
XML

Eskema-lengoaiak

Programazio- eta prozesatze-ereduak

Xabier Arregi Iparragirre

Xabier Artola Zubillaga

Eskema kontzeptualak

- XML dokumentu-klaseak (aplikazioak) *versus* dokumentuen instantziak.
- Dokumentu-moten sintaxia definitzeko: *eskemak*.
- *Eskema-lengoaia*: eskemak adierazteko lengoaia formala.
- Eskema-prozesatzea:
 - › Balidazioa: egiaztatu ea XML dokumentuak betetzen dituen eskemaren betekizunak.
 - › Normalizazioa: dokumentua baliozkoa bada, dokumentuaren bertsio normalizatua lortzen da prozesu honetan. Behin normalizatuta: *dokumentu-instantzia*.

Eskema-lengoaiak

- *Document Type Definition (DTD)*:
 - › XML 1.0 espezifikazioan definitutako eskema-lengoia. SGMLtik dator.
 - › Zabalduea, oraingoz.
 - › Narrazioari orientatutako ikuspegia
- *W3C XML Schema*:
 - › Datuei orientatutako ikuspegia.
 - › Zehatzagoa, aberatsagoa.

DTD: *Document Type Definition*

- DTDak adierazten du:
 - › zein elementu ager daitezkeen edo jarri behar diren dokumentuetan,
 - › elementu horien atributuak,
 - › erabilitako entitateak,
 - › zein testuingurutan azaldu behar duten osagai guztiek.

DTD sinple baten adibidea

```
<!ELEMENT pertsona (izen-abizenak, lanbidea*)>
<!ELEMENT izen-abizenak (izena, abizena)>
<!ELEMENT izena (#PCDATA)>
<!ELEMENT abizena (#PCDATA)>
<!ELEMENT lanbidea (#PCDATA)>
```

elementu-erazagupena

- DTDak, gehienetan, aparteko fitxategietan gordetzen dira (*.dtd* luzapena).
 - Horrela, XML dokumentu askok erreferentzia dezakete.
- DTD bat, azken batean, elementu-erazagupenen zerrenda da.
- Erazagupenen ordenak ez du axola. Erreferentzia aurrerakoak, atzerakoak eta zirkularrak onartzen dira.

Dokumentuen baliozkotasuna

Baliozkoa (aurreko DTDarekiko):

```
<pertsona>
  <izen-abizenak>
    <izena>Andrea</izena>
    <abizena>Ormazabal</abizena>
  </izen-abizenak>
  <lanbidea>Ingeniaria</lanbidea>
  <lanbidea>Irakaslea</lanbidea>
</pertsona>
```

Honako hauek ez dira baliozkoak (aurreko DTDarekiko):

```
<pertsona>
  <izen-abizenak>
    <izena>Felipe</izena>
    <abizena>García</abizena>
    <abizena>Mendikute</abizena>
  </izen-abizenak>
  <lanbidea>Idazkaria</lanbidea>
</pertsona>
```

```
<pertsona>
  <izen-abizenak>
    <abizena>García</abizena>
  </izen-abizenak>
  <lanbidea>Idazkaria</lanbidea>
</pertsona>
```

DTDen erreferentzia XML dokumentuetatik

- Dokumentu-motaren erazagupena (*Document Type Declaration*):

```
<!DOCTYPE pertsona SYSTEM http://www.cafeconleche.org/dtds/person.dtd>
```

- Erazagupen honetan adierazten da:
 - › Zein den dokumentuaren elementu erroa: `pertsona`
 - › Non dagoen (zein URItan) DTD fitxategia.
 - Helbide erlatiboak erabil daitezke, DTDa eta XML dokumentua fitxategi-ingurune berean daudenean.

DTDen erreferentzia XML dokumentuetatik (II)

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<!DOCTYPE pertsona SYSTEM http://www.helbidea.org/dtdak/pertsona.dtd>
<?xml-stylesheet href="pertsona.css" type="text/css" ?>
<pertsona>
  <izen-abizenak>
    <izena>Pier Paolo</izena>
    <abizena>Pasolini</abizena>
  </izen-abizenak>
  <lanbidea>zine-zuzendaria</lanbidea>
  <lanbidea>idazlea</lanbidea>
</pertsona>
```

Prolog zatia

Barruko DTDak

- Adibide-dokumentu kanonikoa lantzean, ez da ideia desegokia DTDa(ren zatia) dokumentuan bertan garatzea.

```
<!DOCTYPE pertsona SYSTEM "izen-abizenak.dtd" [  
    <!ELEMENT lanbidea (#PCDATA)>  
    <!ELEMENT pertsona (izen-abizenak, lanbidea*)>  
>
```

Elementuen erazagupena

`<!ELEMENT elementu-izena edukiaren-eredua >`

- Elementu-izenak XML izen egokia izan behar du.
- Edukiaren ereduak:
 - › EMPTY: edukirik ez. `<!ELEMENT br EMPTY>`
 - › ANY: edozer eduki dezake.
 - › #PCDATA: karaktere-datuak baizik ezin ditzake eduki, umerik ez. `<!ELEMENT lanbidea (#PCDATA)>`

Elementuen erazagupena (II)

- › **Elementu umeak:** `<!ELEMENT fax (tel-zenb)>`
- › **Sekuentziak:** `<!ELEMENT izen-abizenak (izena, abizena)>`
 - Adierazitako ordenan jarri behar dira elementuak dokumentuan.
- › **Aukerak:** `<!ELEMENT argitalpena (liburua | aldizkaria)>`
- › **Eduki mistoa:** `<!ELEMENT oharra (#PCDATA | deialdia)*>`
 - Karaktere-datuekin batera, elementu umeak ere posible.
- › **Kardinalitatea:**
 - Ezer jartzen ez bada: 1
 - ?: 0 edo 1
 - +: 1 edo gehiago
 - *: 0 edo gehiago

Atributuen erazagupena

<!ATTLIST *elementu-izena atributu-izena atributu-mota balio_lehenetsia ...*>

- Elementuek dituzten atributuak definitzeko erabiltzen da.
- ATTLIST erazagupen batean, atributu bat edo gehiago defini daitezke elementu batentzat.

```
<!ATTLIST irudia iturburua CDATA #REQUIRED  
zabalera CDATA #REQUIRED  
altuera CDATA #REQUIRED  
oharra CDATA #IMPLIED  
>
```

elementua

atributuak

Atributuen motak

- CDATA: Orokorra. Edozein string.
- NMTOKEN: *XML izen* baten oso antzekoa (XML izenak ez bezala, edozein karaktere izan daiteke lehenbizikoa). Zuriuneak ezin daitezke erabili.
- NMTOKENS: Token bat edo gehiago (zuriunez bereizita). Adibidez:

```
<emanaldiak datak="2004-07-30 2004-10-08 2005-01-23"/>
```

bezalakoak adierazteko:

```
<!ATTLIST emanaldiak datak NMTOKENS #REQUIRED>
```

Atributuen motak (II)

- Enumerazioa: Atributuak onartzen dituen balioen zerrendatzea, barra bertikalaz bereizita.

```
<!ATTLIST astea eguna (astelehen | astearte | asteazken |  
ostegun | ostiral | larunbat | igande) #REQUIRED>
```

- ID: Elementuak identifikatzeko gakoa.
 - › Onartzen diren balioak XML izenak dira. Kontuz: ezin daitezke digitu batez hasi.
 - › Elementuen ID balioa ezin da errepikatu dokumentu berean.
 - › Elementuek, izatekotan, ID motako atributu bakarra izan behar dute.

```
<!ATTLIST pertsona NA ID #REQUIRED>
```

Atributuen motak (III)

- IDREF: Dokumentuko beste elementu baten IDa erreferentziatzeko.
 - › Elementuen arteko erlazioak ezartzen dira.
 - › Onartzen diren balioak XML izenak dira.
- IDREFS: ID bat baino gehiago erreferentziatu nahi direnean.
 - › Zurienez bereizitako XML izenen zerrenda da.
 - › XML izen horietako bakoitzak elementu baten IDa izan behar du.

```
<pertsona NA="_13533423"> ... </pertsona>
...
<proiektua kod="p10"> ... </proiektua>
...
<ikaslea ident="_13533423" proiekt-kod="p10 p23"/>
```

Atributuen motak (IV)

Arraroak dira honako mota hauek:

- **ENTITY**: DTDan bertan erazagutu den entitate (analizatu gabeko) bat hartzen du baliotzat.
- **ENTITIES**: Entitate bat edo gehiago hartzen ditu, zuriunez bereizita.
- **NOTATION**: DTDan bertan erazagutu den notazio bat hartzen du.

Atributuen balio lehenetsiak

Lau aukera:

- #IMPLIED
 - › Atributua hautazkoa da. Ez dago balio lehenetsirik.
- #REQUIRED
 - › Atributua derrigorrezkoa da. Ez dago balio lehenetsirik.
- #FIXED
 - › Atributuaren balioa konstantea eta aldaezina da, elementuaren instantziak atributua esplizituki jasotzen ez badu ere.
 - › Balioa komatxoaren artean adierazten da.

```
<!ATTLIST biografia xmlns:xlink CDATA #FIXED  
  "http://www.w3.org/1999/xlink">
```

- Literala
 - › Besterik gabe balio lehenetsia adierazten da, komatxoaren artean.

UEU. Iruñea, 2004ko uztaila.

UPV/EHU. Donostiako Informatika Fakultatea. Softwarea garatzeko tresna aurreratuak (2004/07).

Entitateen erazagupena

- Badira bost entitate orokor aurre-definituak.
- Gehiago ere defini daitezke.
 - <!ENTITY *entitate-izena entitate-balioa* >
 - › Barne-entitatea: <!ENTITY WWW "World Wide Web">
 - Entitate-balioa testua da (komatxoan artean).
 - › Kanpo-entitatea: <!ENTITY oinekoa SYSTEM "oinekoa.xml">
 - URI bat erreferentziatzen da, eta helbide horretako fitxategiaren edukia txertatzen da erreferentziatutako tokian (karaktere-datuetan).
 - Notazioak: <!NOTATION gif SYSTEM "image/gif">
- Entitateen erreferentzia, XML dokumentuetan (&*entitate-izena*): &oinekoa; < ...
 - › Behin definitu, eta nahi adina aldiz erabili.

Entitateen erazagupena (II)

- Zenbaitetan, elementu batzuek atributu-zerrendak eta edukien zehaztapenak partekatu egiten dituzte.
- Horrelakoetan, DTDan berridazten ibili gabe: parametro-entitateak.

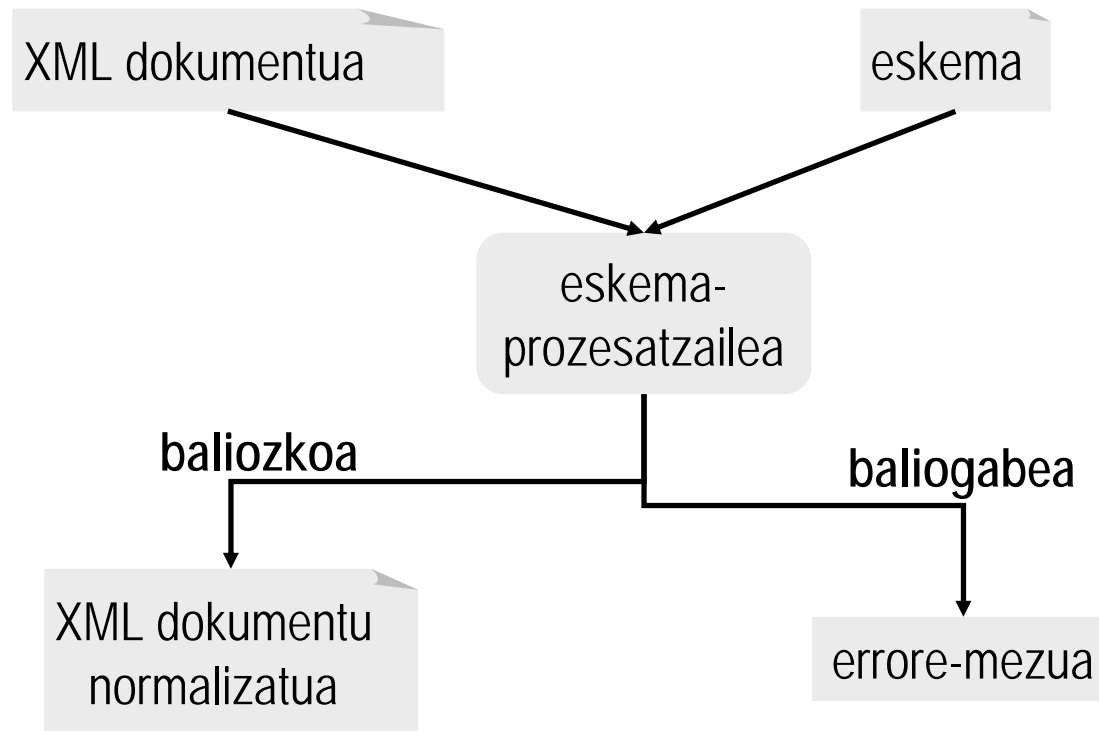
```
<!ENTITY % entitate-izena entitate-balioa >
```

- DTDaren barruan erabiltzen dira.

```
<!ENTITY % goiko_maila "( #PCDATA | irudia | paragrafoa |  
pertsona)*">  
<!ELEMENT definizioa %goiko_maila; >  
<!ELEMENT biografia %goiko_maila; >
```

Balidazioa

- Balidazioa eskemaren araberakoa da.



XML dokumentu ongi eratua *vs.* baliozkoa

- XML dokumentu bat ongi eratua (*well-formed*) bada:
 - › XMLren sintaxia betetzen du
→ dokumentua prozesagarria da.
- Baliozkoa (*valid*) bada:
 - › Dokumentua ongi eratua dago, eta, gainera, edukiek DTD jakin bateko erregelak betetzen dituzte: dokumentua ondo egituratuta dago.

DTD estandarizatuak

- Gero eta gehiago dira DTD estandarrak.
- Ez daude zentralizatuta biltegi bakar batean.
- Egokia da holakoak erabiltzea (baina erne ibili lizentzia kontuekin!)
- Ikuspegi narratibotik, aipagarriak dira:
 - › *TEI (Text Encoding Initiative)*: mota guztietako dokumentuak.
 - TEIko gida-lerroak jarraituz, nork bere DTDa era modularrean sortzeko: [Pizza Chef](#)
 - › *DocBook*: liburuak, oro har (gai teknikoak, batik bat).
 - › *RSS (Rich Site Summary)*: albisteen bildumak.
- Bi helbide:
 - › <http://www.oasis-open.org/home/index.php>
 - › <http://www.xml.org/>

TEI: *Text Encoding Initiative*

- Era askotako dokumentuak kodetzeko **estandarrik** proposatzen ditu: literatura, hiztegiak, ezaugarri-egiturak...
- Hasieran SGML aplikazioa zen; orain TEI DTDa XMLratua dago.
- Dokumentuen egitura:
 - › Elementu erroa: `<TEI . 2>`
 - `<teiHeader>`
 - `<text>`

TEI dokumentua: burukoa, meta-informazioa

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xslt" href="../../frameworks/tei/xml/tei/stylesheet/html/tei.xsl"?>
<?oxygen RNGSchema="http://www.tei-c.org/release/xml/tei/custom/schema/relaxng/teilight.rng" type="xml"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader type="text">
    <fileDesc>
      <titleStmt>
        <title type="main">Writing TEI XML Documents</title>
        <title type="sub">How to create a TEI XML document and convert it to PDF within
          the &lt;oxygen&gt; XML Editor</title>
        <author>syncRO soft Ltd.</author>
      </titleStmt>
      <editionStmt>
        ...
      </editionStmt>
      <publicationStmt>
        ...
      </publicationStmt>
      ...
    </fileDesc>
    <encodingDesc>
      ...
    </encodingDesc>
    <revisionDesc>
      <change>
        <date>2003-09-15</date>
        <respStmt>
          <name>syncRO soft staff</name>
          <resp>edt (Editor)</resp>
        </respStmt>
        <item>Created the initial version of the article</item>
      </change>
      ...
    </revisionDesc>
  </teiHeader>
  ...

```


TEI dokumentua: testua, informazioa

```
...
<text>
  <front>
    <titlePage>
      <docTitle>
        <titlePart type="sub">How to create a <hi rend="bold">TEI</hi> XML document and convert it to
          <hi rend="bold">PDF</hi> within the <hi rend="bold">&lt;oxygen/&gt; XML Editor</hi>
        </titlePart>
      </docTitle>
      <docAuthor>
        <name>syncRO soft</name>
      </docAuthor>
      <docDate>September 15, 2003</docDate>
    </titlePage>
  </front>
  <body>
    <divGen type="toc"/>
    <div type="abstract" xml:id="S0" rend="bold">
      <head rend="bold">Abstract</head>
      <p rend="bold">
        This article is intended...
      </p>
    </div>
    <div type="sect" xml:id="S1">
      <head>Creating an XML Document in &lt;oxygen/&gt; Using the TEI Lite DTD</head>
      <p>
        From the &quot;<hi rend="bold">File</hi>&quot; menu choose &quot;<hi rend="bold">New from
        Templates ...</hi>&quot;. It will be displayed a window, allowing you to choose from a list of built-in
        document templates as starting points for your TEI XML documents:
      </p>
    </div>
  </body>
  ...
</text>
```

DocBook

- XML aplikazioa da (hasieran SGML, gero XML).
- Testu eta dokumentu teknikoetarako dago pentsatua, esate baterako, software dokumentaziorako (*Linux Documentation Project*-eko corpus gehiena *DocBook*-en idatzi da).
- Software irekia da.
- Behin *DocBook* bertsioa edukita, bukaerako bertsioa lortzeko beste formatu batera pasa behar da.

DocBookx dokumentua

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE book SYSTEM "../..//docbook/dtd/docbookx.dtd">
<book>
  <article>
    <title>Welcome to the Docbook</title>
    <para>This is a sample showing &lt;i>oxygen</i></para>
    <para>In order to preview </para>
    <itemizedlist>
      <listitem>
        <para>
          <link id="a" linkend="a">www.xml.com</link>
        </para>
      </listitem>
      <listitem>
        <para>
          <link id="c" linkend="c">www.w3c.org</link>
        </para>
      </listitem>
    </itemizedlist>
  </article>
</book>
```

DTDen gabeziak

1. Ez du XML syntaxirik erabiltzen.
2. Ezin zaie murriztapenik ezarri karaktere-datuei.
3. Atributuen balio-ereduak sinpleegiak dira.
4. Karaktere-datuak eta espresio erregularrak ezin dira konbinatu.
5. Izen-espaziorik ez du ametitzen.
6. Modulartasun eta berrerabilgarritasun eskasa.
7. Eskemen bilakaera, hedapen edo herentziarako aukerarik ez.
8. Dokumentatzeko eredu propiorik ez.

DTDen gabeziak (II)

9. Ezin adierazi testuinguruaren menpeko eduki- edo atributu-erazagupenik.
10. ID atributuen mekanismoa sinpleegia da: ID motarik ez.
11. Elementuetarako balio lehenetsirik ez (soilik atribuetarako).
12. "Edozein elementu" edo "edozein atributu" adierazterik ez.
13. Ezin da esan elementu baten zenbat agerpen onartzen diren.

Balorazioa

- Dokumentu narratiboetarako DTDen kontrol-maila nahikoa bide da.
- Kontua da, ordea, XML gero eta gehiago erabiltzen dela datu-aplikazioetarako (adibidez, SOAP web zerbitzuetaarako), eta horrelakoetarako kontrol zehatzagoa eta zorrotzagoa behar izaten da.
 - › Besteak beste, *izen-espazioak* erabiltzea komenigarria da datu-transferentzietarako.

Izen-espazioak

- Izen-espazioak izen bereko elementuak desanbiguatzeko erabiltzen dira.
- Izan ere, hainbat dokumentutan XML aplikazio bat baino gehiago konbinatzen dira, eta etiketen artean talkak gertatzen dira.
 - › Adibidez, XHTML dokumentu batean SVG irudiak eta MathML ekuazioak ager daitezke. Ez da harritzekoa halakoetan, izen berdineko elementuak anbiguoak izatea. Esate baterako, set elementua SVGkoa eta MathMLkoa da.

Nola desanbiguatu izen-espazioaren arabera

- Elementuei eta atributuei *izen-espazio URIa* (Namespace URI) esleitzen zaie. Eta, horrela, URI desberdina duten elementuak desberdinak dira.
- Izen-espazio URIa: URI formatua duen izen sinbolikoa, izen-espazioa unibokoki identifikatuko duena.

Izen-espazioen esleipena

svg elementuari *izen-espazio URI* bat esleitu zaio,
xmlns atributuaren bidez

Besterik adierazi ezean,
svg-ren elementu ume
guztiak izen-espazio
horretakoak dira.

```
<svg xmlns=http://www.w3.org/2000/svg  
width="12cm" height="10cm">  
  <ellipse rx="110" ry="130" />  
  <rect x="4cm" y="1cm" width="3cm" height="6cm" />  
</svg>
```

Aurritzkiak

- Elementuak zein izen-espaziotakoak diren esplizituki adierazi nahi denean, izen-espazio horiei aurritzkiak atxikitzen zaizkie.
 - › Aurritzkien erabilera oso zabaldua da.

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="pertsona">
    <p> Pertsona bat </p>
  </xsl:template>
</xsl:stylesheet>
```

aurritzkiaren erabilpena

aurritzkiaren definizioa

Adibidea

```
<?xml version="1.0"?>
<!-- bi izen-espazioren aurrizkiak definitu dira -->
<bk:book xmlns:bk='urn:loc.gov:books'
          xmlns:isbn='urn:ISBN:0-395-36341-6'>
  <bk:title>XML in a nutshell</bk:title>
  <isbn:number>1568491379</isbn:number>
</bk:book>
```

XML Schema

- Diseinuko oinarriak eta ezaugarriak:
 1. XMLz idazten dira.
 - › Bada *Schema* bat *Schema*-etarako!
 2. DTDak baino adierazmen handiagoa:
 - › Egitura eta datuak murrizteko mekanismoak.
 - › Izen-espazioak.
 - › Datu-mota aurredefinituak.
 - › Objektuen orientazioa mota-sisteman.
 - › Modulartasuna: eskemak barneratzeko eta birdefinitzeko aukera.
 - › Kardinalitate-murriztapenak.
 - › ID/IDREF atributuen barrutia edo domeinua zehaztea dago.
 - › Karaktere-datu eta atributu onargarriak zehazteko, espresio erregularrak.

XML Schema: adibidea

helbide-eskema.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema'>
  <xs:element name="izena" type="xs:string"></xs:element>
</xs:schema>
```

helbidedok.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<izena>
  Pello Osinalde
</izena>
```

helbidedok.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<izena xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="helbide-eskema.xsd">
  Pello Osinalde
</izena>
```

- Elementuei lotzen zaie eskema.
- Elementu erroari lotuz gero, dokumentu osoari dagokio eskema.

Elementuen erazapupena

- Elementuak erazagutzeko `<xs:element>` elementua erabiltzen da.
- Atributu asko har ditzake. Beste batzuen artean, hauek (denak dira hautazkoak):
 - > `name`: elementuaren izena
 - > `type`: elementuaren mota. Sinplea edo konplexua izan daiteke.
 - > `id`: ID motako atributu baten izena, bakarra.
 - > `maxOccurs`: zenbat aldiz ager daitekeen gehienez.
 - > `minOccurs`: zenbat aldiz ager daitekeen gutxienez.

Elementuak, atributuak eta motak konbinatuz. Adibidea.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema'>
  <xs:element name="bidaltzeAgindua" type="bidaltzeMota"/>
  <xs:complexType name="bidaltzeMota">
    <xs:sequence>
      <xs:element name="nora" type="xs:normalizedString"/>
      <xs:element name="fakturaNori" type="xs:normalizedString"/>
      <xs:element name="oharrak" minOccurs="0"/>
      <xs:element name="paketeak" type="xs:token"/>
    </xs:sequence>
    <xs:attribute name="aginduData" type="xs:date"/>
  </xs:complexType>
</xs:schema>
```

Mota sinpleak

- Oinarrizko datu-moten parekoak.
- Elementu sinpleei dagozkie (testu soila, umerik ez).
- Ezin dute elementu habiaturik eduki :
 1. **xs:anyURI**
 2. **xs:boolean** (true false, edo 0 1)
 3. **xs:dateTime** (data eta denbora balioen konbinazioa. Adb: 2004-07-20T16:15:00-05:00)
 4. **xs:duration** (denbora iraupena; urte, hilabete, egun, ordu... unitatetan)
 5. ID, IDREF, IDREFS, ENTITY, ENTITIES, NOTATION, NMTOKEN, NMTOKENS (XML 1.0 erazapupenekoak)
 6. **xs:integer**
 7. **xs:language** (XML 1.0ko xml:lang atributuaren balioak)
 8. **xs>Name** (XML izen bat)
 9. **xs:string** (Unicode string-a)
 10. **xs:normalizedString** (string-a. Tabulazioen eta lerro-jauzien orde zuriuneak)
 11. **xs:token** (normalizedString bezala, baina zuriune gehiegizkoak kenduta)

Mota konplexuak

- Elementu habiatuak dituzten elementuek mota konplexua dute.
- `<xs:complexType>` elementua erabiltzen da.
 - › Izendatuz gero, motek berrerabiltzeko aukera ematen dute, elementuaren `type` atributuan mota-izen hori jarritz.
 - › `<xs:complexType>` etiketa elementu baten barruan jartzen denean, elementu horren mota da (ez da `type` atributua erabiltzen).
- Mota konplexuen baitan, edukien egitura zehazten da.

Edukiak

- Hutsa: elementu hutsa, edukirik gabea. Adibidez:

```
<xs:element name="telefonoa" minOccurs="0">  
  <xs:complexType>  
    <xs:attribute name="zenbakia" type="xs:string"/>  
  </xs:complexType>  
</xs:element>
```

- Edozer: `<xs:any>`
- Eduki konplexua: elementuak besterik ez daudenean.
- Eduki sinplea: testua, elementu umerik ez.

Edukiak (II)

- Eduki nahasia: testua eta elementu umeak nahasturik. `mixed` atributuak `true` balioa hartzen du.
- Elementuen kokapen-ordena:
 - > `<xs:sequence>`: dauzkan elementuek adierazitako ordenan agertu behar dute.
 - > `<xs:choice>`: aukera batzuetatik bat eta bakarra onartzen du.
 - > `<xs:all>`: dauzkan elementu guztiek behin agertu behar dute, baina edozein ordenatan ager daitezke.

Eduki mistoa. Adibidea

```
<xs:element name="gutuna">
  <xs:complexType mixed="true">
    <xs:sequence>
      <xs:element name="agurra">
        <xs:complexType mixed="true">
          <xs:choice>
            <xs:element name="kaixo" fixed="Kaixo"/>
            <xs:element name="agur"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
      <xs:element name="gorputza">
        <xs:complexType mixed="true">
          <xs:all>
            <xs:element name="noiz" type="xs:date"/>
            <xs:element name="gaia" type="xs:string"/>
          </xs:all>
        </xs:complexType>
      </xs:element>
      <xs:element name="adioa" fixed="Ondo izan"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Atributuak

- Atributuak erazagutzeko `<xs:attribute>` elementua erabiltzen da.

Atributuen
definizioa XMLz

```
<xs:attribute
  default = string
  fixed = string
  form = (qualified | unqualified)
  id = ID
  name = NCName
  ref = QName
  type = QName
  use = (optional | prohibited | required) : optional>
</attribute>
```

Adibidea

```
<xs:attribute name="adina" type="xs:positiveInteger" use="required"/>
```

Erabiltzaileak sortutako eta eratorritako motak

- Ikusi dugu mota konplexuak sor daitezkeela, baina mota simple berriak ere sor daitezke.
- Mota sinpleetatik mota berriak eratortzeko, `<xs:simpleType>` elementuaren baitan hiru bide daude:
 - > `<xs:restriction>`: oinarrizko motaren domeinua murriztuz. *Fazetak*.
 - > `<xs:list>`: oinarrizko motaren balio sekuentzia onartuz, zuriunez bereizita.
 - > `<xs:union>`: oinarrizko motak bilduz.

Erabiltzaileak sortutako eta eratorritako motak (II)

```
<xs:element name="jeans_size">
  <xs:simpleType>
    <xs:union memberTypes="sizebyno sizebystring"/>
  </xs:simpleType>
</xs:element>
```

```
<xs:simpleType name="sizebyno">
  <xs:restriction base="xs:positiveInteger">
    <xs:maxInclusive value="42"/>
  </xs:restriction>
</xs:simpleType>
```

```
<xs:simpleType name="sizebystring">
  <xs:restriction base="xs:string">
    <xs:enumeration value="small"/>
    <xs:enumeration value="medium"/>
    <xs:enumeration value="large"/>
  </xs:restriction>
</xs:simpleType>
```

Erabiltzaileak sortutako eta eratorritako motak (III)

- Mota konplexu eratorriak sortzeko, `<xs:complexContent>` etiketaren barruan bi aukera daude:
 - > `<xs:restriction>`: oinarri-motaren azpimultzoa denean.
 - > `<xs:extension>`: oinarri-motaren edukietan oinarritzen da, baina hedatu egiten da elementu berriekin. Adibidez:

```
<xs:complexType name="mailingAddressType">
  <xs:complexContent>
    <xs:extension base="addr:physicalAddressType">
      <xs:sequence>
        <xs:element name="zipCode" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```


Murriztapenak mota sinpleetan. *Fazetak.*

Oinarri-motari ezarritako murriztapenei deitzen zaie *fazeta*.

Honako fazeta motak daude:

1. **xs:length** (edo **xs:minLength** eta **xs:maxLength**): string baten luzera, adibidez. Zehatza, gutxienekoa, gehienezkoa.
2. **xs:pattern**: string-ak patroi batekin (espresio erregularra) parekatzen dira.
3. **xs:enumeration**
4. **xs:whiteSpace**: zuriuneen kudeaketarako gidalerroa. Hiru balio: `preserve` (daudenean utzi), `replace` (tabulazioen eta lerro-jauzien ordez zuriunea jarri) eta `collapse` (aurrekoa bezala, baina beti zuriune bakarra utziz).
5. **xs:maxInclusive** eta **xs:maxExclusive**: zenbakizko moten barrutiak.
6. **xs:minInclusive** eta **xs:minExclusive**: zenbakizko moten barrutiak.
7. **xs:totalDigits**: zenbakien digitu kopurua (parte osokoak gehi hamartarrak).
8. **xs:fractionDigits**: zenbakien digitu hamartarren kopurua.

Fazetak. Adibideak.

```
<xs:simpleType name="poliza-zk">  
  <xs:restriction base="xs:string">  
    <xs:pattern value="\d\d-\d-\d\d\d\d" />  
  </xs:restriction>  
</xs:simpleType>
```

```
<xs:simpleType name="izena">  
  <xs:restriction base="xs:string">  
    <xs:maxLength value="50" />  
  </xs:restriction>  
</xs:simpleType>
```

Balio unibokoak eta gakoak.

- Dokumentu bateko elementuen eta atributuen balioen arteko erlazioak modu sofistikatuagoan adieraz daitezke. Horretarako dauden aukerak:
 - > `<xs:unique>`: Elementuaren edo atributuaren balioa bakarra izatera behartzen du.
 - > `<xs:key>`: Elementu honen bitartez gako elementua markatzen da, gero `<xs:keyref>` motako elementua erabiliz erreferentziatzen da.

Izen-espazioen erabilera

helbidea izen-espazioa sortu da eskeman

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema' targetNamespace="helbidea">
  <xs:element name="izena" type="xs:string"></xs:element>
</xs:schema>
```

helbidea izen-
espazio hori atxiki
zaio dokumentu-
instantziari.

```
<?xml version="1.0" encoding="UTF-8"?>
<izena xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xsi:schemaLocation="helbidea helbide-eskema-target.xsd" xmlns="helbidea">
  Pello Osinalde
</izena>
```

Zein eskema-lengoaia aukeratu?

- Izaera narratiboa duten dokumentuetarako egokiak izan daitezke DTDak.
- DTDen mugak gainditzeko, *XML Schema* da W3C kontsortzioak bultzatzen duena.
- Hala ere, XML komunitatean gero eta gehiago hedatzen ari dira beste lengoaia batzuk: *RELAX NG*, *Schematron*...
- DSDL (<http://dsdl.org/>): ISO estandar gisa lantzen ari den ekimen bateratua, eskema-lengoaia hauetako batzuen bilgune.

Bestelakoak:

- Programazio-ereduak
- *XInclude*
- *RDDL*

Programazio-ereduak

- Asko dira XML prozesadoreak.
Programazio-lengoia gehienek eskaintzen dute XML dokumentuak prozesatzeko euskarria.
- Prozesadore horiek bi programazio-ereduren inplementazioak izan ohi dira:
 - › *DOM (Document Object Model)*
 - › *SAX (Simple API for XML)*

DOM

- XML dokumentuak zuhaitz motako egituratzat hartzen dira.
- Objektuetan oinarritzen da. Dokumentuen ikuspegi abstraktua lantzen da.
- Dokumentu osoa memorian kargatu, eta gero oso-osorik prozesatzen da.
- Dokumentu txikietarako egokia bada ere, bolumen handiko lanak egiteko astunegia da.
- Adibidez, Node interfazearen metodoak dira hauek:
 - `hasAttributes`
 - `newChild`
 - `replaceChild`

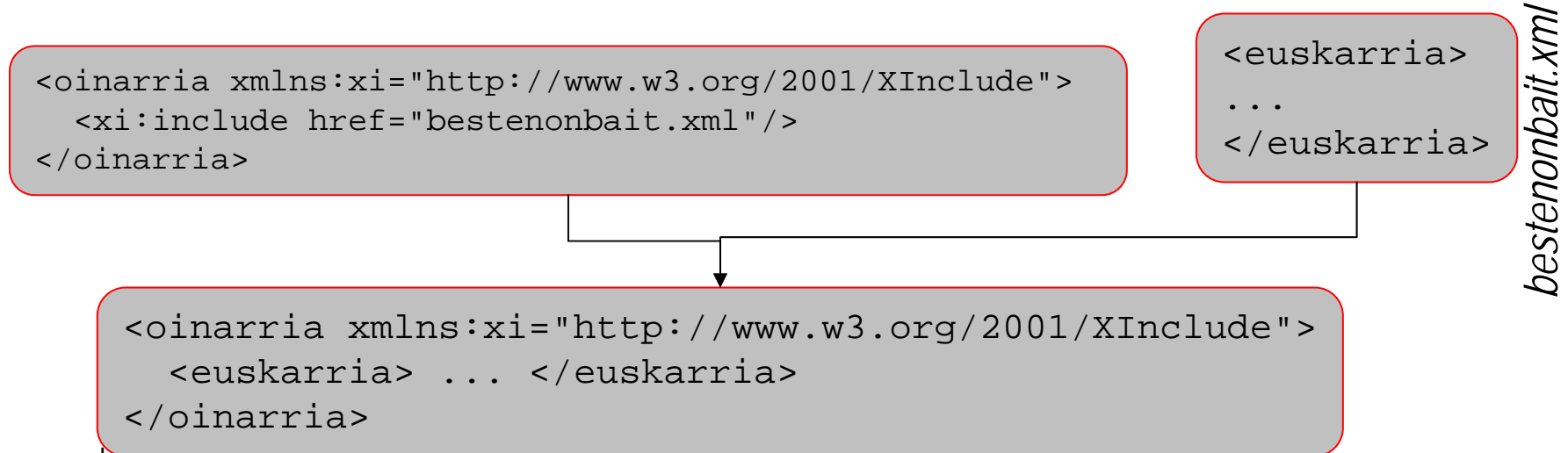
SAX

- Gertaeretan oinarritutako eredua.
- Dokumentua hasieratik bukaerara irakurtzen da. Irakurri ahala trataera jakin batzuk egiten dira.
- XML dokumentu bat hartuz gero, SAXen oinarritutako analizatzaileak horrelako metodoei deituko lieke:

```
startDocument()  
processingInstruction(String target, String data)  
startPrefixMapping(...)  
startElement(...)  
...  
endElement(...)
```

XInclude: XML dokumentuak konbinatzeko

- Berrerabilpena eta modulartasuna hobetzeko, teknika komenigarria da dauden dokumentuak konbinatuz berriak sortzea.
- Horretarako mekanismoa da *XInclude*. Adibidez:



U.E.U. iruzkia, 2004ko aziala.

UPV/EHU. Donostiako Informatika Fakultatea. Softwarea garatzeko tresna aurreratuak (2004/07).

RDDL: baliabideak deskribatzeko.

- *RDDL (Resource Directory Description Language)*, XML aplikazio bat da, XML aplikazioak deskribatzeko.
- Elementu nagusia `<rddl:resource>` da.
- Elementu horren pean deskribatzen da baliabidea, XHTML oinarrizkoa erabiliz horretarako.
- Baliabideak izaera estandar batzuen arabera sailkatzen dira, eta zein URItan aurki daitezkeen adierazi.

Helbide batzuk

- XML (W3C): <http://www.w3.org/XML>
- *XML-dev mailing list*:
<http://www.xml.org/xml/xmldev.shtml>
- Programatzeko baliabideak (liburutegiak, paketeak...):
<http://www.xml.com/resourceguide/>
- XML Schema-ren espezifikazioa:
<http://www.w3.org/TR/xmlschema-1/>
- Tutorial bat:
<http://www.w3schools.com/xml>
- Doako XML tresnak:
<http://www.garshol.priv.no/download/xmltools>
- Online erreferentzia:
<http://www.oasis-open.org/cover>
- TEI: <http://www.tei-c.org>
- DocBook: <http://www.docbook.org>