XML-kurs

Lars Marius Garshol

larsga@infotek.no
http://www.garshol.priv.no/
STEP Infotek A/S

PVV NTNU

Introduksjon

Hvem er jeg?

- Lars Marius Garshol
- Ansatt i STEP Infotek A/S siden høsten '97
- Deltidsansatt Opera Software siden høsten '99
- Hovedfag informatikk v/ UiO i '99
- Har drevet med XML siden januar '97

Kursinnhold

- Hvorfor er XML interessant?
- Rask oversikt over syntaks
- Hva kan det brukes til?
- Anvendelser, verktøy mm

Hvorfor XML?

The World Wide Web, anno 1996

- WWW inneholder millioner av dokumenter; et hav av informasjon
- Dette kan bare leses av mennesker, maskinene forstår ingenting
- Årsaken til problemet er HTML som bare beskriver hvordan sidene skal formateres
- Dette er nok for mennesker, men ikke for maskiner

Et tenkt tilfelle

- Et nettsted som automatisk samler nyheter fra forskjellige kilder
- Brukere kan registrere seg, og velge ut kildene de er interessert i
- Slik lager man sin egen personlige nyhetstjeneste
- Dette burde være enkelt å lage, eller hva?

Hvordan få tak i nyhetene?

- De står jo på websidene, og må kunne leses ut derfra?
- Problemet er at alle kildene har forskjellig struktur
- Man må derfor skrive en programsnutt for hver enkelt kilde
- Dette gjør det mye vanskeligere å legge inn nye kilder
- Se på kildene

Problemer med denne løsningen

- Mye jobb å legge inn en kilde
- Lite pålitelig: inkonsistens i formattering vil skape feil
- Når kilder endrer profil må programmet også endres
- I praksis er dette så tungvint at man sjelden tar seg bryet

Det egentlige problemet

- Formatet på nyhetssidene
- De er skrevet i HTML, og formatet reflekterer utseendet
- Dersom formatet hadde beskrevet nyheter i stedet for utseende kunne man gjort dette mye enklere
- Problemet er at det tillater ikke HTML

Problemet med HTML

- Å utvide HTML går ikke: det er for mange forskjellige typer data
- I tillegg har ikke HTML noen strikt syntaks
- (Formelt har det det, men brukerne følger den ikke alltid)
- Derfor er det svært vanskelig å skrive programvare som leser HTML

SGML?

- SGML løser akkurat dette problemet: å strukturere dokumenter slik at programvare kan lese dem
- SGML har også en klart definert syntaks og er basisen for HTML
- SGML er en ISO-standard fra 1986, og skulle derfor i teorien egne seg
- Problemet var at SGML er en stor og kompleks standard
- I tillegg er den lite praktisk for bruk over nettet

SGML Light?

- Løsningen ble å lage en forenklet utgave av SGML
- Denne skulle tilpasses nettbruk spesielt
- Samtidig skulle den være lettere å implementere
- Resultatet ble XML, ferdig i februar 1998

Hva gjør XML?

- XML lar deg definere ditt eget markeringsspråk
- HTML er definert i SGML, på samme måte lar XML deg definere dine egne alternativer
- XHTML er HTML definert på nytt ved hjelp av XML

Hvordan kan XML hjelpe?

- Hva om vi definerte et NewsML for nyheter?
- Dette kunne dokumenteres og legges ut på web, slik at kilder kunne begynne å ta det i bruk
- Deretter kunne man lage nødvendig programvare for å lese NewsML
- (og sette opp systemet med konvertering for noen kilder)
- Forhåpentlig ville folk begynne å ta det i bruk for å få mer trafikk på sine nettsteder

RSS

- RSS, Rich Site Summary, ble laget på denne måten av Netscape
- I dag finnes hundrevis av RSS-kanaler på nettet
- I tillegg finnes flere RSS-syndikerende nettsteder
- En del spesialiserte RSS-klienter finnes også
- (Det finnes også en del konkurrerende applikasjoner, som også støttes av enkelte tjenester ved siden av RSS)

RSS-eksempel

```
<?xml version="1.0"?><rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns="http://my.netscape.com/rdf/simple/0.9/">

  <channel>
    <title>Slashdot:News for Nerds. Stuff that Matters.</title>
    <link>http://slashdot.org</link>
    <description>News for Nerds.  Stuff that Matters</description>
  </channel>

  <item>
    <title>X-Files Series Spinoff?</title>
    <link>http://slashdot.org/article.pl?sid=00/02/07/0243224</link>
  </item>

  <item>
    <title>BeOS for the Internet: BeIA</title>
    <link>http://slashdot.org/article.pl?sid=00/02/06/2155205</link>
  </item>

  <item>
    <title>Try to Name the SuSE Mascot</title>
    <link>http://slashdot.org/article.pl?sid=00/02/06/2346258</link>
  </item>
</rdf:RDF>
```

Demo

- Vise RSS-viewer
- Vise my.userland.com
- Vise www.geekboys.org

XMLs bestanddeler

Hva tilbyr XML?

- Kun én ting: en måte å definere filformater på
- Ingen støtte for lagring, søking eller slike ting
- Det må man bruke andre verktøy/standarder for å få til
- XML sier ingenting om hvordan dokumentene skal lagres, men vanligvis bruker man bare filer

Elementer

- Den fundamentale byggeklossen i XML er elementene
- Et element starter med '<navn>' og slutter med '</navn>'
- Elementet skal fortelle noe om hva det som er inne i det er
- Elementer må nøstes riktig inne i hverandre (omtrent som parenteser)
- Nøstingen gjør at dokumentene får en trestruktur

Et eksempel

```
<channel>
  <title>Slashdot:News for Nerds. Stuff that Matters.</title>
  <link>http://slashdot.org</link>
  <description>News for Nerds.  Stuff that Matters</description>
</channel>
```

Strukturen til eksempelet

Sorry. Can't get images to work with this PDF generator.

Attributter

- Attributter settes på elementer for å gi mer informasjon om dem
- Typisk eksempel på dette ville være f.eks for å si hva slags språk innholdet i et element er på
- Attributter er litt det samme som elementer, men brukes som regel til ting som ikke skal vises fram
- Dette kan være forvirrende i dokumenter som ikke skal vises fram

Et eksempel

...
Edward Gibbon skrev:

<blockquote lang="en">
The influence of the clergy, in an age of superstition, might be usefully employed to assert the rights of mankind; but so intimate is the connection between the throne and the altar, that the banner of the church has very seldom been seen on the side of the people. A martial nobility and stubborn commons, possessed of arms, tenacious of property, and collected into constitutional assemblies, form the only balance capable of preserving a free constitution against enterprises of an aspiring prince.
</blockquote>
...

Andre konstruksjoner

- Kommentarer (<!-- Dette tas ikke med -->)
- Prosesserings-instruksjoner (<?target data?>)
- Tegn-referanser (A)
- Entitetsreferanser (&entitet;)
- CDATA-seksjoner (<![CDATA[Dette tolkes som tekst]]>)

Noen begreper

- XHTML og RSS er to av mange XML-applikasjoner
- En applikasjon har en dokumentmodell
- Denne består av et sett elementer, noen attributter og lovlige måter å kombinere disse på
- Denne modellen kan deklareres i en DTD (en tekst-fil)
- Både RSS og XHTML har DTD-er
- XML krever ikke at man har en DTD, SGML gjør det

Hvorfor lage en DTD?

- For å gjøre det klart for seg selv og andre hva som er lov og hva som ikke er lov i et dokument
- DTD-en fungerer som dokumentasjon av applikasjonen
- DTD-en kan også brukes til å sjekke at dokumenter er korrekte
- I tillegg kan redigeringsprogrammer bruke DTD-en til å hjelpe brukeren med å skrive dokumenter riktig

Et DTD-eksempel

```
<!ELEMENT channel (title, link, description?)>
<!ELEMENT title       (#PCDATA)>
<!ELEMENT link        (#PCDATA)>
<!ELEMENT description (#PCDATA)>
```

Krav til XML-dokumenter

- Hvis dokumentet følger syntaksen er det velformet
- Dokumenter som ikke er velformet vil ikke fungere
- Hvis dokumentet har en DTD og følger den er det gyldig
- Dokumenter som bryter med DTD-en kan av og til fungere likevel

Ikke velformet

```
<!DOCTYPE channel SYSTEM "channel.dtd">
<channel>
  <title>Slashdot:News for Nerds. Stuff that Matters.</title>
  <link>http://slashdot.org</link>
  <description>News for Nerds.  Stuff that Matters</description>
</chanel>
```

Velformet, men ikke gyldig

```
<!DOCTYPE channel SYSTEM "channel.dtd">
<chanel>
  <title>Slashdot:News for Nerds. Stuff that Matters.</title>
  <link>http://slashdot.org</link>
  <description>News for Nerds.  Stuff that Matters</description>
</chanel>
```

Velformet og gyldig

```
<!DOCTYPE channel SYSTEM "channel.dtd">
<channel>
  <title>Slashdot:News for Nerds. Stuff that Matters.</title>
  <link>http://slashdot.org</link>
  <description>News for Nerds.  Stuff that Matters</description>
</channel>
```

Demo

- Finn frem RSS-DTD-en
- Skriv et dokument i Emacs som følger den
- Åpne dokumentet i Explorer
- Skriv et CSS-stilsett for det?

XML-familien av standarder

XLink

- XLink kan brukes til å lage linker i XML-applikasjoner
- Man velger selv navn på elementene, men programvare som støtter XLink kan gjenkjenne dem som linker
- XLink støtter alt HTML har av linking, men går langt videre
- XLink kan ha linker mellom flere ankere og med forskjellig oppførsel
- Linkene behøver heller ikke befinne seg i noen av de dokumentene de linker (out-of-line links)

XPointer

- XPointer gjør det mulig å peke inn i et XML-dokument fra utsiden
- I HTML må man selv gi navn til stedene andre kan peke til
- Med XPointer blir dette unødvendig, og man kan linke til ethvert sted eller område i et annet dokument
- XPointer fungerer ved å 'klatre' eller 'søke' i dokumenttreet

XSLT

- Et språk for å beskrive transformasjoner mellom XML-dokumenter
- Med XSLT kan man generere et XML-dokument fra et annet
- XSLT er regel-basert og enklere enn programmering
- Syntaksen er XML
- Brukes ofte til konvertering til XHTML og XSL

XSL

- Omfattet tidligere også XSLT, men er nå bare formattering
- En XML-DTD som beskriver formatteringen av et dokument
- Nyttig som utgangspunkt for konvertering til PostScript/PDF
- Ganske avansert formatteringsmodell med sidemodeller, tabeller og mye annet

Flere familiemedlemmer

- XML Schemas: Utvidede DTD-er i XML-syntaks med datatyper
- RDF: XML-basert språk for å beskrive ressurser på web
- CSS: Enklere stilsett-språk for XML (og HTML)
- DOM: Standardisert API mot XML-dokumenter
- SAX: Standardisert API mot XML-parsere
- Spørrespråk: vil komme etterhvert

XML-applikasjonene

- MathML: DTD for å beskrive matematiske formler
- SVG: DTD for vektor-tegninger
- XHTML: HTML i XML
- (XSL hører egentlig også hjemme her)

The big picture

Sorry. Can't get images to work with this PDF generator.

Hva kan XML brukes til?

Dokumenthåndtering

- XMLs opphav, SGML, ble laget for dokumenter
- Ideen her er at man lager en DTD for dokumentenes underliggende struktur
- Formatering legges på automatisk via spesialverktøy
- Dette gjør at dokumentene blir svært fleksible og at mye kan automatiseres

Generalisert markup

- GM er en idé, ikke en standard eller en syntaks
- GM handler om dokumenter, ikke e-noesomhelst
- Idéen er at dokumenter kan beskrives som en trestruktur av elementer, der hvert element spiller en rolle i et abstrakt dokument
- Det vil si, noe er tittel, noe er avsnitt, noe er uthevet ord osv

Et eksempel

- Denne foilen er et eksempel, den har:
  - en tittel
  - en topptekst
  - en bunntekst
  - to lister med punkter, den ene inni den andre
- Formateringen er hint til leseren om hvordan teksten skal tolkes og inndeles i elementer

Syntaks

```
<slide>
<title>Et eksempel</title>
<point>Denne foilen er et eksempel, den har:
<list>
<item>en tittel</item>
<item>en topptekst</item>
<item>en bunntekst</item>
<item>to lister med punkter, den ene inni den andre</item>
</list>
</point>
<point>Formateringen er hint til leseren om hvordan teksten skal
tolkes og inndeles i elementer</point>
</slide>
```

Dokumenttyper

- I GM deles dokumentene inn i klasser:
  - foil-basert presentasjon
  - leksikon-artikkel
  - avisartikkel
- Hver dokumenttype har en abstrakt modell som sier hva dokumentene kan bestå av og hvordan elementene kan kombineres
- Denne modellen kan formaliseres i en DTD

Hensikten

- Å strukturere dataene et dokument består av
- Dette gjør dem uavhengige av en bestemt anvendelse
- Det kan sees på som om man gjør dokumenter om til databaser, med DTD-en som skjema

Resultatet

- En publikasjon inneholder mange utenlandske fraser
- Disse skal listes i en egen liste med forklaringer i et tillegg
- På papir settes uthevede fraser, ord som defineres og utenlandske fraser i kursiv
- Så hvordan generere listen automatisk?
- Og hvordan sjekke automatisk at den er komplett?
- Og hva om du vil sette farge på definerte ord?

XML på web

- XML kan brukes til å effektivisere web-publisering
- Det kan brukes til å gjøre data tilgjengelig for andre (syndikering)
- Det er fundamentet for mange web-standarder
- Det kan brukes til datautveksling, som protokoll-syntaks, ...

Datautveksling

- XML gjør datautveksling enklere fordi det har en standardisert syntaks
- Man trenger bare å bli enige om en DTD (pluss pluss)
- Å produsere XML-dokumenter fra programmer er svært enkelt
- Ferdige parsere gjør det enkelt å lese dem inn igjen

EDI

- Dette har gjort at man har begynt å se på XML til EDI
- XML kan enkelt erstatte EDIFACT som dataformat
- Problemet er avtaleverket rundt EDIFACT som standardiserer kodene i EDI-meldinger

Serialiserte objekter

- Flere open source kontorprogrammer bruker XML som dataformat
- SVG er et annet eksempel på det samme
- Det finnes også applikasjoner for serialisering av objektstrukturer, med implementasjoner i flere språk (WDDX)

Konfigurasjonsfiler

- XML egner seg godt til forskjellige typer konfigurasjonsfiler, fordi:
  - det er lett å lese inn
  - det er lett å skrive ut
  - det kan håndtere vilkårlig komplekse data
- Mozilla bruker dette allerede til å beskrive GUI-et (XUL)

XBEL: Et eksempel

- XBEL er en XML-applikasjon for bokmerker
- Programvare finnes for å konvertere til og fra bokmerker i forskjellige formater
- I tillegg kan XBEL også brukes til å vedlikeholde enkle lenkesamlinger på web
- Gir man ut XBEL-filen kan andre importere disse som bokmerker

Et XBEL-dokument

```
<?xml version="1.0"?>
<!DOCTYPE xbel PUBLIC "+//IDN python.org//DTD XML Bookmark Exchange Language 1.0//EN//XML" "xbel.dtd">
<xbel>
    <desc>LMGs Opera-bokmerker</desc>
    <bookmark href="file://localhost/C:/Program Files/Python/Doc/lib/module-xmllib.html" added="20000115" visited="20000115" >
      <title>12.3 xmllib -- A parser for XML documents</title>
    </bookmark>
    <bookmark href="http://pc-grove.infotek.no/cgi-bin/viewcvs.cgi/" added="19991214" visited="19991214" >
      <title>CVS Repository</title>
    </bookmark>
</xbel>
```

Demo

- Vis hvordan kan konvertere fra Opera -> XBEL -> Netscape
- Vis at Netscape-bokmerker kan importeres
- Vis hvordan XBEL kan konverteres til HTML

Et XML-basert nettsted

- Jeg vedlikeholder en oversikt over gratis XML-verktøy
- For hvert verktøy har jeg litt metadata samt en kort beskrivelse
- Nettstedet kan navigeres via en rekke indekser, samt et søkeverktøy
- I tillegg produseres en liste over oppdateringer automatisk
- Denne listen publiseres også som en RSS-kanal

Teknisk løsning

- Innholdet redigeres som ett stort XML-dokument
- En Python-modul kan laste dette inn i en objektstruktur
- Derifra kan Python-script produsere alle websidene og RSS-kanalen
- I tillegg dumpes søkeindeksene for bruk av CGI-script

XML i selvangivelsen

- Skattedirektoratet har gjort et prosjekt for elektronisk innsending av selvangivelser
- Tanken er å gradvis erstatte papirskjemaene fullstendig
- Systemet skal kunne motta data fra mange forskjellige kilder, som f.eks mail, web, x.400, ...

Mottakssentralene

Sorry. Can't get images to work with this PDF generator.

En måte å bruke XML på

- Den mest opplagte måten å bruke XML på i dette prosjektet er å lagre dataene i selvangivelsene i XML
- Slik får man ett format som er enkelt å parse, validere og generere
- Samme format kan også brukes for oversendelse til Skattedirektoratet

En annen måte

- Skattelovene endres svært ofte, og skjemaene må jo følge dem
- Dette krever endringer i:
  - valideringsprogramvaren
  - grensesnittet for innskriving av data elektronisk
- Å endre kildekode er dyrt...

En løsning

- Man kan lagre feltene i skjemaet og forholdene mellom dem i XML:
  - felt 1 er 'Navn', felt 2 er 'Adresse', ...
  - felt 15 er summen av 11-14
  - felt 17 må være 0 eller større enn 15'000,-
  - hvis felt 18 er 'ja', må felt 19-22 fylles inn
  - ...

Hvordan dette hjelper

- Valideringsprogramvaren kan generaliseres
- Ved å ta utgangspunkt i XML-beskrivelsen av skjemaet kan man validere innholdet i hver enkelt selvangivelse svært detaljert
- Grensesnittet ellers kan også generaliseres
- Framvisning, validering og brukerhjelp kan baseres på skjemabeskrivelsen

RecipeML

- En XML-applikasjon for å beskrive matoppskrifter
- Har elementer for beskrivende tekst, ingredienser, beskrivelse av tilberedning og metadata
- Mulige metadata:
  - opphavsland
  - tilberedningstid
  - vanskelighetsgrad
  - type rett (suppe, forrett, ...)

RecipeML 2

- Ingrediensene beskrives med navn, mengde og enhet
- Alternativer kan også angis
- Tilberedningslisten kan også ha alternativer

recipes.com

- Søketjenesten for å finne oppskrifter kan gjøres svært avansert
- Vil typisk implementeres ved at metadata ekstraheres fra XML-dokumentene og legges i en database
- Selve dokumentene kan konverteres til HTML for framvisning, og PDF for de som ønsker å skrive dem ut

Ingrediensinformasjonen

- Kan kobles mot en database med næringsinformasjon
- Dermed kan man lage et nytt metadatafelt: kalorier
- Kan også kobles mot en database med prisinformasjon
- Nytt metadatafelt: pris

Registrerte brukere

- Kan få systemet til å komponere menyene deres, gitt visse preferanser
- Gitt menyene kan man jo også sette sammen handlelister

Intelligent framvisning

- Med en applet eller JavaScript kan man vise frem oppskriften intelligent
- Alternativer som velges bort i ingredienslisten fjernes, og tilsvarende skritt i tilberedningen fjernes
- Næringsinnholdet oppdateres når alternativer velges

Intelligent framvisning

- Under tilberedningen kunne appleten:
  - utelate skritt i tilberedningen som er fullført
  - vise ingrediensene som trengs i dette skrittet
  - gi beskjed om ting som må gjøres, som f.eks forvarming av ovn, skritt som må startes snart, ting som har stått i ovnen lenge nok

Science fiction

- I fremtiden kan det være at kjøkkenmaskinene vil ha IP-adresser
- Appleten kunne dermed faktisk holde rede på matlagingen din
- Du ville også kunne søke på "oppskrifter jeg kan lage med de ingrediensene jeg har i mitt kjøleskap"

Email-demo

- Designe DTD med forslag fra salen
- Lage et eksempel-dokument

Email i XML

- Vil trenge egne mail-klienter som forstår DTD-en
- Disse vil kunne vise frem mailer slik bruker ønsker:
  - farge/font på signatur (utelate?)
  - farge/font på sitert tekst (utelate?)
  - kollapse/folde ut sitert tekst
- I tillegg trenger man ikke lenger passe linjelengden
- Mailprogram formaterer automatisk (uten at man mister ASCII-grafikk)

XML-programvare

Parsere

- Parsere er programpakker som kan brukes til å lese XML-dokumenter
- De sparer programmer for arbeidet med å lese elementer og attributter selv
- Parsere er først og fremst nyttige for programmerere
- De kan også brukes til å sjekke at dokumenter er velformede/gyldige
- To typer:
  - Validerende: leser DTD og sjekker gyldighet
  - Ikke-validerende: kan lese DTD, sjekker ikke gyldighet

Parser-modellen

Sorry. Can't get images to work with this PDF generator.

Editorer

- En XML-editor forstår strukturen i et dokument
- Den kan sørge for at man kun produserer velformede dokumenter
- Kan også lese DTD-en og bruke den til veiledning og kontroll
- Gode editorer kan også scriptes for å automatisere oppgaver
- Noen editorer bruker også stilsett for å vise frem dokumentet

Nettlesere

- Mozilla, MSIE 4, MSIE 5 og Opera 4 støtter XML + CSS
- MSIE 5 støtter også XML + XSL (en gammel utgave av XSL)
- Ingen av nettleserne støtter XLink eller XPointer
- Ingen støtter XSL:FO
- Mozilla støtter MathML!

Konverteringsverktøy

- To typer: til XML og fra XML
- XSL brukes mye til konvertering fra XML til HTML og andre ting
- For konvertering til XML finnes mange ulike verktøy
- Typiske datakilder: HTML, databaser, RTF, Word-dokumenter

Avslutning

Hvordan komme i gang med XML?

- Finn deg en XML-applikasjon, og gjør noe med den
  - MathML: vise frem og utveksle formler
  - SVG: lage og vise frem vektorgrafikk
  - RSS: lag din egen weblog
  - XBEL: publisér URL-lister fra XML
- Skaff deg en parser, og validér dokumentene dine
- Skaff deg kanskje også en editor?
- MSIE, Mozilla og Opera kan vise frem dokumentene dine

Å lage en applikasjon selv

- Lag et eksempel-dokument først!
- Deretter kan du lage en DTD
- Så kan du begynne med
  - CSS for fremvisning i en nettleser?
  - XSLT for konvertering?
  - DOM/SAX for programmering?

Programmeringsspråk og XML

- Java: masser av programvare, alt du trenger
- Python: nok programvare, stort sett det du trenger
- Perl: også nok programvare, mesteparten av det du trenger
- C/C++: noe mangler fortsatt, men det kommer seg
- Ellers: parsere finnes for de fleste språk

Mer informasjon

- http://www.infotek.no/ (legger foilene her på mandag)
- http://www.oasis-open.org/cover/
- http://www.garshol.priv.no/download/xmltools/
- http://www.xmlinfo.com/