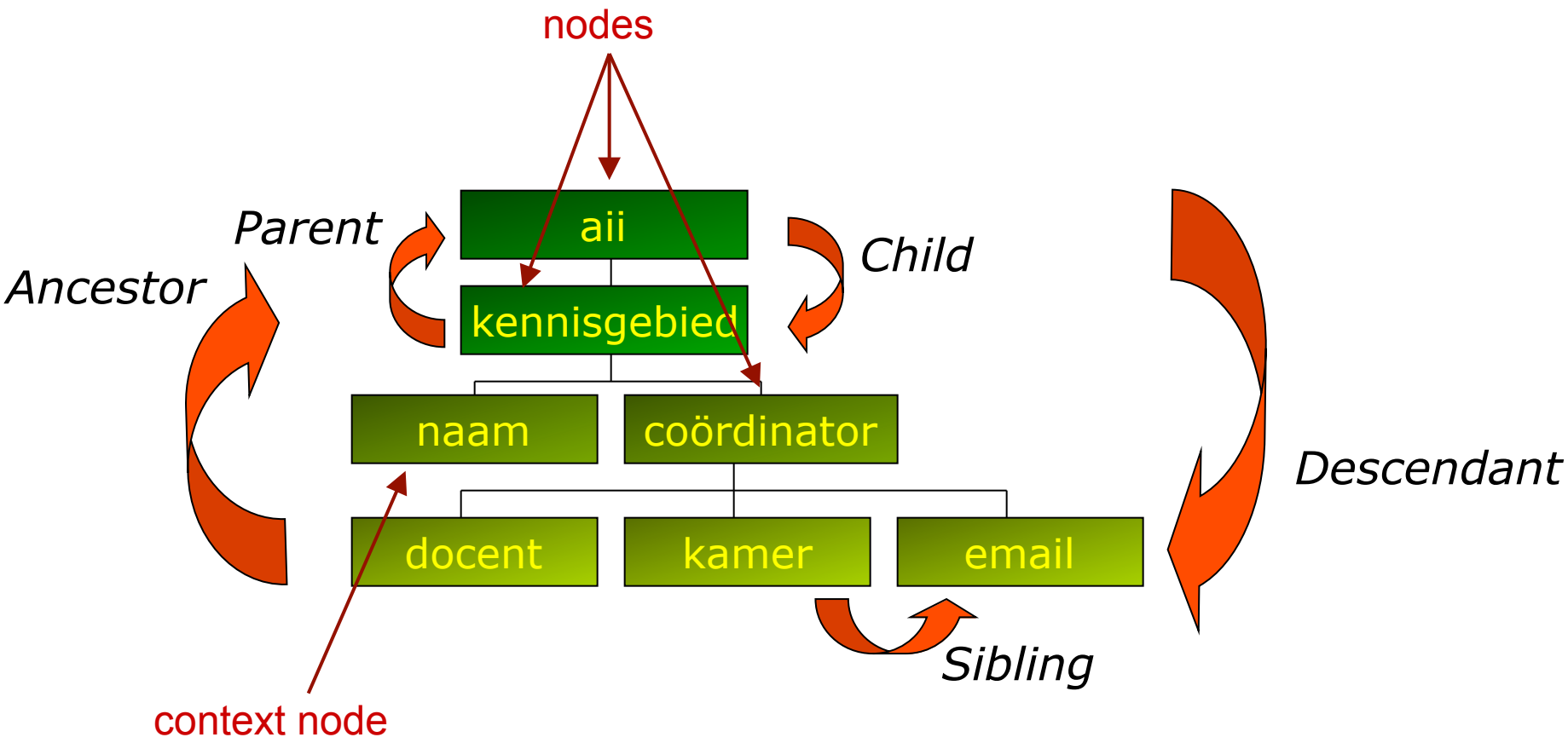
XSLT – stylesheets & templates

- XSLT stylesheets worden gebouwd op templates
- Een template beschrijft wat de transformatie van de invoer naar de uitvoer is voor het betreffende deel
- XSLT gebruikt tekst die letterlijk overgenomen wordt en XSLT instructies (voorafgegaan door 'xsl-prefix')

```
<xsl:template match="/">
…
  <xsl:apply-templates select="xxx"/>
…
</xsl:template>

<xsl:template match="xxx">
  … do your thing …
</xsl:template>
```

Hiërarchie in XML



Templates & XPath - 1

- XPath wordt gebruikt om "de weg" te vinden binnen een XML document

- XML gebruikt de term node om te verwijzen naar een willekeurig gedeelte van een document, of het nou een element, een attribuut, commentaar of iets anders is.

- Belangrijk is de term document root: de "virtuele" root helemaal bovenaan in de hiërarchie
  - *Naar deze root wordt verwezen met behulp van het symbool "/".*
  - *definitie van template voor root van document:*
    ```
    <xsl:template match="/">
    ```

Templates & XPath - 2

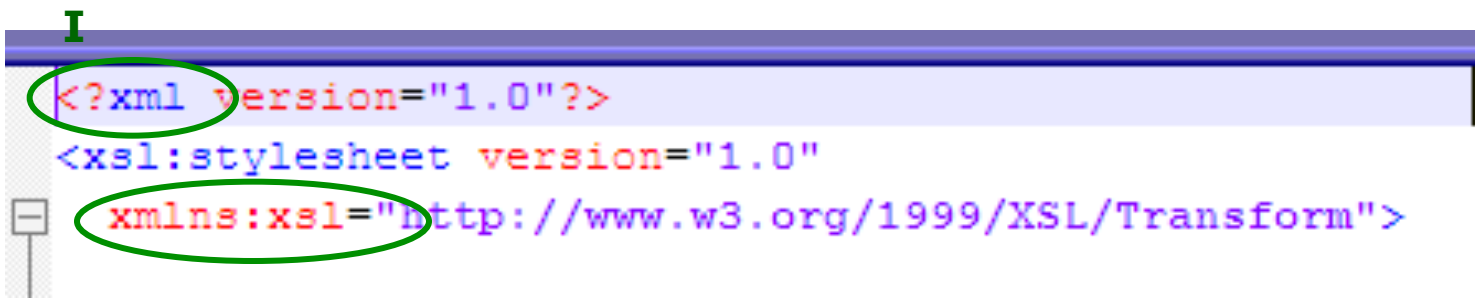
- Location path geeft aan waar je heen gaat in een document.
- Actuele plaats context node.
- Elk template zorgt er voor dat de context node verandert in het "pad" dat bij het match-attribuut staat.

```
<xsl:template match="xxx">
    … do your thing …
</xsl:template>
```

- XPath uitdrukkingen binnen template zijn relatief tov context node.

- Let op verwijzing naar XSl-stylesheet!
- Voorbeeld 1: wk2\aii_xsl_ie.xml & wk2\aii_xsl_ff.xml
- Voorbeeld 2: wk2\persys_all.xml

Voorbeeld XSL - 1



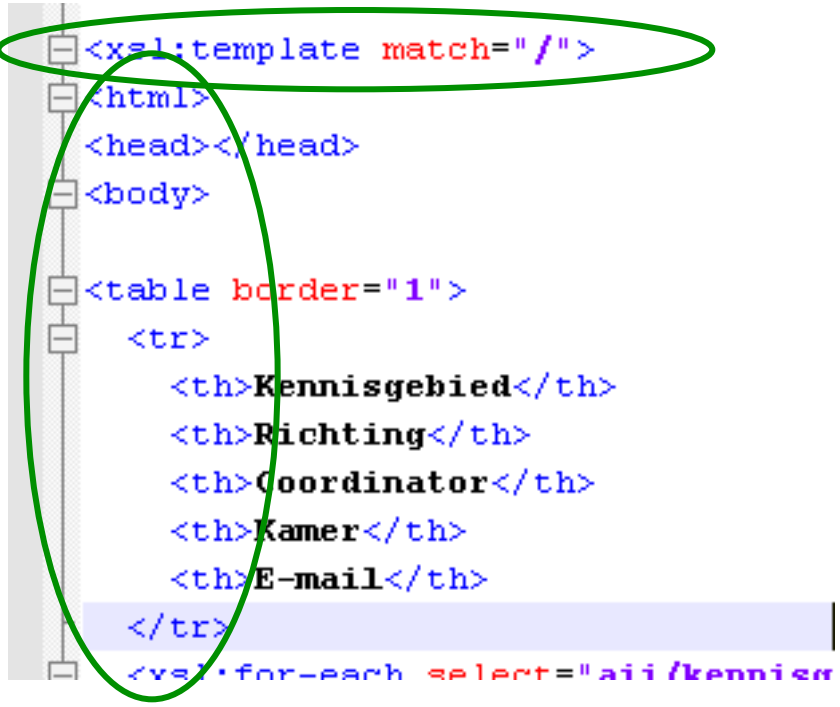
I Ook het stylesheet is een XML-document
- *==> well-formedness (ook voor de HTML-tags!)*

II Één namespaces met prefixes:
- *xsl (ondersteund door IE7/FF)*

Voorbeeld XSL - 2

III

IV



III Hier begint het template voor de root-node

IV Spuugt alleen maar HTML uit (tot nu toe)

Voorbeeld XSL - 3

V VI VII VIII

```
<xsl:for-each select="kennisgebied">
  <xsl:sort select="@richting" />
    <tr>
      <td><xsl:apply-templates select="naam"/></td>
      <td><xsl:apply-templates select="@richting"/></td>
      <td><xsl:apply-templates select="coordinator/docent"/></td>
      <td><xsl:apply-templates select="coordinator/kamer"/></td>
      <td><xsl:apply-templates select="coordinator/email"/></td>
    </tr>
  </xsl:for-each>
```

V XSL kan selecteren; let op dat het XML is
VI Voorbeeld van XPath
VII XSL kan sorteren
VIII Formatteercode in een aparte ‘template’

Voorbeeld XSL - 4

```
    </xsl:for-each>
</table>

</body>
</html>
</xsl:template>

<xsl:template match="naam">
    <i><xsl:value-of/><xslt:text>:</xslt:text></i>
</xsl:template>

<xsl:template match="@richting">
    <center><xsl:value-of/></center>
</xsl:template>

<xsl:template match="coordinator/docent">
    <xsl:value-of/>
</xsl:template>

<xsl:template match="coordinator/kamer">
    <xsl:value-of/>
```

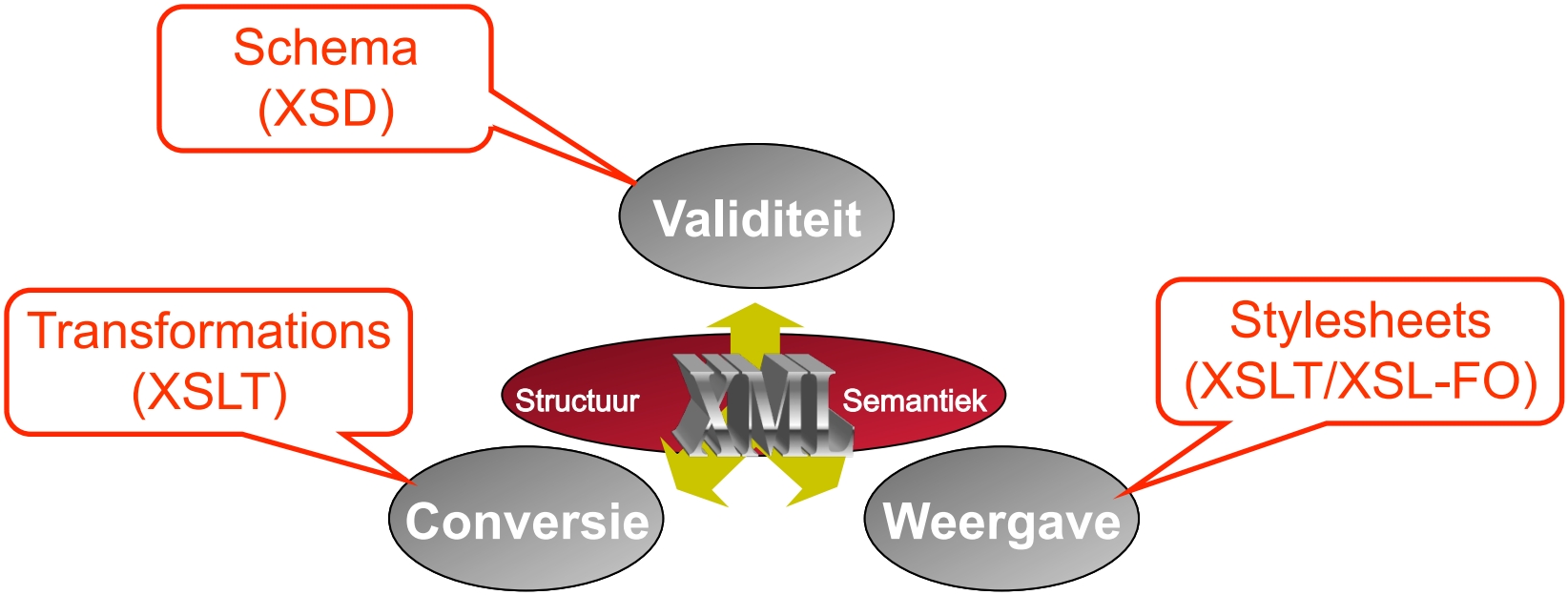
IX Einde root-template

X Begin volgende template

XI Gebruik van XSLT voor uitbreiding element-waarde

XII Ook attributen kennen templates

Conclusie



Structuur, Semantiek, Grammatica, Weergave, Transformaties (en …) worden allemaal in XML gedefinieerd!!

Info

- PDF-jes op BB!!
- http://www.ibiblio.org/xml/books/bible2/chapters/ch17.html !!
- http://www.w3.org/tr/xslt
- http://www.xml.com
- http://msdn.microsoft.com
- http://www.w3schools.com
- http://xsl.startkabel.nl/
- enz, enz

- **OEFENEN TIJDENS WORKSHOP!!**

Vragen ???