XML_Schema_teoria_1.pdf
ELEC-C8203 - Automaatiojärjestelmät 2
XML Tietointegraatio

XML Schema – Teoria

1. Johdanto

XML Schema on kieli, jolla määritellään muita XML-pohjaisia kieliä. XML Schema -määrittelyillä luodaan kielen rakenteet

- määritellään käytettävissä olevat elementit ja attribuutit
- asetetaan rajoituksia elementtien sisäkkäisyydelle ja peräkkäisyydelle
- määritetään attribuuttien arvoilla tyypit ja mahdolliset raja-arvot
- lisätään attribuuteille oletusarvot tai todetaan tietyt attribuutit pakollisiksi

Kielen rakenteen pohjalta syntyy dokumentin sisältö ja semantiikka. XML-dokumenttien kielenmukaisuus voidaan validoida kielen schemaa vasten. Määritelty rakenne mahdollistaa dokumenttien tarkistuksen ja helpottaa dokumenttien koneellista käsittelyä.

XML dokumentti 09-08.xml

```
<?xml version="1.0"?>
<wonder xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="09-06.xsd" >
  <name>Colossus of Rhodes</name>
  <location>Greece</location>
  <height>107</height>
</wonder>
```

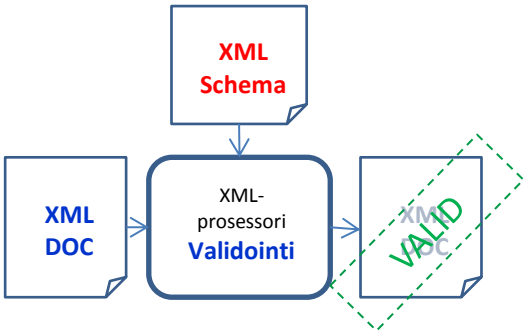
XML Schema dokumentti 09-06.xsd

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="wonder">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="location" type="xs:string"/>
      <xs:element name="height" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

Kuva 1. Esimerkki yksinkertaisesta XML-dokumentista ja sen rakenteen määrittelevästä XML-Schemasta.

2. XML-dokumentin validointi

Validoinnissa verrataan XML-dokumenttia sen XML Schemaan. Dokumentin jokainen elementti, attribuutti ja tekstikenttä tarkastetaan. Mikäli Schema ei salli jotain komponenttia tai jokin vaadittu komponentti puuttuu, dokumentti ei ole Scheman mukainen ja validointi epäonnistuu. Validoinnilla sovellus voi varmistaa, että sisään luettu dokumentti on ainakin syntaktisesti oikein. Validointi ei kuitenkaan löydä semanttisia tai loogisia virheitä.
HUOM: Validi dokumentti ei tarkoita samaa kuin oikeamuotoisuus (well formed). Kaikkien XML-dokumenttien tulee olla oikeamuotoisia. Kun taas validi XML-dokumentti on jonkin Scheman määrittelemän kieliopin mukainen.



Kuva 2. Kaaviokuva XML-dokumentin validoinnista. ’XML prosessori’ on kirjasto-ohjelma (XML API).

3. Elementtien tietotyypit

XML Schema määrittää kaksi tietotyyppikategoriaa. XML-dokumentin elementit voivat olla joko yksinkertaisia tai monitahoisia:

Yksinkertainen (simpleType)

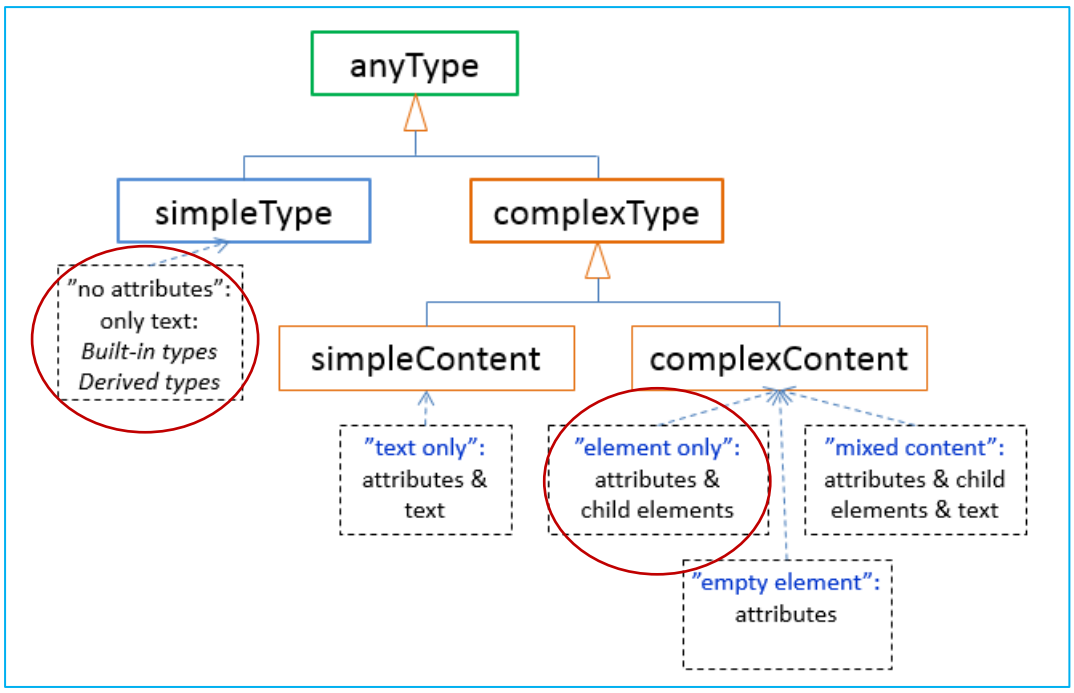
- attribuuttien arvot
- tekstikentät elementeissä
- valmiiksi määritellyt (built-in) tyypit
- käyttäjän omat johdetut tyypit (derived custom simple types)

Monitahoinen (complexType)

- lapsielementtejä ja/tai attribuutteja sisältävät elementit
- ei ole valmiiksi määriteltyjä tyyppejä eli kaikki monitahoiset tyypit täytyy johtaa

Yksinkertaista tyyppiä (simpleType) oleva elementti voi sisältää vain tekstiä mutta ei attribuutteja.
Monitahoista tyyppiä (complexType) oleva elementti voi sisältää:

- Attribuutteja ja tekstiä
- Attribuutteja ja lapsielementtejä
- Attribuutteja
- Attribuutteja ja lapsielementtejä ja tekstiä
- Ei mitään



Kuva 3. Elementtien tyyppihierarkia. Tässä dokumentissa esitellään vain ympyrällä merkatut tyypit.

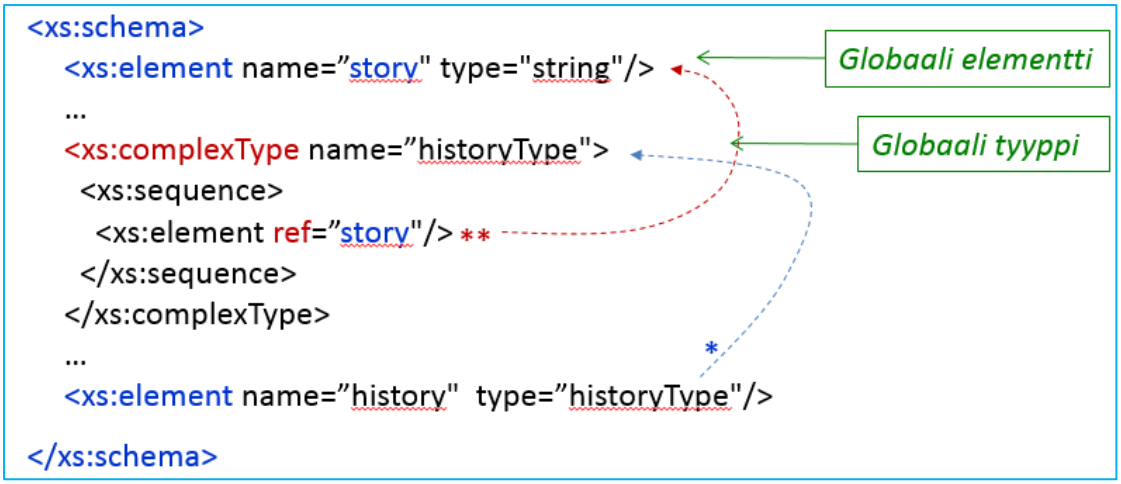
4. Elementtien määrittely

Elementit määritellään scheman <xs:element>-elementillä. Esimerkiksi XML-dokumentin elementti: <document> voidaan määritellä seuraavasti: *<xs:element name="document"/>*

attribuuteilla voidaan lisätä määreitä elementtiin, esimerkiksi säätää elementin sisältö pelkäksi tekstiksi: *<xs:element name="document" type="string"/>*

Paikallinen ja globaali tyyppi

Elementin tyyppi/sisältö voidaan määrittää joko paikallisesti (local type) tai käyttäen globaaleja tyyppejä (global type). Paikallisessa määrittelyssä <xs:element> sisältää lapsenaan joko <xs:complexType> tai <xs:simpleType> elementin. Globaalissa määrittelyssä viitataan valmiiksi määriteltyyn nimettyyn globaaliin tyyppiin, jolloin <xs:complexType> on suoraan <xs:schema> elementin lapsi.
Esimerkiksi Kuvassa 1 *'wonder'* elementin tyyppi on määritelty lokaalista, kun taas Kuvassa 4. elementin *'history'* tyyppi on määritelty globaalisti nimellä *'historyType'*.



Kuva 4. Esimerkki globaalin tyypin ja globaalin elementin määrittelystä ja niihin viittaamisesta (*, **)

Elementit tyyppiä simpleType

Yksinkertaista tyyppiä (Simple type) oleva elementti voi sisältää vain tekstiä (numeroita, kirjaimia, erikoismerkkejä, unicode-merkkejä), mutta ei attribuutteja. Tekstisisällön tyyppi voi olla esimerkiksi:

- *String; Integer; Boolean; Date*

Tai mitä tahansa muuta XML-Scheman *built-in* tyyppiä (ks Kuva 5) tai built-in tyypeistä johdettuja tyyppejä tai yksinkertaisia tietorakenteita kuten:

- *Range; Enumeration; List; Regex pattern*

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="simple_types">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="height" type="xs:string"/>
                <xs:element name="year_built" type="xs:integer"/>
                <xs:element name="birth" type="xs:date"/>
                <xs:element name="time_painted" type="xs:time"/>
                <xs:element name="when_shot" type="xs:dateTime"/>
                <xs:element name="strike_length" type="xs:duration"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

```
<?xml version="1.0"?>
<simple_types xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                    xsi:noNamespaceSchemaLocation="10-07.xsd">

    <height>39</height>
    <year_built>1280</year_built>
    <birth>1879-03-14</birth>
    <time_painted>21:08:00</time_painted>
    <when_shot>1968-04-04T18:01:00-05:00</when_shot>
    <strike_length>P5D</strike_length>

</simple_types>
```

-05:00= - 5 hour offset from UTC

Kuva 5. *complexType* määrittely, joka sisältää kuuden *simpleType* elementin määrittelyn.

Elementit tyyppiä complexType

Monitahoinen (complexType) elementti voi sisältää lapsielementtejä ja/tai attribuutteja. Kaikki monitahoiset tyypit täytyy määritellä schemassa.
Monitahoiset elementit jaetaan edelleen kahteen alaryhmään (ks. kuva 3)

- Yksinkertainen tietosisältö (simple content): vain teksti ja attribuutit sallittu (ks. XML_Schema_extra.pdf)
- Monitahoinen tietosisältö (complex content): myös lapsielementit sallittu

HUOM: complexContent on oletussisältötyyppi, joten tätä alaryhmää ei tarvitse aina eksplisiittisesti ilmoittaa. (Oletustyyppi on xs:anyType tyypistä rajoittamalla johdettu xs:complexContent tietotyyppi. ks. tarkemmin _extra.pdf).
Kuvassa 5 on esitetty esimerkki *complexType* oletustietotyyppiä olevan elementin määrittelystä. Huomaa, että *xs:complexType* elementin lapsena on *xs*:*sequence* elementti, joka on yksi ns. sisältömalleista

Sisältömallit

Elementtejä sisältävän monitahoisen tyypin rakenne määritellään sisältömallilla (content model): Rakenne ja lapsielementtien järjestys määritellään malliryhmien avulla (model group)

1. Elementtien järjestetty sekvenssi (sequence)
2. Elementtien järjestämätön lista (all)
3. Elementtien vaihtoehtoiset valinnat (choise)

Sequence – järjestetty lista

Sequence-elementti määrittelee tyypin lapsielementtien järjestetyn listan. Sekvenssissä määriteltyjen lapsielementtien tulee esiintyä validissa XML-instanssidokumentissa sekvenssi-määrittelyn mukaisessa järjestyksessä.

- Sekvenssin esiintymiskerrat määritellään attribuuteilla minOccurs ja maxOccurs
- Tietotyypin sekvenssi-malliryhmään kuuluvat lapsielementit määritellään xs:element-elementillä
- name-attribuutin arvo määrää lapsielementin nimen
- lapsielementin mahdollinen lukumäärä määritellään minOccurs ja maxOccurs attribuuteilla (molempien oletusarvo on yksi 1)
- Sequence-malliryhmä voi sisältää toisia sequence- ja/tai choise-malliryhmiä

All – järjestämätön lista

All-elementti määrittelee tyypin lapsielementtien järjestämättömän listan eli määritellyt lapsielementit voivat esiintyä validissa XML-instanssidokumentissa missä tahansa järjestyksessä.

- All-malliryhmän sekä lapsielementtien esiintymiskerrat määritellään attribuuteilla minOccurs ja maxOccurs, mutta niiden arvot voivat olla vain nolla (0) tai yksi (1)

Choise – vaihtoehtoiset elementit

Choise-elementti määrittelee tyypin lapsielementtien mahdolliset vaihtoehtoiset lapsielementti-valinnat. Vain yksi Choice-malliryhmän lapsielementeistä saa esiintyä validissa XML-instanssidokumentissa, mutta sen lukumäärä voidaan asettaa kardinaliteetti-attribuuteilla.

- Choise-malliryhmän sekä lapsielementtien esiintymiskerrat määritellään attribuuteilla minOccurs ja maxOccurs

- Choise-malliryhmä voi sisältää sequence- ja/tai choise-malliryhmiä

```
...
<xs:complexType name="wonderType">
    <xs:sequence>
        <xs:element name="name" type="nameType"/>

        <xs:choice>
            <xs:element name="location" type="xs:string"/>
            <xs:sequence>
                <xs:element name="city" type="xs:string"/>
                <xs:element name="country" type="xs:string"/>
            </xs:sequence>
        </xs:choice>

        <xs:element name="height" type="heightType"/>
        <xs:element name="history" type="historyType"/>
        <xs:element name="main_image" type="imageType"/>
        <xs:element name="source" type="sourceType"/>
    </xs:sequence>
</xs:complexType>
...
```

Kuva 6. Esimerkki sisäkkäisten sequence ja choise malliryhmien käytöstä. Validissa XML-dokumentissa ’wonder’:n sijainnin voi ilmoittaa joko ’location’-elementillä tai ’city ’ja ’country’ elementtien yhdistelmällä.

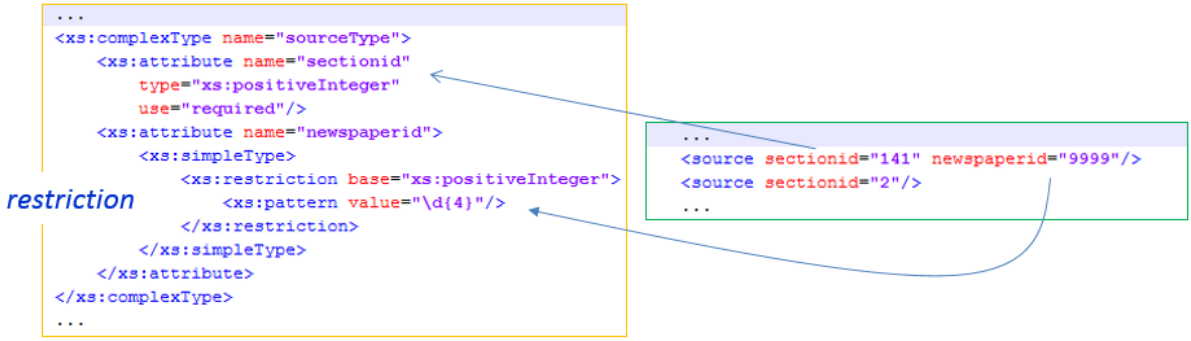
5. Attribuuttien määrittely

Kaikki elementit, joilla on attribuutteja kuluvat, johonkin neljästä monitahoisen tyypin (complex type) ryhmästä (ks. kuva 3). Attribuutit itsessään ovat aina yksitahoista tyyppiä (simple type). Attribuutit määritellään elementillä <xs:attribute> ja sen name-attribuutin arvo määrää attribuutin nimen. Attribuutin tyyppi määritellään joko type-attribuutin arvolla tai rajoittamalla (restriction) tai laajentamalla (extension) jotain base-tyyppiä.

```
<xs:complexType name="actuator_type">
    <xs:sequence>
        <xs:element ref="name" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="description" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="ID" type="xs:string"/>
    <xs:attribute name="type" type="xs:string"/>
</xs:complexType>
```

Kuva 7. Esimerkki kahden xs:string tyyppisen attribuutin määrittelystä

XML-Schema mahdollistaa hyvin tarkan attribuuttien sallittujen arvojen ja esitystapojen määrittelyn (mikä ei ole mahdollista elementtien tekstisisällön suhteen). Tämä kannattaa ottaa huomioon päätettäessä esitetäänkö jokin tieto elementtien vai attribuuttien avulla. Seuraavassa kuvassa 8 on esimerkki tiukemmasta attribuuttien arvoalueiden määrittelystä (Ks. Kuva 8).



Kuva 8. Esimerkki kahden xs:positiveInteger tyyppisen attribuutin määrittelystä, jossa *'newspaperid'* attribuutin arvo määrätään esitettäväksi neljällä numerolla (regular expression \d(4))