

Luento 14: Kertaus

AS-0.110 XML-kuvauskielten perusteet

Janne Kalliola

Kertaus

- XML
 - nimiavaruus
 - validointi
- XML:n käyttö
 - tietorakenteiden kuvaus
 - ohjelmointi
- XML-pohjaiset kielet
 - peruskielet
 - muut kurssilla käsitellyt kielet
- Metatieto

- Merkintäkieli on tekstipohjainen kieli, jolla voidaan kuvata rikasta joukkoa tietoa hyvin yksinkertaisen säännösten pohjalta
 - Yleensä merkintäkielen avulla luodaan tiedostoon haluttu rakenne
- Extensible Markup Language (XML) on World Wide Web Consortiumin (W3C) suositus elektronisen tiedon esitysmuodoksi
 - XML määrittelee ainoastaan tavan esittää tietoa
- XML on merkintä- ja metakieli
 - XML:lla voidaan määritellä uusia kieliä, joilla vasta kuvataan itse tietoa
 - XML on kirjainherkkä kieli (case sensitive)

- XML-dokumentti rakentuu peräkkäisistä ja sisäkkäisistä elementeistä
 - elementit merkitään kulmasuluilla:
`<elementti>...</elementti>`
 - elementillä on alku- ja loppumerkintä (start & end tag)
 - elementti voi olla myös tyhjä, jolloin se voidaan merkitä lyhennetyksi: `<elementti/>`
- Elementin alkumerkintään voidaan lisätä attribuutteja (attribute)
 - attribuutti on avain-arvopari, joka täsmentää elementtiä
 - attribuutti merkitään `avain="arvo"` tai `avain='arvo'`

XML-dokumentin rakenne (2/2)

- Elementtien välissä voi olla leipätekstiä tai toisia elementtejä
- XML-dokumentin täytyy kokonaisuudessaan olla yhden elementin (juurielementti, root element) sisällä
- Dokumentissa voi olla kommentteja merkkien `<!--` ja `-->` sisällä
 - `<` ja `&` eivät saa esiintyä yksinään dokumentissa
 - muut entiteetit täytyy itse määritellä

Hyvämuotoinen XML

- XML-dokumentti on hyvämuotoista, kun
 - dokumentti alkaa XML-määrittelyllä
 - isäelementeillä on alku- ja loppuosa
 - tyhjät elementit on merkitty loppukauttaviivalla
 - elementit ovat sisäkkäin, eivät ristikkäin
 - dokumentissa on yksi elementti, jonka lapsia kaikki muut elementit ovat
 - tätä elementtiä kutsutaan dokumentin juurielementiksi tai juureksi
 - dokumentissa ei käytetä merkkejä `<` ja `&` kuin elementtien ja entiteettien alussa
- Jos dokumentti ei täyty em. vaatimuksia, se ei ole virallisesti XML:a

XML:n käyttö, perusperiaate

- XML-dokumenttien käyttö on yleensä kaksivaiheista:
 - XML-prosessori lukee XML-dokumentin jostakin lähteestä
 - sovellus saa XML-prosessorilta XML-dokumentin sisällön käyttöönsä
- Näin on pyritty helpottamaan sovellusten laatijoiden työtaakkaa ja samalla pitämään huoli, että XML-dokumentteja käsitellään samalla tavalla sovelluksesta toiseen
 - melkein kaikkiin ohjelmointikieliin löytyy yksi tai useampia rajapintoja XML-dokumenttien käsittelyyn
 - näiden rajapintojen takana on yleensä XML-parseri (tai -jäsenin), joka lukee ja muokkaa XML-dokumentin johonkin ohjelmointikielille sopivaan muotoon
- XML ei ota kantaa, mitä sovellus dokumentilla tekee

Nimiavaruus

- Nimiavaruus määritellään aina jollakin URI:lla, ts. WWW-osoitteella
- URI:t ovat hankalia, jos niitä pitäisi toistaa jokaisen elementin kohdalla
 - on kehitetty lyhennysmerkintä
 - dokumentin alussa nimiavaruus-URI:in sidotaan joku lyhenne, jota käytetään elementtien nimen edessä, esimerkiksi: `<fo:block/>` tai `<xsl:apply-templates/>`
- Myös attribuuteilla voi olla nimiavaruus
 - oletusarvoisesti attribuutti kuuluu samaan nimiavaruuteen kuin elementti, mutta myös attribuutin nimen eteen voi liittää nimiavaruuden

Nimiavaruuden määrittely

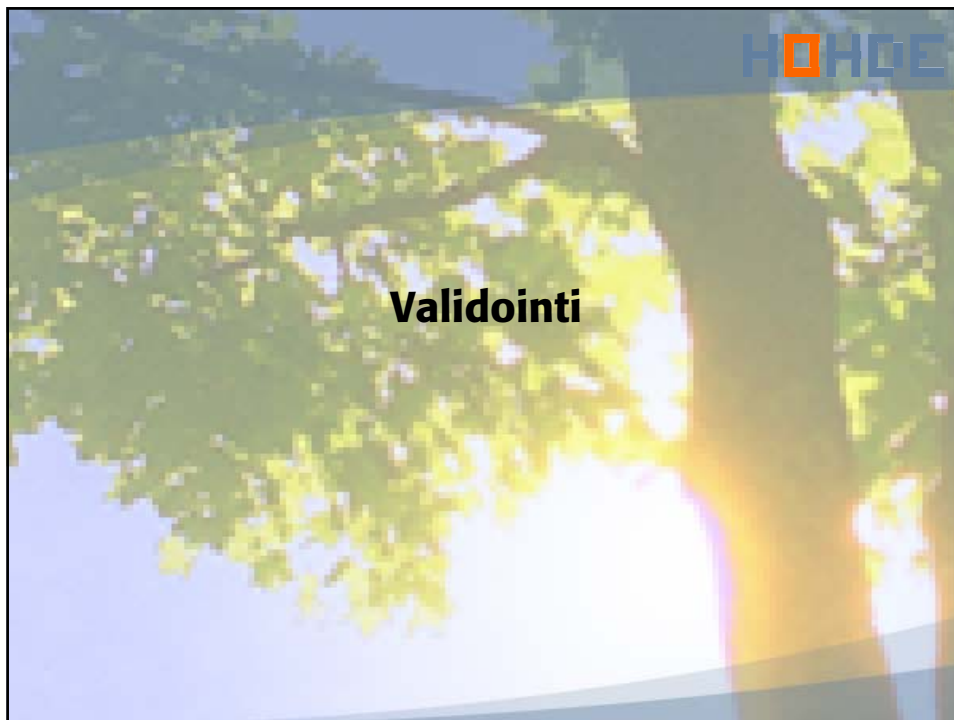
- Käytetään attribuuttia `xmlns`

```
<hohde:element xmlns:hohde="http://www.hohde.com/">
```

- Määritellään nimiavaruus ja käytetään sitä elementissä
 - tässä tapauksessa nimiavaruuden nimi on "hohde"
 - erotetaan elementistä kaksoispisteellä
- Nimiavaruuden URI:n ei tarvitse viitata mihinkään tiettyyn dokumenttiin

XML-dokumentti puuna

- XML-dokumentti voidaan esittää puumuodossa
- Sisäkkäiset elementit muodostavat puun rakenteen
 - ulompi elementti on isäsolmu ja sisempi lapsisolmu
 - vierekkäiset elementit ovat samalla tasolla puussa, sisaruksina
 - attribuutit eivät ole lapsia, vaan sisältyvät elementtiä kuvaavaan solmuun
 - tekstisirpaleet, kommentit, käsittelyohjeet ja muut vastaavat rakennepalat esitetään myös puun solmuina
 - näillä ei voi olla lapsisolmuja



Document Type Definition

HOHDE

- Document Type Definition (DTD) on määrittely, jolla luodaan XML-dokumentin rakenne
 - DTD määrittää käytettävissä olevat elementit, niiden attribuutit ja leipätekstin mahdolliset paikat
 - elementtien järjestystä ja sisäkkäisyyttä voidaan säädellä
 - attribuutit voidaan määrittää pakollisiksi tai vapaaehtoisiksi, ja niille voidaan antaa oletusarvot

www.hohde.com | © Hohde Consulting 2004

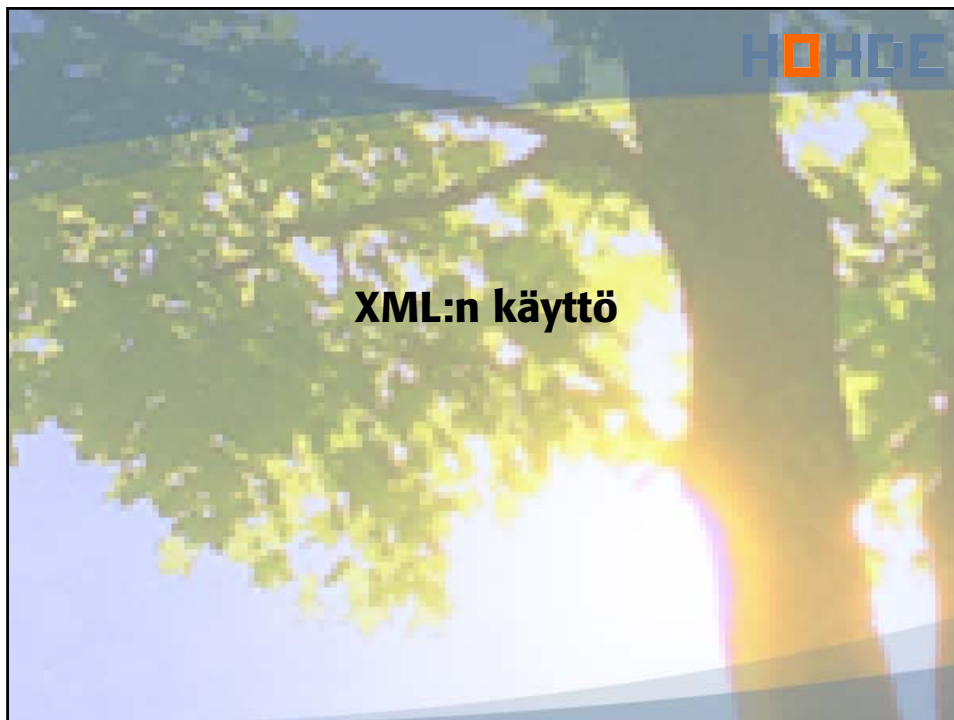
12

XML Schema

- 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 tyytit ja mahdolliset raja-arvot
 - lisätään attribuuteille oletusarvot tai todetaan tietyt attribuutit pakollisiksi
 - rakenteen pohjalta syntyy dokumentin sisältö ja semantiikka
 - määritely rakenne mahdollistaa dokumenttien tarkistuksen ja helpottaa dokumenttien koneellista käsittelyä
- XML Scheman tarkoitus on korvata DTD

Validointi

- Validoinnissa verrataan XML-dokumenttia sen DTD-määrittelyyn tai Schemaan
 - dokumentin jokainen elementti, attribuutti ja leipätekstisirpale tarkastetaan
 - mikäli DTD/Schema ei salli em. osasta tai joku vaadittu osanen uupuu, dokumentti ei ole DTD:n/Scheman mukainen ja validointi epäonnistuu
- Validointi on vapaaehtoista
 - XML-dokumentin ei ole edes pakko viitata DTD:iin/Schemaan
 - XML-prosessori suorittaa validoinnin
 - prosessorille voidaan ilmoittaa, että validointia ei saa suorittaa tai se on pakko suorittaa
- Validoinnilla sovellus voi varmistaa, että sisään luettu dokumentti on ainakin syntaktisesti oikein
 - validointi ei löydä semanttisia tai loogisia virheitä
- Validointi voi tapahtua myös XML Schemaa vasten



Tietorakenteiden esittäminen

- Suurin osa tietorakenteista liittyy rakenteessa oleviin tietoyksilöihin (solmut, nodes) ja niiden välisiin yhteyksiin (kaaret, arcs)
 - yleensä tietorakenne kannattaa kuvata kuvaamalla sen solmut elementeiksi
 - yhteydet syntyvät joko automaattisesti XML-dokumentin rakenteen mukana tai sitten ne koodataan erikseen
 - solmujen sisällä oleva tieto täytyy myös pystyä kuvaamaan
- Huomaa, että XML-kuvauksen ja tietorakenteen ei tarvitse sisältää täsmälleen samaa määrää informaatioita
 - Tietorakenne sarjallistetaan XML-dokumentiksi ja XML-dokumentista luodaan tietorakenne
 - Molemmissa prosesseissa ohjelma voi poistaa ja lisätä implisiittistä tietoa

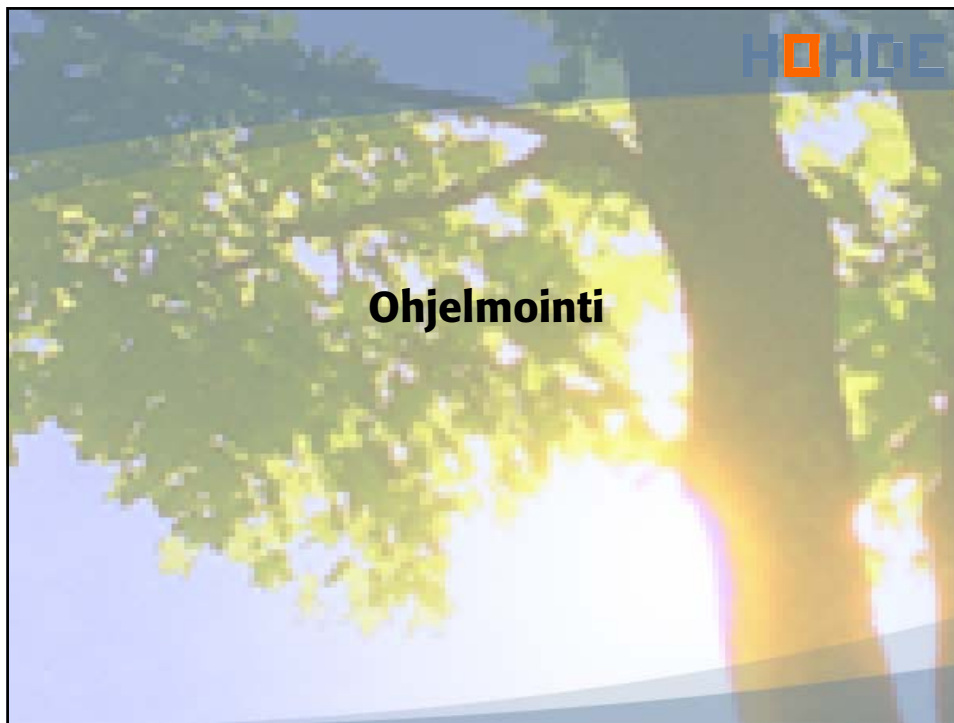
www.hohde.com | © Hohde Consulting 2004 16

Tietorakenteiden yhdistely

- Yleensä yksittäinen tietorakenne per dokumentti ei ole riittävä ratkaisu
 - Sovellukset käyttävät laajoja tietorakenteita, jotka merkitsevät kokonaisuuksina, ei yksittäisinä paloina
- Suurempi tietorakenne voidaan sarjallistaa XML:ksi kahdella eri periaatteella:
 - jokainen rakenne sarjallistetaan omaksi alidokumentiksi
 - rakenteet sarjallistetaan sisäkkäin
- Tietorakenteiden XML-muunnosten rakentaminen helpottuu, jos primitiivityypeille on sovittu yksiselitteinen esitystapa
 - tällöin ei tarvitse keksiä joka kerta uutta tapaa kuvata esimerkiksi kokonaislukua

Viittausten tarkistus

- Viittausten hallinta saattaa muodostua hankalaksi ja XML-dokumentista ei päällisin puolin näe, onko kaikki viittaukset kunnossa
 - Kone voi kuitenkin tarkistaa viittausten järkevyyden osittain automaattisesti
 - XML:ssa attribuutin tyyppi voidaan määrittää ID, jolloin sen arvon täytyy olla ainutkertainen dokumentin ID-attribuuttien joukossa
 - Siis kaikkien ID-attribuuttien, ei vain samannimisten
 - Attribuutin tyyppi voi olla myös IDREF, jolloin sen täytyy viitata johonkin ID-attribuuttiin
 - Useampaan ID-attribuuttiin voidaan viitata kerralla IDREFS-tyypillä
 - Jos ehdot eivät täyty, dokumenttia ei voida validoida



XML-tuki ohjelmointikielissä

HOHDE

- Kaikissa olennaisissa ohjelmointikielissä on mahdollisuus käyttää XML-dokumentteja
 - perustasolla tuki perustuu yleensä DOM- ja SAX-malleihin
 - tarjoavat alhaisen tason käsittelymallit XML-dokumenteille
 - käytännössä kaikki muut XML-rajapinnat rakentuvat näiden päälle
 - näiden lisäksi useissa kielissä on korkeamman tason rajapintoja, esimerkiksi olioiden sarjallistamiseksi XML-dokumenteiksi ja takaisin

www.hohde.com | © Hohde Consulting 2004

20

DOM & SAX

- Document Object Model on W3C:n suositus dokumenttien esittämiseksi ohjelmissa
 - pohjautuu dokumentin puumuotoon
 - alusta- ja kieliriippumaton
 - mahdollistaa dokumentin muokkauksen
 - DOM-rajapintojen käyttämiseksi täytyy ottaa käyttöön joku DOM-toteutus
- SAX on tapahtumapohjainen rajapinta
 - XML-dokumentti luetaan jostakin lähteestä
 - Jokainen XML:n elementti aiheuttaa tapahtuman
 - XML tulkitseva ohjelma saa dokumentin rakenteen tapahtumien kautta
 - erillistä puuta ei rakenneta

Web-palvelut

- Web-palveluissa kaksi sovellusta vaihtaa tietoa verkon yli käyttäen hyväkseen XML:a ja jotakin yleiskäyttöistä protokollaa
 - SOAP määrittää Web-palvelujen toimintamallin
 - WSDL määrittää palveluiden rajapinnan
- Näiden lisäksi voidaan puhua laajennetusti Web-palveluista, joissa pyyntö lähetetään normaalina HTTP GET-pyyntö ja vastaus saadaan XML-dokumenttina
 - XML over HTTP

SOAP

- Simple Object Access Protocol on XML-pohjainen protokolla koneiden väliseen kommunikointiin tietoverkoissa
- SOAP ei määritä mitään tiettyä protokollaa viestien siirtämiseen
 - yleensä käytetään HTTP:ta
- SOAP-viestintä sidotaan tiettyyn protokollaan (binding)
 - tämä sidonta määrittää, kuinka viestit siirretään verkon solmulta toiselle

WSDL

- Web Serviced Description Language (WSDL) on kieli, jolla voidaan määrittää Web-palveluiden rajapintoja
- WSDL-määrittämisen avulla Web-palvelun rajapintaa vastaan voidaan laatia sovellus
 - sovellus kytkeytyy palveluun ja käyttää sen tarjoamia rajapintoja
- Yksi WSDL-dokumentti voi määrittää useita Web-palveluita

Web-palvelujen käyttö

- Käytännössä Web-palvelut toteutuaan puoliautomaattisesti generoimalla koodia
 - Web-palvelun WSDL-kuvauksen pohjalta laaditaan valmis toteutus, joka pystyy lähettämään ja vastaanottamaan SOAP-viestejä
 - tämä toteutus käyttää sisäisesti edellisen kalvon kirjastoa
 - jokainen WSDL:ssa kuvattu palvelu muuntuu metodiksi
 - tätä metodia voidaan kutsua ohjelmassa normaalisti
 - SOAP:n käyttö saattaa lisätä joitakin poikkeustilanteita, joita ei normaalisti olisi
- Web-palveluun liittymisen voi tehdä myös käsin
 - kaikille ohjelmointikielille tai ympäristöille ei ole välttämättä automaattista generointia tai muutakaan SOAP-tukea

Web-palvelujen laadinta

- Vastaavasti Web-palvelujen laadinnassa ei tarvitse enää kirjoittaa WSDL-dokumenttia
 - sovellukseen laaditaan normaali rajapinta ohjelmointikielen omilla primitiiveillä
 - tämä rajapinta muunnetaan SOAP-rajapinnaksi erillisellä generaattorilla
 - tuotetaan WSDL-kuvaus
 - rakennetaan komponentti, joka tulkitsee SOAP-viestejä ja kutsuu niiden pohjalta rajapinnan metodeja
- Mikään ei tosin estä toteuttamasta kaikkea käsin



XPath

- XPath (XML Path Language) on kyselykieli, jolla XML-dokumentista voidaan poimia yksittäisiä elementtejä, attribuutteja tai jopa dokumentin osia
 - Näitä saatuja tuloksia joko käytetään hyväkseen suoraan tai sitten niitä käytetään lähtökohtana erilaisille toimenpiteille, esimerkiksi XSLT-muunnoksilla (tästä myöhemmillä luennoilla)
- XPath-lauseke koostuu joukosta askelia (step)
 - askelista muodostuu polku (path) alkupisteestä loppujoukkoon
 - jokainen askel muokkaa valittua joukkoa (set)
 - askel määrää siirtymän (axis), solmutestin (node test) ja mahdollisesti joukon ehtoja (predicates)

www.hohde.com | © Hohde Consulting 2004 28

Lausekkeen evaluointi

- XPath-lauseke evaluoidaan aina puumuotoisessa XML:ssa
 - jos XML:a ajattelee tekstidokumenttina, XPath on hyvin monimutkainen asia
- Lauseke evaluoidaan askel askeleelta
 - ensimmäinen askel otetaan esimerkiksi ns. nykyisestä (current) elementistä tai juurielementistä
 - tämä riippuu lausekkeesta
 - seuraava askel otetaan siitä, mihin edellinen jäi
 - jos askelta ei pystytä ottamaan, lauseke palauttaa tyhjän joukon
- XPath operoi joukoilla
 - jokainen askel voi saada lähtökohdaksi useita solmuja XML-puussa
 - solmujen tyyppien ei välttämättä tarvitse olla samoja

XQuery

- XQuery on helposti ymmärrettävä kyselykieli, jolla voidaan poimia tietoa XML-dokumenteista
 - XQuery on XPath 2.0 –suosituksen laajennus
 - tarkalleen ottaen molemmat suositukset ovat vielä kesken
 - XQuery käyttää XPathin funktioita ja pystyy hyödyntämään myös XPath-lausekkeita
 - tyyppitykseltään XQuery tukeutuu XML Schemaan
- XQueryyn on laadittu sekä ihmisille että koneille sopivat syntaksit
 - koneille sopivassa syntaksissa XQuery-lausekkeet ilmaistaan XML:na

XQueryn käyttö

- XQuery pohjautuu pitkälti sekvenssien käsittelyyn
 - sekvenssit voidaan ymmärtää järjestetyksi oliojoukoksi
 - XPath-lausekkeet ja funktiot palauttavat solmujoukon sekvenssinä
- XQueryn yleisin käyttötapa on käydä tietty solmujoukko läpi
 - solmujoukon sisältä poimitaan arvoja, jotka palautetaan kyselyn vastauksessa

XLink

- XLink tarjoaa tietyn joukon käsitteitä ja attribuutteja käytettäväksi missä tahansa XML-pohjaisessa kielessä
 - XLinkin attribuutit sijaitsevat omassa nimiavaruudessa, jonka URI on <http://www.w3.org/1999/xlink>
 - lyhennetään yleensä xlink
 - näillä attribuuteilla voidaan määrittää linkin tyyppi, viitattava resurssi ja antaa ohjeita dokumenttia käsittelevälle ohjelmalle

- HTML on standardointityössä korvattu XML-pohjaisella XHTML:lla
 - XHTML 1.0 pohjautuu suoraan HTML 4.01 –suositukseen
 - XHTML 1.1 on edelleen kehitetty versio 1.0:n strict-versiosta
- XHTML:a kehitetään eteenpäin
 - tarkoituksena parantaa 1.1:ssä määriteltyä modulaarista kieltä, josta voidaan koostaa tarpeita vastaavia ratkaisuja
- XHTML:a voi käyttää nykyään normaalisti HTML:n sijasta
 - selaimet osaavat tulkita molempia muotoja

- XHTML 2.0 on modulaarinen kieli
 - kaikki kielen erikoispiirteet on siirretty erillisiin moduuleihin
 - esimerkiksi typografia
 - moduulit määrittävät sekä elementtejä että attribuutteja
 - moduuli voi koostua pelkästään attribuuteista
- Moduuleista koostetaan lopullinen kieli
 - esimerkiksi XHTML+MP (mobile profile) langattomia laitteita varten
 - selaimet ymmärtävät näitä koostettuja kieliä

XForms

- XForms on W3C:n suositus lomakkeiden laadintaan WWW-sivuille
- XForms pyrkii laajentamaan WWW-lomakkeiden käyttötapoja vastaamaan kysyntää
 - tällä hetkellä moni lomake nojautuu voimakkaasti joko JavaScriptiin tai palvelimessa olevaan ohjelmistoon
- XForms-lomakkeet on jaettu kahteen osaan
 - model – määrittää lomakkeen lähettämiseen liittyvät asiat
 - body – sisältää itse lomakkeen

XSLT

- XSL Transformations (XSLT) on XML-pohjainen kieli XML-dokumenttien muuttamiseen muodosta toiseen
 - XSLT-tyylisivu kuvaa säännöt, joiden avulla XSLT-prosessori käsittelee sisään luettavaa dokumenttia (input document) ja tuottaa ulos tulevaa dokumenttia (output document)
 - XSLT-tyylisivu voi lisätä ja poistaa elementtejä ja attribuutteja sekä käsitellä leipätekstiä
- Tuotettava dokumentti on joko XML:a tai tavallista tekstiä
 - yleensä lopputuloksena on toinen XML-dokumentti
 - jos halutaan tuottaa HTML:a, tuotetaan käytännössä XHTML:a

XSLT-tyylisivun rakenne

- XSLT-tyylisivu koostuu joukosta määrittäjiä ja malleista (template)
 - template voidaan suomentaa myös sapluunaksi
- Määrittäjiä vaikutetaan tyylisivun tuottamaan dokumenttiin liittyviä asioita tai määrittämällä muuttujia myöhempää käyttöä varten
- Mallineet sisältävät dokumentin muunnossa käytettävät säännöt, sisään luetun dokumentin käsittelyohjeita ja tuotetun dokumentin elementtejä

Mallineet

- Malline koostuu kahdesta osasta
 - mallineen laukaiseva sääntö
 - mallineen sisältö
- Sääntö määritetään XPath-lausekkeella
- Mallineen sisällön rakenteen määrää tuotettavan dokumentin rakenne ja sisältö
 - malline voi tuottaa uusia elementtejä ja attribuutteja, muun muassa, tuotettuun dokumenttiin
 - malline voi valita sisään luetun dokumentin elementtejä
 - kutsua uusia malleita elementtien pohjalta
 - käsitellä elementit itse
 - malline voi sisältää ehtoja ja toistoja

Nykyinen solmu

- XSLT-prosessori käsittelee aina yhtä XML-dokumentin solmua (node) kerrallaan
- XSLT-tyylisivun mallineista valitaan se, jonka sääntö osuu tarkiten nykyiseen solmuun
 - vain yksi malline valitaan
 - tämä malline ohjaa toimintaa ja käskyy XSLT-prosessoria jossakin vaiheessa valitsemaan uuden elementin tai muun XML-dokumentin osasen nykyiseksi solmuksi
 - tämän jälkeen etsitään uusi malline
- Prosessori aloittaa dokumentin käsittelyn juurielementistä
 - nykyinen solmu viittaa juurielementtiin
 - käsittely jatkuu edellä kuvatulla tavalla

XSL-FO

- XSL Formatting Objects (XSL-FO) on kieli ulkoasun kuvaamiseksi
 - kielen avulla voidaan luoda esimerkiksi tulostettavia dokumentteja
 - pohjautuu voimakkaasti Cascading Stylesheets (CSS) versioon 2
- Kieli on sangen monimutkainen
 - yleensä XSL-FO-dokumentti syntyy XSLT-tyylisivun avulla
- FO-dokumentti koostuu
 - pohjien määrittelyistä
 - sivupohjista (page master)
 - sivujaksopohjista (page sequence master)
 - sivujaksoista (page sequence)

FO-dokumentin rakenne

- Sivupohja määrittelee dokumentissa käytettävän yksittäisen sivun koon
- Sivujaksopohja määrittelee käytettävien sivupohjien järjestyksen
 - vähintään yhtä sivupohjaa täytyy käyttää
 - sivupohjien käytölle voidaan asettaa ehtoja
- FO-dokumentin sisältö sijoitetaan sivujakson sisään
 - sivujaksossa voi olla useita staattisia lohkoja (static-content) ja yksi vuo (flow)
 - sivujaksoja voi olla useita ja ne voivat käyttää eri sivujaksopohjia
 - staattiset lohkot toistuvat sivusta toiseen samoina
 - vuo sisältää leipätekstin
 - sisältö koostuu sisäkkäisistä ja peräkkäisistä lohkoista (block)

XSL-FO-dokumentin käyttö

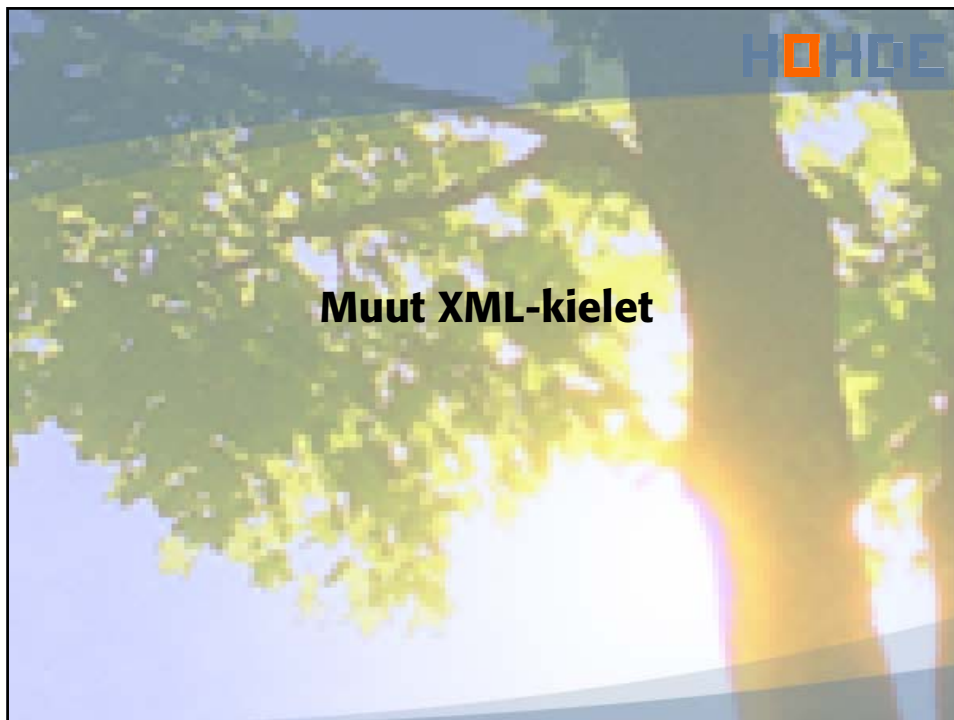
- Dokumentti laaditaan yleensä XSLT-tyylisivulla
 - tyylisivun rakenteen määrittää alkuperäinen dokumentti
 - alkuperäistä dokumenttia käsitteleviin mallineisiin sisällytetään fo-koodi
- Muista, että XSL-FO on pelkästään ulkonäköä kuvaava kieli
 - ei ole suurta väliä kuinka joku asia toteutetaan, kunhan lopputulos näyttää halutulta
- XSL-FO dokumentti näytetään tai tulostetaan erillisellä XSL-FO-ohjelmalla
 - taittaa dokumentin XSL-FO-dokumentin ohjeiden mukaan

SVG

- Scalable Vector Graphics (SVG) on XML-pohjainen kieli vektorigrafiikan esittämiseen
- SVG:lla pystyy esittämään kaikki oleelliset kaupallisista vektoripiirto-ohjelmista tutut ominaisuudet
- SVG tukee kolmen tyyppisiä olioita
 - vektoripohjaiset muodot (vector graphic shapes)
 - koostuvat kaarien ja suorien viivojen muodostamista käyristä
 - perusmuodot ovat nelikulmio (rect), ympyrä (circle), soikio (ellipsi), viiva (line), moniviiva (polyline) ja monikulmio (polygon)
 - bittikuvat (images)
 - teksti

SVG:n tuottaminen

- SVG-dokumentteja ei yleensä kannata laatia käsin
- Tuottaminen XSLT:lla soveltuu toistuvaan tuottamiseen
- Monet piirto-ohjelmat osaavat tallentaa SVG-muodossa
- SVG-kuvat vaativat tällä hetkellä erillisen näyttöohjelman
 - Javaan löytyy SVG-kirjasto
- Osa uudemmissa selaimista sisältää SVG-tuen
- SVG:ta voidaan yleensä käyttää XSL-FO:n sisällä kuvina



HOHDE

DocBook

- DocBook on XML-pohjainen kieli tekniseen dokumentointiin
 - kyseessä siis DTD-kokoelma
 - Kieli on varsin yleisessä käytössä sekä yrityksissä että Open Source –projekteissa
- Kieli on suunniteltu erityisesti tietokoneisiin liittyvän dokumentaation laadintaan
- DocBook on suunniteltu sisältöä kuvaavaksi kieleksi
 - ei ota juurikaan kantaa sisällön lopulliseen esitykseen
 - tämä on jätetty tyylisivujen ja muiden esitysjärjestelmien huoleksi

www.hohde.com | © Hohde Consulting 2004

46

Dokumentin rakenne

- DocBook-dokumentti voi olla joko kirja (book) tai artikkeli (article)
- Kirja (book) koostuu osista (part) ja luvuista (chapter)
 - kirjassa ei välttämättä tarvitse olla erillisiä osia, vaan se voi suoraan jakaantua luvuiksi
 - kirjan luvuiksi jakautuvat edelleen osioihin (sect1-5 tai section), jotka sisältävät tekstikappaleet (para)
- DocBook tukee normaalin tekstin lisäksi muutamia yleisempiä rakenteita ja suurta joukkoa tekstin tarkentimia
 - rakenteita ovat esimerkiksi taulukot ja listat
- DocBook sisältää tuen myös erilaisille luetteloille ja indekseille

DocBook-dokumentin käyttö

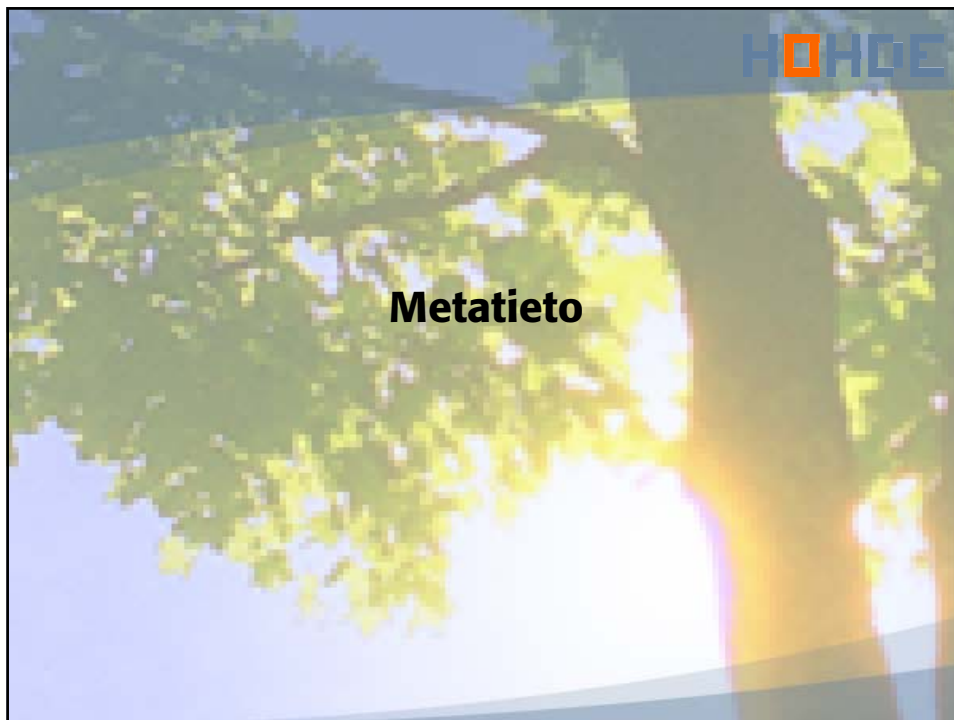
- Valmis DocBook-dokumentti ei sinällään ole suoranaisesti käytettävissä
 - ihmisten hankala lukea tekstiä XML-merkintöjen seasta
 - ei ole valmiita ohjelmia dokumenttien näyttämiseen
- Dokumentti muunnetaan toiseen muotoon XSLT:lla
 - tyylisivun voi laatia itse
 - verkossa on useita valmiita tyylisivuja
 - tuottavat HTML:a tai XSL-FO-koodia
 - osa näistä tyylisivuista mahdollistaa omien määritysten käytön

WAP

- WAP:ssa XML:a käytetään kahdessa eri muodossa
 - vanhemmat puhelimet ymmärtävät WML:lla (Wireless Markup Language) laadittuja dokumentteja
 - WML on sangen yksinkertainen kieli WWW-tyylisen sisällön esittämiseen WAP-puhelimissa
 - kieli sisältää hypertekstin perusprimitiivit
 - WML-sivu koostuu yhdestä tai useammasta kortista (card)
 - uudemmat puhelimet pystyvät näyttämään WML:n lisäksi XHTML MP (Mobile Profile) –dokumentteja
 - muut tietotyypit ovat julkisia (GIF, JPEG, JavaScript)

Synchronized Multimedia Integration Language (SMIL)

- SMIL on W3:n suositus XML-pohjaiseksi multimediasesityskieleksi
- SMIL koostuu moduuleista
 - moduulit määrittelevät tiettyyn aihealueeseen (ajastus, animaatio yms.) liittyvän toiminnallisuuden ja tarvittavat elementit
 - SMIL-näyttöohjelma toteuttaa tietyn määrän moduuleja
- Graafisen asemoinnin lisäksi SMIL mahdollistaa elementtien kytkemisen aikajalalle
 - aikajana alkaa nolosta SMIL-tiedoston näyttämisen alkaessa ja etenee näytettävän ajan mukaan
 - alueiden näkyvyys voidaan kytkeä aikajanaan



Metatieto HOHDE

- Metatieto on tietoa tiedosta
 - tarkoituksena kuvata sisältöä sovitulla tavalla
- Ilman metatietoa tietomassat ovat käyttökeltvottomia
 - koneet eivät ymmärrä semantiikkaa
- Metatieto voi kuvata periaatteessa minkä tahansa muotoista sisältöä
- Metatieto täytyy kytkeä tietoon jollakin tavalla
- Yleensä tiedon ja metatiedon välillä on yksi-yhteen – suhde

www.hohde.com | © Hohde Consulting 2004 52

Resource Description Framework

- Metatiedon määrittelykieli
 - esityskieliriippumaton
- Metatieto esitetään lausekkeilla
 - kuvaavat dokumenttien ominaisuuksia ja keskinäisiä suhteita
- RDF-metatieto koostuu kolmikoista (triplet):
 - subjekti – määrittelyn kohde
 - objekti – määrittelyn arvo
 - predikaatti – määriteltävä ominaisuus
- Kolmikossa objekti sidotaan subjektiin predikaatilla



Semanttinen web

- Semanttinen web on kehys (framework), jonka avulla tietoa voidaan jakaa helposti
 - sovellusten välillä
- Semanttisen webin ajatuksena on mahdollistaa koneiden välinen tiedonvaihto metatiedon avulla
- Yleensä siirrettävät dokumentit ovat XML-pohjaisia
- Semanttinen web ei ole sama WWW-ympäristö, johon olemme tottuneet
 - Oleellista on välittää riittävästi semanttista tietoa (rakennetta ja metatietoa), jotta kone pystyy tulkitsemaan vastaanottamaansa tietoa

Kysymyksiä? Kommentteja?