Treball de Fi de Carrera

XML i Web Semàntica

Els formats per a la sindicació del contingut web : RSS i Atom

Estudiant: David Puchercós Ferrer
Titulació: Enginyeria Tècnica Informàtica de Gestió
Consultor: Òscar Celma Herrada
Data: 19 de Juny del 2006

RESUM

El format RSS (de l'anglès: *Rich Site Summary* versió 0.91-, o *RDF Site Summary* -versions 0.9 i 1.0-, o *Really Simple Syndication* –versió 2.0-) és una família de formats que, juntament amb el format Atom, són l'estàndard *de facto* per a la sindicació de contingut web.

En aquesta memòria desenvolupem els conceptes associats al RSS i Atom:

- què és la web semàntica i la sindicació web,
- quines són les particularitats del XML i del seu llenguatge de consulta, el XQuery,
- les característiques dels formats RSS i Atom, i
- les bases de dades XML natives.

També s'ha realitzat un cas pràctic usant aquestes tecnologies. Es tracta d'una aplicació web en PHP que permet al usuari accedir a una web on es mostren les particularitats dels formats RSS i Atom. L'usuari pot accedir , a través de la web, a la base de dades XML nativa *eXist* on es troben documents en formats RSS i Atom precisament per exemplificar aquests formats.

Índex

Índex de figures

1. INTRODUCCIÓ

1.1. Punt de partida

Actualment la World Wide Web està basada principalment en documents escrits en HTML.

HTML és vàlid per proporcionar dades per adequar l'aspecte visual del document i incloure objectes multimèdia i hipervincles. Però dona poques possibilitats per categoritzar el text més enllà de les típiques funcionalitats estructurals.

Les investigacions sobre web semàntica s'ocuparà de resoldre aquestes deficiències.

Ja existeixen aplicacions sobre aquestes recerques, com per exemple la sindicació web.

La tecnologia que possibilita la sindicació web és el llenguatge XML, en concret dos dels seus llenguatges:els formats RSS i Atom.

1.2. Objectius

- Conèixer el concepte de web semàntica i les tecnologies i investigacions que s'estan portant a terme.
- Introduir-se en el llenguatge XML i en el seu llenguatge de consulta, el XQuery.
- Estudiar els formats RSS i Atom, així com la sindicació web.
- Ampliar els coneixements sobre les base de dades , en concret sobre les base de dades XML natives.

1.3. Enfocament i mètode seguit

Inicialment hem recollit la informació necessària relacionada amb tots els conceptes mencionats en el punt anterior.

Aquesta ha estat el gruix més important de feina portada a terme, i que trobem recollida en aquesta memòria.

Un cop assimilats, hem estudiat el llenguatge PHP per desenvolupar una aplicació web.

Finalment hem construït l'aplicació que es tracta d'una web didàctica, on trobem les característiques i exemples dels formats RSS i Atom.

1.4. Planificació del projecte

La planificació per tasques següent i el diagrama de Gantt han estat elaborats amb el programari Microsoft Project.

Diagrama de Gantt

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	**Pac 1**	**11 días?**	**mar 28/02/06**	**lun 13/03/06**	
2	Recull informació inicial	6 días	mar 28/02/06	mar 07/03/06	
3	Elaboració del pla de treball	4 días?	mar 07/03/06	vie 10/03/06	
4	Període de correccions de la Pac1	2 días?	sáb 11/03/06	lun 13/03/06	3
5	Lliurament Pac 1	0 días	lun 13/03/06	lun 13/03/06	
6	**Pac 2**	**28 días?**	**mar 14/03/06**	**mar 18/04/06**	**1**
7	Recull d'informació formats XML , R	3 días?	mar 14/03/06	jue 16/03/06	
8	Estudi dels formats XML. RSS i Atoı	3 días?	vie 17/03/06	mar 21/03/06	7
9	Recull d'informació llenguatge Xque	3 días?	mié 22/03/06	vie 24/03/06	8
10	Recull d'informació bases de dades	4 días?	dom 26/03/06	mié 29/03/06	9
11	Instal•lació de la BD:XML nativa sele	1 día?	mié 29/03/06	mié 29/03/06	
12	Estudi del llenguatge XQuery	5 días?	jue 30/03/06	mié 05/04/06	11
13	Conèixer l'organització i estructura	4 días?	jue 06/04/06	mar 11/04/06	12
14	Definició de l'especificació de l'apli	1 día?	vie 14/04/06	sáb 15/04/06	13
15	Elaboració del document PAC2	21 días?	lun 20/03/06	sáb 15/04/06	
16	Període de correccions de la Pac2	3 días?	dom 16/04/06	mar 18/04/06	15
17	Lliurament Pac 2	0 días	mar 18/04/06	mar 18/04/06	
18	**Pac 3**	**25 días?**	**mié 19/04/06**	**lun 22/05/06**	**6**
19	Instal•lació del servidor web Apach	1 día?	mié 19/04/06	mié 19/04/06	
20	Estudi dels paquets del API Servlet	4 días?	mié 19/04/06	lun 24/04/06	
21	Implementació de l'aplicació web Ja	19 días?	mar 25/04/06	vie 19/05/06	20
22	Període de correccions de la Pac3	2 días?	sáb 20/05/06	lun 22/05/06	21
23	Lliurament Pac 3	0 días	lun 22/05/06	lun 22/05/06	
24	**TFC**	**22 días?**	**lun 22/05/06**	**lun 19/06/06**	
25	Elaboració Memòria	13 días?	lun 22/05/06	mié 07/06/06	
26	Elaboració Presentació	6 días?	jue 08/06/06	jue 15/06/06	25
27	Període de correccions del TFC	3 días?	vie 16/06/06	lun 19/06/06	26
28	Liurament TFC	0 días	lun 19/06/06	lun 19/06/06	

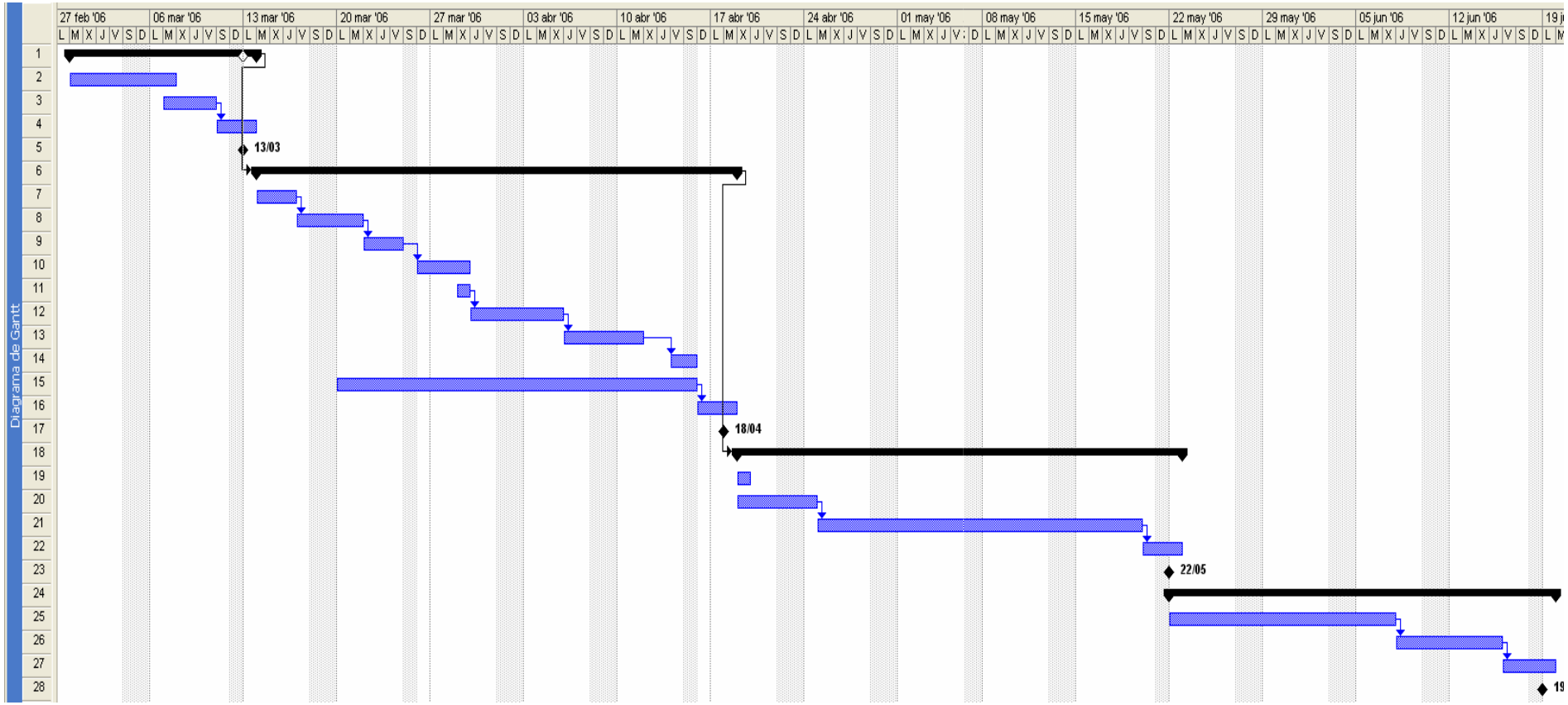


Diagrama de Gantt
27 feb '06
06 mar '06
13 mar '06
20 mar '06
27 mar '06
03 abr '06
10 abr '06
17 abr '06
24 abr '06
01 may '06
08 may '06
15 may '06
22 may '06
29 may '06
05 jun '06
12 jun '06
13/03
18/04
22/05

1.4.1. Canvis en el pla de treball

En el pla de treball inicial proposat es definia Java, amb les seves *Java Server Pages* i els *Servlets*, com el llenguatge de programació a utilitzar en el desenvolupament de l'aplicació , i el servidor *Apache TOMCAT* com a implementació dels JSP i dels *Servlets*.

Finalment, ens hem decidit per PHP i el servidor *Apache* .

Això només afecta a la planificació de la pac 3 que figura en el pla de treball pel que respecta als continguts a tractar:

PAC3	Instal·lació del servidor web Apache	19-IV
	Estudi del llenguatge de programació PHP i el conjunt de classes PHeXist	Del 19-IV fins el 24-IV
	Implementació de l'aplicació	Del 25-IV fins el 19-V
	Període de correccions de la Pac3	Del 20-V fins el 22-IV
Lliurament		22-V

1.5. Producte obtingut

L'aplicació que hem desenvolupat per exemplificar els coneixements adquirits en aquest treball és una aplicació web, programada en PHP, sobre el servidor Apache d'accés a la base de dades XML nativa eXist mitjançant el conjunt de classes PHeXist que utilitzen el protocol SOAP .

L'aplicació és una web que mostra les particularitats dels formats RSS i Atom.

El sistema permet fer consultes, des del navegador web, sobre els *feeds* emmagatzemats a la base de dades.

Es poden realitzar cerques sobre els diferents ítems, i camps, que composen els documents XML que es troben guardats a la BD.

Aquests documents XML són *feeds* de weblogs, que es troben en formats RSS o Atom.

La inserció de documents XML dins la base de dades s'ha realitzat manualment , cercant documents RSS o Atom a la web i inserint-los a la BD mitjançant un client de la BD.

2. LA WEB SEMÀNTICA

La Web Semàntica proporciona un marc comú que permet a les dades ser compartides i reutilitzades per diverses aplicacions i sistemes informàtics .Podríem dir que és una iniciativa per permetre que el significat del contingut existent a la Web sigui accessible (i es pugui processar), no tan sols pels humans, si no també per les màquines. [b.5]

És un treball col·laboratiu dirigit pel *World Wide Web Consortium* (W3C) amb la participació d'un gran nombre d'investigadors i patrocinadors de la industria del software. Un dels pilars és el *Resource Description Framework* (RDF). RDF defineix una infraestructura per a la descripció de recursos , el qual integra una varietat d'aplicacions que utilitzen el ***eXtensible Markup Language*** (XML) per a la sintaxi i les ***Uniform Resource Identifier*** (URI) que permet identificar recursos localitzats a la Web, i el *Web Ontology Language* (OWL): un llenguatge per definir ontologies a l'entorn Web. [b.6]

El llenguatge RDF és utilitzat per representar informació i intercanviar coneixement a la Web. El llenguatge OWL s'usa per publicar i compartir conjunts de termes (ontologies) permetent cerques avançades a la web i la gestió de coneixement mitjançant agents de programari. [b.6]

Un dels primers fruits de l'aprofitament d'aquestes tecnologies ha estat la possibilitat de **sindicar** part del contingut web gràcies al format **RSS** (de l'anglès: *Rich Site Summary* versió 0.91-, o *RDF Site Summary* -versions 0.9 i 1.0-, o *Really Simple Syndication* –versió 2.0-) i al format **Atom**. Parlarem de la sindicació a 3 d'aquest document.

RSS i Atom són dos formats que formen part de la família **XML**, per tant proveeixen de certa estructura a la informació que es crea i es distribueix a la Web. XML i els formats per a la sindicació web seran tractats abastament en 4 i 5 respectivament.

També ens endinsarem en el món de les base de dades, concretament amb les base de dades XML natives, orientades a emmagatzemar documents XML i que ofereixen el llenguatge de consulta **XQuery**. Veurem amb detall el llenguatge XQuery en 6 i **les bases de dades XML natives** en 7.

3. LA SINDICACIÓ DEL CONTINGUT A LA WEB

Podem definir la sindicació web com el procés de fer que certes dades d'un lloc web siguin disponibles per l'accés, transmissió, agregació o publicació a la xarxa. Això pot significar simplement donar llicència del contingut d'aquestes dades perquè hom pugui utilitzar-lo, o fer disponible el contingut d'un lloc web (*Web feed*) perquè hom pugui visualitzar una llista de continguts que es trobi actualitzada. [b.9]

És en aquest darrer cas, l'ús dels *Web feed*, on centrarem el nostre estudi. En aquest apartat explicarem què és un *Web Feed*, també coneguts com *Syndications Feeds,* quines eines per sindicació existeixen, i què són exactament, i com podem utilitzar, uns programes anomenats agregadors o *feed readers*.

3.1. Què són els Syndications Feeds

Els *Syndication feeds* s'han convertit un una eina estàndard a la Web. És fàcil trobar-se amb un petit botó taronja marcat amb unes lletres blanques que diuen XML , o potser Atom , RSS 2.0, RSS 1.0 o , inclús , *Syndication* o *Feed*. També podríem reconèixer aquest botó: 

Aquests són tots ells exemples de *syndication feeds*: són arxius amb contingut en format XML, i que mostren un conjunt dels ítems més recentment publicats ordenats cronològicament, començant pel més recent i acabant pel més antic. [b.4]

Un dels formats de *feed* més comú és el **RSS 2.0** . El RSS 2.0 és un format XML, considerat altament estable, que permet extensions exclusivament a través dels *namespaces* (veure'm els *namespaces* en 4, quan parlem amb profunditat del llenguatge XML).

L'especificació RSS 2.0 és mantinguda actualment a Harvard, i la podem trobar al lloc web http://blogs.law.harvard.edu/tech/rss. Encara que l'especificació està actualment parada, hi ha una llicència *Creative Common* que permet a qualsevol desenvolupador escriure codi pel format. Un consell consultor de tres persones prenen totes les decisions referents al status de la especificació.

Una versió anterior al RSS 2.0 és el format **RSS 0.91** que s'utilitza àmpliament en diversos llocs web.

L'altre dels formats RSS més important és el **RSS 1.0**, el qual és mantingut per una organització lliure de desenvolupadors que mantenen el seu treball en una llista de mail de *Yahoo Groups* localitzada a http://groups.yahoo.com/group/rss-dev/. L'especificació s'ha mantingut estable per diversos anys, i la podem trobar a http://web.resource.org./rss/1.0/ .

El darrer format principal s'anomena **Atom** (http://atom-enabled.org/). Va ser creat a partir de l'ús d'una *wiki* amb tota la gent interessada convidada a participar i contribuir. Atom es troba sota l'auspici del *Internet Engineering Task Force* (IETF), una organització internacional d'estàndards.

Hi ha altres variacions i versions de *feeds*, però aquestes quatre RSS 2.0, RSS 0.91, RSS 1.0 i Atom són les més comunament usades. Estudiarem els seus detalls tècnics en l'apartat 5.

3.2. El suport de la indústria

Encara que els *feed*s s'han estès àmpliament per diferents classes de llocs web, han arribat al seu nivell de popularitat avui en dia degut al seu ús dins del fenomen del *weblogging*. Els *weblogs*, o simplement *blogs*, són llocs web periòdicament actualitzats que recopilen cronològicament textos o articles d'un

o diversos autors on el més recent apareix primer, amb un ús o temàtica particular, i permeten sempre al autor publicar el que vulgui amb plena llibertat. Habitualment estan escrit amb un estil personal i informal. [b.10]

Totes les eines de creació i manteniment de *Weblogging* proporcionen funcionalitats per generar f*eeds* en un o més formats. Anem a fer una ullada a algunes d'aquestes eines:

Blogger:

L'avi de tots els entorns hostejats de *weblogs* Blogger encara respon per un percentatge significant de *weblogs*. Originàriament era una entitat separada, però ara és propietat de *Google*. *Blogger* proporciona només Atom com a tipus de *feed* predeterminat encara que si tu tens un compte, pots també seleccionar RSS 2.0. Actualment RSS 1.0 no és suportat.

http://www.blogger.com/start

Six Apart:

Ofereix tres eines *weblogging*, la més popular coneguda com LiveJournal, que només suporten Atom i RSS 2.0.

http://www.sixapart.com/

http://www.livejournal.com/

WordPress:

WordPress proporciona suport per als tres formats de *feeds* més importants – RSS 1.0, RSS 2.0 I Atom-, així com alguns formats heretats tals com RSS 0.9x.

http://wordpress.org/

ExpressionEngine:

Ofereix suport per als tres *syndication feeds* més importants.

http://www.expressionengine.com/

Blosxom:

Blosxom té diversos *plugins* que suporten RSS 1.x, RSS 2.0, Atom, i el format RSS 3.0 un *feed* no XML purament textual que va ser creat més amb finalitats didàctiques que per a un ús real.

http://www.blosxom.com/

Drupal:

Podríem considerar-lo més un complert *Content Management System* (CMS), sistema de gestió de continguts, que una eina de *weblogs*. Drupal proporciona suport per Atom, RSS 1.0, i RSS 2.0, així com per RSS 0.91 (en la seva implementació central).

http://drupal.org/

Podríem seguir amb moltes d'aquestes eines (per veure un llistat de les mateixes i els enllaços als seus portals podem accedir al recurs http://en.wikipedia.org/wiki/Blog).

Com podem observar, la majoria ofereixen suport pel RSS 2.0 i Atom, vàries ho fan pel RSS 1.0, i algunes pels encara àmpliament usats formats heretats, com RSS 0.9x.

3.3. Descobrint els *feeds*

Tornant al principi, aquells enllaços, botons de color taronja, etiquetats amb Atom, RSS, o XML, tots condueixen a un *syndication feed*. Pitjant sobre aquests botons obrirem aquest *feed* al navegador (o l'aplicació que s'hagi definit per gestionar aquest *feed*). L'eina de publicació afegeix aquesta al teu lloc web, o es pot afegir fàcilment, creant un enllaç *hypertext* amb l'etiqueta apropiada.

Per facilitar la subscripció al lloc web podem també incloure botons i enllaços cap els agregadors (o lectors de feeds) .

Els agregadors, *feed readers* o *aggregators*, són un tipus de programari que redueixen l'esforç i el temps per verificar regularment els teus llocs webs preferits (subscrits) per cercar les actualitzacions dels mateixos, creant un sol espai d'informació, a la manera d'un "tauler de notícies personal". Tal i com ja apuntàvem, els agregadors faciliten la subscripció als *feeds*.

La millor manera per aproximar-se a la publicació d'un *feed* és utilitzar el mètode de *autodiscovery*, autodescubrir-se. L' *autodiscovery* implica posar una simple línia dins la línea HEAD de cada document web, i apuntar la referència a la localització del *feed* que el conté. També s'ofereix informació respecte al tipus de *feed* de manera que els agregadors sàpiguen com trobar-lo. Quan els lectors accedeixen a la pàgina amb aquesta línia amb un navegador sensible a la sindicació *feed* es veu un indicador que senyala que el lloc web té un, o més d'un, *feed* associat i que poden subscriure's. Si els lectors posen la URL del lloc web dins del agregadors que usin, aquest trobarà de nou l'enllaç al *feed*, sense cap esforç. [b.4]

Veurem una mostra d'eines d'agregació en la propera secció.

3.4. *Feeds*: com subscriure's i llegir-los

Igual que les eines per publicar, descrites al punt 2.2, i els formats per sindicar, explicats a l'apartat 2.3, tenim moltes eleccions per a triar quina classe d'agregador *feed* usar. Podem utilitzar una eina basada en navegador -com FireFox, Safari o Opera-, llavors el *feed* apareix d'una manera similar a un marcador. O bé podem usar agregadors basats en web -integrats dins de portals ,com My Yahoo! o Google -, o d'escriptori -integrats al correu com Mozilla Thunderbid-, o per aparells de telefonia mòbil, etc.

Podem trobar una extensa llista d'agregadors al recurs http://www.newsonfeeds.com/faq/aggregators . Algunes eines només es troben disponibles per a alguns sistemes operatius i algunes requereixen que altre software es trobi instal·lat.

Un agregador d'escriptori et permet configurar el temps de comprovació de les verificacions de les actualitzacions dels *feed*. Una opció comú és fer-ho cada hora: fer-ho més freqüentment és considerat com a "mals modals", ja que afegeix tràfic innecessari al servidor del *feed*.

Un cop instal·lada l'eina i configurada, ja només cal subscriure's als lloc webs que trobis més interessants: des de *weblogs* personals fins a les publicacions més importants com el *New York Times*.

Si triem un agregador basat en web tindrem l'avantatge que no hem d'instal·lar cap programari. La desavantatge ,per descomptat, és que no tindrem accés als *feeds* quan estem desconnectats. [b.4]

3.5. Nous reptes per a la sindicació web

No essent més que un simple concepte,els *syndication feed* han generat un gran i intens interès al seu voltant en els últims anys. Una de les raons és la introducció dels *podcasting*.

El *podcasting* consisteix en fer una gravació d'una retransmissió, i llavors penjar aquesta gravació a la xarxa. Els llocs web de *podcasting* poden també oferir una descàrrega del arxiu directament, però el *feed* a que et pots subscriure que automàticament lliura nou contingut és el que distingeix un *podcast* d'un simple lloc de descàrregues o de *streaming* en temps real.

L'essència del *podcasting* és crear contingut (àudio o vídeo) per a una audiència que volen sentir i veure quan, com i el què volen.

Trobem el fenomen *feed* en altres àmbits, com per exemple:

- Moltes organitzacions de notícies, incloent Reuters, la CNN i la BBC. Aquests organitzacions permeten a altres *websites* incorporar el seu "titular sindicat" o línia de capçalera que fa de curt resum, sota algunes restriccions d'ús.
- RSS és actualment usat per diversos propòsits incloent *Marketing*, informes d'errades o qualsevol altra activitat relacionada amb publicacions o actualitzacions periòdiques.
- Moltes corporacions estan triant RSS o Atom per publicar les seves notícies i substituir *email*.

El fet és que molts consumidors i periodistes poden, actualment, disposar a l'instant de les notícies més recents en comptes d'haver de cercar-les: els agregadors verifiquen les llistes en format RSS o Atom i mostren qualsevol actualització dels articles que hi troba. [b.4]

4. XML

L'**XML**, o llenguatge d'etiquetatge extensible**,** és un llenguatge informàtic d'etiquetatge que deriva del llenguatge *Standard Generalized Markup Language* (SGML) el qual permet representar i intercanviar informació entre ordinadors o programari. [b.14]

Estrictament parlant XML no és un llenguatge d'etiquetatge. Un llenguatge té fixats un lèxic i una gramàtica, però l'XML realment no en defineix cap d'aquest elements. En comptes d'això, disposa d'unes regles sintàctiques sobre les quals hom pot construir el seu propi llenguatge. L'avantatge d'això és que , un cop construït aquest llenguatge , serà automàticament compatible amb totes les eines genèriques de software XML disponibles. [b.14]

Amb XML podem:

- Emmagatzemar i gestionar dades. XML és una bona opció per desar dades en moltes ocasions: és fàcil d'analitzar sintàcticament i d'escriure. XML funciona millor per fitxers de dades petits o per dades que no requereixen cerques aleatòries – per emmagatzemar grans volums de dades i gestionar-les ràpidament, probablement no utilitzarem XML, ja que és un mitjà d'emmagatzematge seqüencial; una base de dades relacional és una molt millor opció per aquestes tasques-.
- Transferir dades. XML és independent de la plataforma, un format llegible simultàniament per humans i màquines, suporta Unicode i verifica la integritat de les dades.

XML és un estàndard obert, que va ser dissenyat pel W3C. La recomanació actual data del 2004 i és la versió XML1.1.

Amb el terme XML s'agrupa tota una família d'estàndards, extensions i aplicacions XML que fabriquen ponts cap a altres formats o fan més fàcil treballar amb dades. [b.14]

Anem a veure un resum d'aquesta família, que tractarem, en alguns casos, en apartats posteriors:

- **Sintaxi central**. Inclou les recomanacions centrals i la seva extensió, els *namespaces* espais de noms, en XML. Veure'm la **sintaxi XML** l'apartat 3.2.
- **Documents de lectura**. Aquí trobem els llenguatges pels documents que realment es llegeixen en contrast amb els de dades;l'e**X**tensible **H**yper**t**ext **M**arkup **L**anguage(XHTML), la millora del llenguatge *Hypertext Markup Language* (HTML), per codificar pàgines web; o el *DocBook* per manuals tècnics, els quals contenen molts termes tècnics i estructures complexes com ara taules i llistes .
- **Modelatge**. En aquest grup es troben totes les tecnologies desenvolupades per crear models de documents que formalitzen un llenguatge d'etiquetatge i que poden ser usades per verificar la gramàtica de les instàncies de documents. Aquestes inclouen els Document Type Definition (DTD), el *W3C XML Schema* i el RELAX NG, dels que parlarem en l'apartat 3.1, quan parlem de la sintaxi XML.

- **Localització i enllaços** . Les dades només són útil si són accessibles. És per això que hi ha un conjunt de protocols disponibles per obtenir les dades dins dels documents XML. **XPath** proporciona un llenguatge per especificar el camí a un component individual d'una dada. XPointer i XLink utilitzen aquests camins per crear un enllaç des d'un document a un altre. El *XML Query Language* (**XQuery**) permet extreure i manipular dades des de documents XML o qualsevol font de dades que pugui ser visualitzada com XML, fins a base de dades relacionals o documents estructurats. Aquest llenguatge serà estudiat abastament a l'apartat 5, on també tractarem el llenguatge XPath.
- **Visualització**. XML no és un llenguatge orientat a la presentació de dades. Si així ho volguéssim hauríem d'usar una fulla d'estil. Les dues més populars són els *Cascading Style Sheets* (CSS) i el *Extensible Style Language* (XSL). El primer és molt simple i adequat per la majoria de documents. El segon és força detallat i millor per documents que requereixen qualitat d'impressió. EL XSL inclou el XSLT l'*'Extensible Style Language Transformation*, amb el que pots transformar automàticament documents XML en altres formats, com XHTML.
- **Multimèdia**. El *Scalable Vector Graphics* (SVG) és un llenguatge de descripció de gràfics de dos dimensions i d'aplicacions gràfiques en XML. SVG s'utilitza en diferents àrees incloent gràfics Web, animació, interfícies d'usuari, intercanvi de gràfics, aplicacions mòbils y disseny d'alta qualitat.
- **Ciències**. Les aplicacions científiques van adoptar ben aviat l'XML. El *Chemical Markup Language* (CML) representa molècules en XML, i el MathML construeix equacions. El programari transforma les instàncies d'aquests llenguatges d'etiquetatge en representacions visuals que els científics estan acostumats a veure.
- **Comunicació**. XML és una excel·lent manera perquè diferents sistemes es comuniquin uns amb els altres. Les crides entre sistemes host estan essent estandarditzades mitjançant aplicacions com XML-RPC, SOAP, WSDL i UDDI.
- **Desenvolupament.** La majoria de llenguatges de programació tenen suport per l'anàlisi sintàctic i navegació XML. Habitualment fan ús de dues interfícies estàndards. El *Simple API for XML* (SAX) és molt popular per la seva simplicitat i eficiència. El *Document Object Model* (DOM) descriu una interfície per moure's per un arbre objecte de un document per processaments més complexes. Amb l'aparició dels sistemes de base de dades XML natius (BD:XML), l'API XML:DB és una altre opció pel desenvolupament d'aplicacions que utilitzen aquests tipus de BD. Parlarem de les BD:XML natives i de l'API XML:DB en l'apartat 6.

  [b.2]

4.1. Sintaxi bàsica del XML

Aquí tenim l'exemple d'un missatge utilitzant XML (aquest exemple ha estat parcialment extret del llibre **Ray, E.T** (2001). *Learning XML* . O'REILLY [b.2]) i que usarem durant tot aquest apartat sobre la sintaxi XML

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<!DOCTYPE missatge
  SYSTEM "/xml-resources/dtds/missatge.dtd"
[
<!ENTITY client  "Sr.Armando Bronca Segura">
<!ENTITY agent "Sra.Felicidades Blancas Rojas ">
<!ENTITY telefon "<numero>933304147"</numero>>
]>
<missatge prioritat="urgent">
<capçalera>Estimat &client;</capçalera>
<cos>tenim una oportunitat per tu!! Gaudeix d'unes merescudes vacances en
un dels millors balnearis de la Costa Blava <negreta>Le Chauteu
Bleu</negreta>.Per reservar una plaça per les teves vacances truca a la
&agent; al &telefon;.
Afanya't &client;.No queden gaires places!</cos>
</missatge>
```

La primera línia que hem pintat en color vermell és la **declaració XML**. És una línia opcional que prepara el processador XML per treballar amb el document i informa de

- quina versió estem usant (la 1.1 és la oficial actualment),
- del tipus de codificació(si no està definida la codificació per defecte és la UTF-8), i
- si existeixen dependències externes (hem de tenir en compte que es diferencien les majúscules i les minúscules en els noms del paràmetres i dels seus valors)

La *Document Type Declaration* (DTD) és la part de l'exemple de color blau . Hi ha dos raons per utilitzar-la:

1. per definir **entitats** o valors per defecte dels atributs
2. per **validar** el document: és un tipus d'anàlisi sintàctic que comprova la gramàtica i el vocabulari del marcatge

1.definició d'entitats:

XML proporciona dos mètodes per representar caràcters especials: les *entity references* (referències d'entitat) i les *numeric character references.*

Una **entitat** en XML es una dada, usualment text com pot ser un caràcter inusual, a la qual li donem un nom.

Una ***entity reference*** és un referència que representat una entitat. Consisteix en el nom d'una entitat precedida per un *ampersand* ("&") i seguit per un punt i coma (";").Disposem de 5 entitats predefinides:

&amp;	&
&lt	<
&gt	>
&apos	‘
&quot	“

Aquí tenim un exemple utilitzant entitats XML predefinides per representar l'*ampersand* en el nom "AT&T"

```
<nom-companyia>AT&amp;T</nom-companyia>
```

Si hem de declarar més entitats, diferents de les predefinides, ho farem dins del *Document Type Declaration*. A continuació tenim un exemple d'un DTD intern

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE example
 [
 <!ENTITY copy "&#xA9;">
 <!ENTITY copyright-notice "Copyright &copy; 2006, XYZ
   Enterprises">
  ]>
<arrel>
 &copyright-notice;
</arrel>
```

El que veuríem en un navegador apropiat seria:

```
<arrel>Copyright © 2006, XYZ Enterprises </arrel>
```

Les *numeric character references* semblen entitats, però en comptes d'un nom contenen el caràcter "#" seguit d'un número. El número (en decimal o hexadecimal si va precedit del prefix "x") representa un codi Unicode, i s'utilitza habitualment per representar caràcter que no són fàcils d'escriure, tals com caràcters aràbics en documents produïts en un computador europeu. El *ampersand* en "AT&T" podria també ser un exemple :

```
<nom-companyia>AT&#38;T</nom-companyia>
<nom-companyia>AT&#x26;T</nom-companyia>
```

2.La **validació** d'un document **XML** :

Es basa amb la verificació de l'acompliment d'un seguit de regles definides per l'usuari. Aquestes regles les trobem en els XML S*chemes, o esquemes XML*. Disposem de diferents tipus d'esquemes XML, cadascun amb els seus punts forts i dèbils:

- *Document Type Defination* DTD
- W3C XML Schema
- Relax NG

El primer pas consisteix en indicar al programa que llegeix l'arxiu XML l'*schema* que ha d'utilitzar per mitjà de la declaració DOCTYPE. Per exemple, en l'exemple inicial

```
<!DOCTYPE missatge
  SYSTEM "/xml-resources/dtds/missatge.dtd"
>
```

Això indica que el DTD és un DTD intern i es troba a /xml-resources/dtds/missatge.dtd.

També podem indicar la ubicació del *schema* com a direcció URL, es tracte dels DTD externs:

```
<!DOCTYPE html
   PUBLIC  "-//W3C//DTD HTML 3.2//EN"
   "http://www.w3.org./TR/HTML/html.dtd"
>
```

El primer identificador, de color vermell, s'anomena **identificador públic** , que el propi programa que llegeix l'arxiu XML pot reconèixer (el que ens permet utilitzar una còpia de l'*schema* en comptes d'haver d'accedir a la web.

El segon identificador, de color blau, inclou sempre una URL de seguretat.

La declaració DOCTYPE també indica quin serà l'element arrel del document:apareix especificat en l'entrada situada després de la paraula DOCTYPE, en el exemple anterior és **html.**

La sintaxi dels *schemes* es troba fora de l'abast d'aquest treball. Si es vol aprofundir en aquesta matèria podem consultar els enllaços proporcionats anteriorment amb els tipus de *schemes* existents.

La part final del document d'exemple, de color verd, consisteix en **elements** encapsulats, alguns dels quals contenen **atributs** i **contingut**. Un element consisteix habitualment de dues etiquetes, una d'inici i una final, possiblement envoltant text i altres elements. L'etiqueta d'inici consisteix en un nom envoltat per uns parèntesis en angle tal com "<missatge>"; l'etiqueta final és el mateix nom envoltat per parèntesis en angle, però amb una barra inclinada cap a la dreta precedint el nom tal com "</missatge>". El **contingut** de l'element és tot el que apareix entre l'etiqueta d'inici i la final, incloent el text i altres elements.

La sintaxi bàsica per a un element en XML és la següent

```
<nom atribut="valor">contingut</nom>
```

Aquí tenim un element XML,amb una etiqueta inicial, contingut textual i una etiqueta final

```
<negreta>Le Chauteu Bleu</negreta>
```

Un element pot contenir **atributs:** parells nom-valor inclosos dins de l'etiqueta inicial després del nom de l'element. Els valors dels atributs sempre han d'estar entre cometes, simples o dobles,i cada nom d'atribut només ha d'aparèixer un cop dins de cada element

```
<missatge prioritat="urgent"> cosMissatge</missatge>
```

A més del text un element pot contenir altres elements. XML requereix que el elements es trobin ben anidats –els elements no poden mai estar encavalcats. Quan això succeeix es diu que el document no està **ben format**.

Tot document XML ha de tenir exactament un **element arrel**.

Per tant el següent exemple es troba mal format:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!- - ERROR! XML NO BEN FORMAT! - - >
<alguna-cosa> primera cosa </alguna cosa>
<alguna-cosa> segona cosa </alguna cosa>
```

XML proporciona una sintaxi especial per representar un element amb contingut buit. En comptes d'escriure una etiqueta d'inici seguida immediatament per una etiqueta final, un document pot contenir una **etiqueta d'element buit** on la barra inclinada segueix al nom de l'element. Per tant els dos exemples següents son equivalents:

```
<foo></foo>
```

```
<foo/>
```

[b.13]

4.2. XML Namespace

El XML *namespace* o espais de noms XML és un estàndard del W3C per proporcionar elements i atributs inequívocament diferenciats encara que puguin tenir el mateix nom. Una instància XML pot contenir noms d'elements o d'atributs de més d'un vocabulari XML. Si cada vocabulari té un *namespace* llavors l'ambigüitat entre noms o atributs que es diuen igual queda resolta.

Tots els noms d'elements dins d'un *namespace* han de ser únics.

Un *namespace* es declara utilitzant l'atribut reservat **xmlns**, el valor del qual ha de ser una referència URI.

Els *namespace* no requereixen que el seu vocabulari es trobi definit, encara que és força comú localitzar un DTD o un XML schema definint l'estructura de dades precisa en la URI del *namespace*. [b.14]

4.3. Seccions a ignorar i comentaris

Les **seccions CDATA** senyalen un fragment del document on el programari que analitza la sintaxi no ha de tenir en compte la majoria de caràcters. Veiem l'estructura d'una secció CDATA

```
<![CDATA[
```

```
 AQUI PODEM POSAR TOT EL QUE VOLGUEM P.E. <>
]]>
```

Tot el que posem dins d'aquesta secció no serà interpretat. L'única cadena que no poden posar dins d'aquesta secció és]] .

Els **comentaris** comencen amb <! - - i acaben amb - - >. Els comentaris poden contenir qualsevol dada exceptuant el literal --. Els comentaris no són part textual dels documents XML i els poden emplaçar entre qualsevol etiquetes de marcatge. [b.14]

5. RSS I ATOM

Com ja s'ha explicat a la introducció, RSS és una família de formats especificats en XML i utilitzats per a la sindicació web; tema, aquest últim, del que hem parlat abastament a l'apartat 2.

Ara estudiarem les característiques tècniques més rellevants d'aquesta família de formats.

L'abreviació **RSS** es refereix a un dels següents estàndards:

- Rich Site Summary (RSS 0.91)
- RDF Site Summary (RSS 0.9 i 1.0)
- Really Simple Syndication (RSS 2.0)

Certes deficiències en el format RSS 2.0 va motivar el desenvolupament del format **Atom** el qual està també basat en XML.

El format i l'ús dels *syndication feed* és bastant senzill. Cada *feed* conté un cert nombre de *items* individuals, normalment amb un *title* (títol), *last update* (última actualització), *update frequency* (freqüència d'actualització), *site owner* (propietari del lloc web), i un llarg etcètera. Dins d'aquests camps simples, hi ha una gran varietat d'opcions que condueix a una enorme diferenciació entre *feeds*. Per fer-ho més entenedor, farem una ullada a cadascun dels formats dels *feeds* per veure quins camps són compartits, i quins són específics per cada format. [b.4]

Elements *Container*

El *feed* "*container*" és informació de tot el lloc web, consistint en diversos camps que es troben llistats dins del *feed*. Inclosos dins d'aquest *container* es troben els següents camps que es repeteixen a través de la majoria de *syndication feeds*:

Camp	Descripció
link	Es tracta d'un enllaç per la URL del lloc web
title	És el títol del lloc
description	És una descripció del lloc web, i habitualment conté el subtítol del lloc. En Atom és el camp *subtitle*.
autor	En Atom, *author* és una estructura que conté el nom de l'autor i el seu email, o només el mail de l'autor. En RSS 2.0, aquest camp és canviat per *webMaster* i *managingEditor*, les quals són adreces mail. En RSS1.x, normalment es tracta per *dc:creator*, el qual pot tenir o bé una estructura o un valor simple.
date	En RSS 1.x, aquest camp habitualment dc:date és la data de quan el feed va ser actualitzat. En Atom és el camp *update*; en RSS 2.0, aquest és *lastBuildDate*
generator	En RSS 2.0, aquesta és la eina utilitzada per generar el *syndication feed.* El *feed* Atom també utilitza *generator*, mentre que RSS 1.x usa *generatorAgent*
copyright	Informació del copyright
language	En quina llengua es troba el document
id	Camp específic per Atom, que proporciona un identificador únic pel lloc web
image	Una icona o imatge representant el *feed* o el lloc web tant per RSS

	1.x com per RSS 2.x. En Atom això pot ser un *icon* encara que *logo* pot ser usat per una representació del logo. En RSS 1.x i RSS 2.x *image* és una estructura apuntant a una URL d'imatge, el títol i un enllaç al lloc web.

Anem a veure alguns exemples de *feeds:*

RSS 1.0 *feed*

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:sy="http://purl.org/rss/1.0/modules/syndication/"
xmlns:admin="http://webns.net/mvcb/"
xmlns:cc="http://web.resource.org/cc/"
xmlns="http://purl.org/rss/1.0/"
xmlns:feedburner="http://rssnamespace.org/feedburner/ext/1.0">
<channel rdf:about="http://radar.oreilly.com/">
<title>O'Reilly Radar</title>
<link>http://radar.oreilly.com/</link>
<description>http://radar.oreilly.com/</description>
<dc:creator />
<dc:date>2005-10-13T08:47:28-08:00</dc:date>
<admin:generatorAgent rdf:resource="http://www.movabletype.org/?v=3.2" />
<cc:license rdf:resource="http://creativecommons.org/licenses/by-nc-sa/1.0/"
/>
...
</channel>
</rdf:RDF>
```

RSS 2.0 feed

```
<rss version="2.0">
<channel>
<title>Scripting News</title>
<link>http://www.scripting.com/</link>
<description>It's even worse than it appears.</description>
<language>en-us</language>
<copyright>Copyright 1997-2005 Dave Winer</copyright>
<pubDate>Thu, 13 Oct 2005 04:00:00 GMT</pubDate>
<lastBuildDate>Thu, 13 Oct 2005 15:42:37 GMT</lastBuildDate>
<docs>http://blogs.law.harvard.edu/tech/rss</docs>
<generator>UserLand Frontier v9.0.1</generator>
<managingEditor>dwiner@cyber.law.harvard.edu</managingEditor>
<webMaster>dwiner@cyber.law.harvard.edu</webMaster>
...
</channel>
</rss>
```

Atom *feed*

```
<feed xmlns="http://purl.org/atom/ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:feedburner="http://rssnamespace.org/feedburner/ext/1.0"
version="0.3" xml:lang="en">
<title>O'Reilly Radar</title>
<link rel="alternate" type="text/html" href="http://radar.oreilly.com/" />
<modified>2005-10-13T17:13:35Z</modified>
<tagline>http://radar.oreilly.com/</tagline>
<id>tag:radar.oreilly.com,2005://24</id>
<generator url="http://www.movabletype.org/" version="3.2">Movable Type
</generator>
<copyright>Copyright (c) 2005, O'Reilly Media, Inc.</copyright>
<link rel="start" href="http://feeds.feedburner.com/oreilly/radar/atom"
type="application/atom+xml" />
...
</feed>
```

La diferència més important entre el format RSS 1.0 i els altres formats és que tots els *items* (que veurem tot seguit) inclosos en un *feed* RSS 1.0 són primer llistats en un element anomenat *items*, i després definits completament en altres parts del document

RSS 1.0 *feed* (exemple de llistat de *items*)

```
<items>
<rdf:Seq>
<rdf:li rdf:resource="http://radar.oreilly.com/archives/2005/10/re_sensible_
email_messages.html" />
<rdf:li rdf:resource="http://webkit.opendarwin.org/" />
<rdf:li rdf:resource="http://radar.oreilly.com/archives/2005/10/bitkeeper_v_
everyone.html" />
<rdf:li rdf:resource="http://www.peatsbooks.com/books" />
<rdf:li rdf:resource="http://www.engadget.com/entry/1234000207062697/" />
<rdf:li rdf:resource="http://docs.yahoo.com/docs/pr/release1265.html" />
<rdf:li
rdf:resource="http://news.com.com/Palm%20drops%20Zire%2C%20Tungsten%
20names/2100-1041_3-5893455.html" />
<rdf:li rdf:resource="http://radar.oreilly.com/archives/2005/10/yahoo_
research_berkeley_launch.html" />
<rdf:li rdf:resource="http://www.dailykos.com/
storyonly/2005/10/11/154544/44" />
<rdf:li rdf:resource="http://radar.oreilly.com/archives/2005/10/eurooscon_
maker_faire_lineup.html" />
<rdf:li rdf:resource="http://radar.oreilly.com/archives/2005/10/google_maps_
and_their_data_pro_1.html" />
<rdf:li rdf:resource="http://radar.oreilly.com/archives/2005/10/your_money_
or_your_mysql_1.html" />
</rdf:Seq>
</items>
```

Elements *Entry*

Cada *entry* entrada individual d'un lloc web té la seva pròpia *entry* dins del *feed* fins el nombre designat per cada *feed* (habitualment es troba entre 5 i 15). En Atom les *entry* es troben designades per elements *entry*, mentre que en RSS es llisten com *item.*

Anem a examinar els elements més comuns dins d'un *entry*.

Camp	Descripció
link	Es tracta d'un enllaç al *item* individual
title	És el títol del *item*, si en té
description	En RSS, *description* és un resum o descripció del *item*. En Atom és el camp *summary*
autor	En Atom , *author* és una estructura que conté el nom de l'autor i el seu email, o només el mail de l'autor. En RSS es tracta per *dc:creator*.
guid	Proporciona un identificador únic pel *item*. En Atom, és *id*
content	És el complert contingut codificat del *item* incloent qualsevol etiqueta de marcatge HTML o XHTML. També pot ser anomenat *content:encoded* en RSS1.x.
pubDate	Data de publicació del *item*. Això és *dc:creator* en RSS
category	La categoria del *item*

Exemples d'entrades *feed:*

RSS 1.0 *feed*

```
<item rdf:about="http://webkit.opendarwin.org/">
<title>The WebKit Open Source Project</title>
<link>http://feeds.feedburner.com/oreilly/radar/rss10?m=520</link>
<description>By rael
WebKit, the embeddable Web browser behind Mac OS X's Safari, Mail.app,
Dashboard, and third party apps like NetNewsWire and TextMate, now makes
nightly builds available....&lt;img src="http://feeds.feedburner.com/oreilly/
radar/rss10?g=520"/&gt;</description>
<dc:subject>apple browser affordances web etech etech06 safari</dc:subject>
<dc:date>2005-10-13T08:34:02-08:00</dc:date>
<feedburner:origLink>http://webkit.opendarwin.org/</feedburner:origLink>
</item>
```

RSS 2.0 feed

```
<item>
<description>
Blog Herald: <a href="http://www.blogherald.com/2005/03/06/a-short-history-
ofblogging/">
A short history of blogging</a>.
</description>
<pubDate>Thu, 13 Oct 2005 05:46:28 GMT</pubDate>
<guid>
http://archive.scripting.com/2005/10/13#When:1:46:28AM
</guid>
</item>
```

Atom *feed*

```
<entry>
<title>The WebKit Open Source Project</title>
<link rel="alternate" type="text/html" href="http://feeds.feedburner.com/
oreilly/radar/atom?m=520" />
<id>http://webkit.opendarwin.org/</id>
<created>2005-10-13T08:34:02Z</created>
<modified>2005-07-03T09:06:11Z</modified>
<author>
<name>rael</name>
</author>
<dc:subject>apple</dc:subject>
<dc:subject>browser</dc:subject>
<dc:subject>affordances</dc:subject>
<dc:subject>web</dc:subject>
<dc:subject>etech</dc:subject>
<dc:subject>etech06</dc:subject>
<dc:subject>safari</dc:subject>
<summary type="text/html" mode="escaped">By rael
WebKit, the embeddable Web browser behind Mac OS X's Safari, Mail.app,
Dashboard, and third party apps like NetNewsWire and TextMate, now makes
nightly builds available....</summary>
<content type="text/html" mode="escaped" xml:lang="en" xml:base="http://radar.
oreilly.com/">By rael
WebKit, the embeddable Web browser behind Mac OS X's Safari, Mail.app,
Dashboard, and third party apps like NetNewsWire and TextMate, now makes
nightly builds available....
&lt;img src="http://feeds.feedburner.com/oreilly/radar/atom?g=520"/&gt;</
content>
<feedburner:origLink>http://webkit.opendarwin.org/</feedburner:origLink>
</entry>
```

5.1. Resum de l'especificació del Rich Site Summary (RSS 0.91) [b.21]

Introduït el 1999 per Netscape com un marc de descripció d'un canal i mecanisme de categorització de contingut per al seu portal *My Netscape Network*.

Ens centrarem en les etiquetes més rellevants d'aquest format. Per ampliar-ho trobareu l'especificació complerta a http://my.netscape.com/publish/formats/rss-spec-0.91.html

En el nivell arrel, un document RSS conté un element <rss> amb un atribut obligatori anomenat versió, que en aquest cas el seu valor és 0.91. Subordinat al element <rss> trobem un element únic <channel> (canal), el qual conté informació sobre el canal que inclou.

Etiqueta	**Descripció**	**Sub-elements obligatoris**
<channel>	Informació respecte un canal particular. Tot el que pertany a un canal es troba dins d'aquesta etiqueta.	*description* *language* *link* *title*
<description>	Text amb la descripció d'un ítem, canal, imatge o entrada de text.	
<language>	Especifica el llenguatge d'un canal	
<link>	URL on un usuari pot navegar	
<pubDate>	Data de quan el canal va ser publicat	
<rss>	Identifica l'inici i el final del contingut RSS. És obligatori l'ús de l'atribut *version="0.91"*	*channel*
<title>	Una cadena de text identificadora del recurs. Quan s'utilitza amb un canal és el seu títol	

5.2. Resum de l'especificació del RDF Site Summary (RSS 1.0) [b.22]

Les creixents noves necessitats del marc de metadades que va suposar l'estàndard RSS 0.91 va fer que es comencessin a fer servir elements no oficials (per exemple <category>,<author>,<data>).

Per solucionar aquest problema es va proposar afegir més elements al cor del RSS. Aquesta solució, si bé la més simple i ràpida en posada a punt, sacrificava l'escalabilitat i requeria modificacions iteratives al format central,afegint funcionalitats que necessitàvem i eliminant les no utilitzades.

Una segona solució, el RSS 1.0 la darrera versió del qual data del Maig del 2001, va ser compartimentar les funcionalitats específiques dins de mòduls RSS usant *XML Namespaces*. Afegir i eliminar funcionalitats RSS és només una qüestió d'incloure un grup particular de mòduls adients a les tasques requerides. Cap modificació al cor del RSS és necessari. Per tant, podem dir que ens trobem amb un format extensible, mitjançant *XML Namespaces* i

Resource Description Framework (RDF), per a la sindicació que és compatible amb el format RSS 0.9.

Una restricció imposada als sub-elements d'un element canal, imatge, ídem o text és que aquests elements no poden contenir sub-elements repetits. Aquests proposta només afecta als sub-elements immediats .Qualsevol nivell més profund està ja ben definit utilitzant la sintaxi RDF.

Una extensió específica per un document RSS 1.0 no és requerida.Tant rdf com xml són recomanades, essent la primera preferida.

Tractarem només les etiquetes obligatòries <rdf> i el seu element obligatori <channel> amb els seus respectius elements obligatoris.

Etiqueta amb la seva sintaxi	**Descripció**	**Sub-elements obligatoris**
<channel rdf:about= "{recurs}">	Informació d'un canal particular, incloent un títol, una breu descripció i un link URL al canal descrit.	*description items link title*
<description>	Text amb la descripció del contingut, funció, origen etc d'un canal	
<items> <rdf:Seq> <rdf:li resource={item_uri}"/> .... </rdf:Seq> </items>	Una taula RDF de continguts, associant els ítems del document amb aquest canal RSS en particular.	
<link>	URL que enllaça amb el lloc web HTML relacionat amb el canal	
<rdf:RDF xmlns:rdf="{recurs}" xmlns="{recurs"}>	Identifica l'inici i el final del contingut RSS	*channel*
<title>	Un títol descriptiu d'un canal	

Trobem la seva especificació complerta a http://web.resource.org/rss/1.0/

Al 2005 es va lliurar una nova especificació la 1.1 per corregir alguns ítems. RSS 1.1 es pot trobar a http://inamidst.com//rss1.1/

5.3. Resum de l'especificació del *Really Simple Syndication* (RSS 2.0) [b.23]

Inicialment presentada a la tardor del 2002, afegeix la capacitat introduïda al format RSS 1.0 d'extensibilitat mitjançant *namespaces*.

En el nivell arrel, un document RSS conté un element <rss>, amb un atribut obligatori anomenat versió, que en aquest cas el seu valor és 2.0.

Subordinat al element <rss> trobem un element únic <channel> (canal), el qual conté informació sobre el canal que inclou.

Etiqueta	Descripció	Sub-elements obligatoris
<*channel*>	Informació respecte un canal particular. Tot el que pertany a un canal es troba dins d'aquesta etiqueta.	*description* *link* *title*
<*description*>	Text amb la descripció d'un canal.	
<*link*>	URL del lloc web HTML corresponent al canal.	
<*pubDate*>	Data de publicació del contingut del canal.	
<*rss*>	Identifica l'inici i el final del contingut RSS. És obligatori l'ús de l'atribut *version="2.0"*	*channel*
<*title*>	Una cadena de text identificadora del recurs. Quan s'utilitza amb un canal és el seu títol.	

Podem trobar l'especificació complerta a http://blogs.law.harvard.edu/tech/rss

5.4. El format Atom [b.24]

L' Atom 1.0 presentat l'agost del 2005, és un format de documents basat en XML que descriu llistes d'informació relacionada -que coneixem com *feeds* dels que hem parlat amb extensió en l'apartat 2-. Els *Feeds* es troben composats de *items* coneguts com *entries* (entrades) cadascuna amb un grup extensible de metadades adjuntades. Per exemple,cada entrada té un *title* (títol).

Existeixen dos classes de documents Atom: els *Atom Feed Documents* i els *Atom Entry Documents*.

Un *Atom Feed Document* és una representació d'un feed Atom,incloent metadades del feed i algunes o totes les entrades associades amb ell. La seva arrel és l'element **atom:feed**.

Un *Atom Entry Documents* representa exactament una *entry* Atom fora del context d'un feed Atom. La seva arrel és l'element **atom:entry**.

Els documents Atom han de ser han de ser XML ben formats i tenen l'extensió atom.

Elements	**Descripció**	**Sub-elements obligatoris**
<*atom:feed*>	És l'element arrel dels documents *Atom Feed Document*	*atom:author* (1 o més)* *atom:id* (exactament 1) *atom:title* (exactament 1) *atom:link* amb valor de l'atribut rel igual a "self" (exactament 1) *atom:update* (exactament 1)
*a menys que tots els elements fills atom:entry continguin al menys un element atom:author		
<*atom:entry*>	Representa una *entry* individual actuant com a contenidor per les metadades i dades associades a l'entrada. Pot aparèixer com a fill d'un element atom:feed,o com a element arrel d'un *Atom Entry Document*	*atom:author* (1 o més)* *atom:id* (exactament 1) *atom:content* (màxim 1)** *atom:summary* (exactament 1) *atom:title* (exactament 1) *atom:update* (exactament 1)
*a menys que contingui un element atom:source que contingui un element atom:author o, dins d'un Atom Feed Document,l'element atom:feed contingui ell mateix un element atom:author ** no és obligatori l'existència d'aquest element, si no hi és, llavors ha de tenir un element atom:link amb el valor de l'atribut rel igual a "alternate"		
<*atom:content*>	Conté o enllaça amb el contingut del *entry*.	
<*atom:author*>	Representa l'autor del *entry* o *feed.*	
<*atom:id*>	Expressa un identificador únic i permanent (International Resource Identifier IRI) per a un *entry* o *feed*	
<*atom:link*>	Defineix una referència a un recurs Web. Ha de tenir un atribut **href** el valor del qual ha de ser una referència IRI	
<*atom:summary*>	Text que proveeix d'un resum curt, abstracte, o un extracte d'un *entry*	
<*atom:title*>	Text que proporciona un títol llegible per humans de un *entry* o *feed*	
<*atom:update*>	Data que indica la més recent modificació d'un *entry* o un *feed* sempre i quan el *publisher* consideri aquesta modificació significativa d'alguna manera.	

Podem trobar l'especificació Atom a http://atom-enabled.org/

6. XQUERY

XQuery 1.0 és un llenguatge de consulta d'informació desada en XML. XQuery és producte d'alguns anys de treball per part d'individus i companyies al voltant de mon. Activament desenvolupat pel *W3C XML Query* i *XSL Working Groups* des del primer taller sobre *Query languages* en 1998 fins avui dia, està construït sobre les expressions **XPath**: comparteixen el mateix model de dades i suporten les mateixes funcions i operadors. Estudiarem XPath en l'apartat 5.1.

Tot i que actualment no és encara un estàndard,ho esdevindrà; llavors els desenvolupadors podran estar segurs de que el codi funcionarà entre productes diferents.

Només el temps ens dirà fins a quin punt d'èxit tindrà, però tot apunta que es podria convertir pel XML el que el *Structured Query Language* (SQL) s'ha convertit per les dades relacionals.

XQuery defineix centenars d'operadors i funcions. És un llenguatge ric per la construcció i navegació de documents XML.

Les característiques centrals de XQuery són:

- El seu sistema de tipus de valors,
- Els camins (path) de navegació, i
- Les expressions FLWOR

XQuery també permet als seus usuaris definir les seves pròpies funcions. [b.3]

6.1. XPath [b.18]

XPath és , des del novembre de 1999 ,un estàndard W3C . XPath defineix una sintaxi per accedir a parts d'un document XML.

El següent fragment de document XML que correspon a un element *entry* del format Atom ens servirà per exemplificar les característiques de XPath.

Exemple XML d'un element *entry* dins d'un *feed* en format Atom

```
<entry>
<title>The WebKit Open Source Project</title>
<link rel="alternate" type="text/html" href="http://feeds.feedburner.com/
oreilly/radar/atom?m=520" />
<id>http://webkit.opendarwin.org/</id>
<created>2005-10-13T08:34:02Z</created>
<modified>2005-07-03T09:06:11Z</modified>
<author>
<name>rael</name>
</author>
<dc:subject>apple</dc:subject>
<dc:subject>browser</dc:subject>
<summary type="text/html" mode="escaped">By rael WebKit, the
embeddable Web browser behind Mac OS X's Safari, Mail.app,
Dashboard, and third party apps like NetNewsWire and
TextMate....</summary>
</entry>
```

6.1.1. Nodes XPath

En XPath hi ha set classes de nodes: element, atribut, text, *namespace*, instrucció de processament, comentari i arrel. Els documents XML són tractats com arbres de nodes. L'arrel de l'arbre s'anomena node arrel.

Exemple de nodes en el document XML exemple:

<entry>	node arrel
<title>	node element
type="text/html"	node atribut

Valors atòmics

Els valors atòmics són nodes sense pares ni fills.

Exemple de valors atòmics:

http://webkit.opendarwin.org/
"alternate"

Ítems

Ítems són valors atòmics o nodes.

Relacions entre nodes

- **Pare** Cada element i atribut té un sol pare. En l'exemple anterior,l'element *entry* és el pare de *title***,** *link, id, created, modified, author, dc:subject i summary;* l'element *author* és, a la vegada, pare de *name.*
- **Fills** Els nodes element poden tenir zero, un o més fills. En l'exemple anterior de *title***,** *link, id, created, modified, author, dc:subject i summary* són tots fills de l'element *entry*. També observem l'element *name* que és fill de *author.*
- **Germans** Són els nodes que tenen el mateix pare. En l'exemple *title***,** *link, id, created, modified, author, dc:subject i summary* són tots germans.
- **Antecessors** El pare d'un node,el pare del pare, etc. En l'exemple anterior els antecessors de l'element *name* són els elements *author* i *entry*.
- **Descendents** El fill d'un node,el fill del fill, etc. En l'exemple anterior els descendents de l'element entry són els elements *title***,** *link, id, created, modified, author, dc:subject, summary i name .*

6.1.2. Sintaxi XPath

Seleccionar nodes

XPath utilitza expressions *path* (camins) per seleccionar els nodes o els grups de nodes dins d'un document XML. El node és seleccionat seguint el *path* o per passes.

Les expressions *path* més útils es troben llistades en la següent taula:

Expressió	Descripció
Nom del node	Selecciona tots els nodes fill del node
/	Selecciona des de el node arrel
//	Selecciona els nodes dins del document des del node actual que coincideixi amb la selecció, no importa on es trobin
.	Selecciona el node actual
..	Selecciona el pare del node actual
@	Selecciona atributs

En la taula següent hem llistat algunes expressions *path* i el resultat de les expressions:

Expressió *path*	Resultat
entry	Selecciona tots els nodes fills de l'element *entry*
/entry	Selecciona l'element arrel *entry*
entry /author	Selecciona tots els elements *author* que son fills de *entry*
//author	Selecciona tots els elements *author* sense que importi on es troben dins del document
entry //author	Selecciona tots els elements *author* que són descendents d'un element *entry*, no importa on es trobin per sota l'element *entry*
//@type	Selecciona tots els atributs que es diuen *type*

Predicats

Els predicats son usats per trobar un node específic o un node que conté un valor específic.

Els predicats sempre es troben dins de parèntesi quadrats.

En la taula següent hem llistat algunes expressions path amb predicats i el resultat d'aquestes expressions:

Expressió *path*	Resultat
/entry/author[1]	Selecciona el primer element *author* que és fill de l'element *entry*
/entry/author[last()]	Selecciona l'últim element *author* que és fill de l'element *entry*
/entry/author[last()-1]	Selecciona el penúltim element *author* que és fill de l'element *entry*
/entry/author[position()<3]	Selecciona els dos primers elements *author* que són fills de l'element *entry*
//link[@rel]	Selecciona tots els elements *link* que tenen un atribut anomenat *rel*

//link[@rel='alternate']	Selecciona tots els elements *link* que tenen un atribut anomenat *rel* el valor del qual és *alternate*

Seleccionant Nodes Desconeguts

Podem usar comodins XPath per seleccionar elements XML , XML els quals no coneixem la seva localització exacta dins de l'estructura jeràrquica del document.

Comodí	Descripció
*	Qualsevol node element
@*	Qualsevol node atribut
Node()	Qualsevol node de qualsevol classe

Exemples

Expressió *Path*	Resultat
/*entry*/*	Selecciona tots els nodes fills de l'element *entry*
//*	Selecciona tots els elements del document
//*ref*[@*]	Selecciona tots els elements *ref* que tenen un atribut

Seleccionant més d'un *Path*

Utilitzant l'operador | en una expressió XPath es poden seleccionar més d'un *path*.

Exemples

Expressió *Path*	Resultat
/*entry/title*	//*entry/link*	Selecciona tots els elements *link* i *title* de tots els elements *entry*
//*title*	//*link*	Selecciona tots els elements *title* i *link* del document
/*entry/author/name*	//*title*	Selecciona tots els elements *name* de l'element *author* de l'element *entry* i tots els elements *title* del document

6.1.3. Eixos XPath (*XPath Axes*)

Un eix (*axis*) defineix un grup de nodes relatius al node actual.

Nom de l'eix	Resultat
ancestor	Selecciona tots els antecessors del node actual
ancestor-or-self	Selecciona tots els antecessors del node actual i el node actual
atribut	Selecciona tots els atributs del node actual
child	Selecciona tots els fills del node actual
descendant	Selecciona tots els descendents del node actual
descendant-or-self	Selecciona tots els descendents del node actual i el node actual

following	Selecciona tot en el document després de la etiqueta de tancament del node actual
following-sibling	Selecciona tots els germans després del node actual
namespace	selecciona tots els nodes *namespaces* del node actual
parent	selecciona el pare del node actual
preceding	Selecciona tot en el document abans de la etiqueta d'inici del node actual
preceding-sibling	Selecciona tots els germans abans del node actual
self	selecciona el node actual

Expressions de p*ath* localitzat

Un p*ath* pot estar localitzat absolutament o relativament.

Un p*ath* absolut comença amb una barra invertida / i un relatiu no. En ambdós casos ens trobem en diversos passes separats per barres invertides

un *path* absolut	/pas1/pas2/...
un *path* relatiu	pas1/pas2

Un pas, que sempre s'avalua a partir dels nodes dins del grup de nodes actuals, consisteix en:

- Un eix (defineix les relacions d'arbre entre els nodes seleccionats i els nodes actuals)
- Un node (identifica un node dins d'un eix)
- Zero o més predicats (per concretar encara més el grup de nodes seleccionat)

La sintaxi per un pas és:

nomEix::node[predicat]

Exemples:

Exemple	**Resultat**
child::title	Selecciona tots els nodes *title* que son fills del node actual
attribute::ref	Selecciona l'atribut *ref* del node actual
*child::**	Selecciona tots els fills del node actual
*attribute::**	Selecciona tots els atributs del node actual
child::text()	Selecciona tots els nodes text fills del node actual
child::node()	Selecciona tots els nodes fill del node actual
descendant::title	Selecciona tots els descendents *title* del node actual
ancestor::title	Selecciona tots els antecessors *title* del node actual

child:: **/child::name*	selecciona tots els fills dels fills que siguin elements *name* del node actual

6.1.4. Operadors XPath

Una expressió XPath retorna o un grup de nodes,o un *String*, o un *Boolean* o un nombre.

A continuació tenim una llista dels operadors que poden ser usats en les expressions XPath:

Operador	Descripció	Exemple	Valor de retorn
|	Computa dos grups de nodes	*//author* | *//title*	Retorna un grup de nodes amb tots els elements *author* i *title*
+	Suma	6+4	10
-	Resta	6-4	2
*	Multiplicació	6*4	24
div	Divisió	8 div 2	2
=	Igual	preu=8.00	*true* si el preu és 8.00 *false* si el preu és 8.10
!=	No igual	preu!=8.00	*true* si el preu és 8.10 *false* si el preu és 8.00
<	Menys que	preu<8.00	*true* si el preu és 7.00 *false* si el preu és 8.00
<=	Menys o igual que	preu<=8.00	*true* si el preu és 7.00 *false* si el preu és 8.00
>	Més gran que	Preu>8.00	*true* si el preu és 9.00 *false* si el preu és 8.00
>=	Més gran que	Preu>8.00	*true* si el preu és 9.00 *false* si el preu és 8.00
or	o	Preu=8.00 or preu=8.50	*true* si el preu és 8.00 *false* si el preu és 9.00
and	i	Preu>8.00 or preu<8.50	*true* si el preu és 8.25 *false* si el preu és 9.00
mod	mòdul (resta de la divisió entera)	9 mod 5	4

6.1.5. Funcions predefinides

XPath disposa d'una llibreria de funcions predefinides.

Podem trobar aquestes funcions a

http://www.w3.org/2005/02/xpath-functions/

6.2. XQuery: l'exemple que utilitzarem

Utilitzarem un exemple d'un *feed* en format RSS, el document s'anomena *feed.xml* , per explicar les característiques de XQuery

feed.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<rss xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:content="http://purl.org/rss/1.0/modules/content/" version="2.0">
<channel>
<title>vilaine fille</title>
<link>http://www.vilainefille.com/vilaine_fille/</link>
<description>a proud member of the insanity-based community</description>
<language>en-US</language>
<lastBuildDate>Mon, 17 Apr 2006 00:01:00 -0400</lastBuildDate>
<generator>http://www.typepad.com/</generator>
<item>
<title>Vacances</title>
<link>http://www.vilainefille.com/vilaine_fille/2006/04/vacances.html</link>
<description>vilaine fille soon takes off to visit Lobo and his bipeds, her sisters K and A,
and her darling friend W. But for the pensée du dimanche and some almanac-style
postings to keep you amused, we go silent until...
</description>
<category>Sundry</category>
<dc:creator>vilaine fille</dc:creator>
<pubDate>Mon, 17 Apr 2006 00:01:00 -0400</pubDate>
</item>
<item>
<title>Pesach 5766</title>
<link>http://www.vilainefille.com/vilaine_fille/2006/04/pesach_5766.html</link>
<description>This was the month when the Jewish people were redeemed from Egypt, it
will be the month in the future when the final redemption will take place, and in every
Nissan there is the hope and expectation that something...
</description>
<category>Sundry</category>
<dc:creator>vilaine fille</dc:creator>
<pubDate>Wed, 12 Apr 2006 00:01:00 -0400</pubDate>
</item>
</channel>
</rss>
```

6.3. Expressions XQuery FLWOR [b.17]

XQuery utilitza les funcions, les expressions i els predicats de XPath.
Anem a veure una d'aquestes expressions X*Path*

```
Doc("feed.xml")/rss /channel/item/data(title)
```

L'expressió seleccionarà tots els elements *title* amb valor *Vacances* per sota dels elements *item* que es troben per sota de l'element *channel* que a la vegada es troba per sota de *rss*.

L'expressió FLWOR de continuació seleccionarà exactament el mateix que l'expressió *path* anterior:

```
for $x in doc("feed.xml")/rss /channel/item/title
where data(title) =Vacances
return $x/title
```

Amb FLWOR podem ordenar resultats:

```
for  $x in doc("feed.xml")/rss /channel/item
order by $x/title
return $x/title
```

El resultat serà:

```
<title>Pesach 5766</title>
<title>Vacances</title>
```

FLWOR és un acrònim per " *For, Let, Where, Order by, Return* ".
La clàusula ***for*** selecciona tots els elements *item* dins de l'element *channel* dins de *rss* dins d'una variable anomenada $x.
La clàusula ***let*** ens permet declarar variables i assignar-li un valor.
La clàusula ***where*** selecciona només els elements que compleixen alguna condició (en el primer cas que l'element *title* fos igual a *Vancances*)
La clàusula ***order by*** defineix la manera amb què hem d'ordenar el resultat. Ho farem per l'element *title*.
La clàusula ***return*** especifica què ha de ser retornat. Aquí ho seran els elements *title*.

6.4. XQuery FLWOR + HTML [b.17]

Anem a examinar la següent expressió XQuery FLWOR

```
for $x in doc("feed.xml")/rss /channel/item/title
order by $x
return $x
```

L'expressió anterior seleccionarà els elements *title* dels *item* dins dels elements *channel* que es troben dins de l'element *rss*, i els retornà en ordre alfabètic.
Si ara volguéssim llistar tots els items del nostre *feed* en una llista HTML, hauríem d'afegir les etiquetes <ul> i <li> a l'expressió FLWOR:

```
<ul>
        {
        for $x in doc("feed.xml")/rss /channel/item/title
        order by $x
        return <li>{$x}</li>
        }
</ul>
```

El resultat seria:

```
<ul>
<li><title>Pesach 5766</title></li>
<li><title>Vacances</title></li>
</ul>
```

Ara volem eliminar l'element title , i només mostrar les dades dins de l'element title:

```
<ul>
  {
    for $x in doc ("feed.xml")/rss /channel/item/title
    order by $x
    return <li>{data($x)}</li>
  }
</ul>
```

El resultat serà una llista HTML:

```
<ul>
<li>Pesach 5766</li>
<li>Vacances</li>
</ul
```

6.5. Terminologia XQuery

Tots el termes i conceptes relacionats amb els nodes i les relacions entre aquests són exactament els mateixos que els descrits en l'apartat 5.1.1 sobre XPath.

6.6. Sintaxi XQuery [b.17]

6.6.1. Regles de sintaxi bàsiques

- XQuery diferencia entre majúsculs i minúscules
- Els elements, atributs i variables han de ser noms XML vàlids
- Un valor *String* XQuery pot portar cometes simples o dobles
- Una variable XQuery es defineix amb un $ seguit d'un nom
- Els comentaris es troben delimitats per : i :

6.6.2. Les expressions XQuery condicionals

Les expressions "*If-Then-Else*" són permeses en XQuery.

```
for $x in doc("feed.xml")/rss /channel/item
return if ($x/data(category)=Sundry)
then <sundry>{data($x/title)}</sundry>
else {data($x/title)}
```

Fem notar que la sentència *else* és necessària, però pot ser *else().*
El resultat de l'exemple serà:

```
<sundry>Pesach 5766</sundry>
<sundry>Vacances</sundry>
```

6.6.3. Comparacions XQuery

En XQuery hi ha dues maneres de comparar valors

1. comparacions generals: =, !=, <, <=, >, >=
2. comparacions per valor: eq, ne, lt, le, gt, ge

La diferència entre els dos mètodes de comparacions són els següents:

```
$rss /channel/item @q > 10
```

L'expressió anterior retorna **verdader** si algun dels atributs q té un valor més gran que 10

```
$rss /channel/item @q gt  10
```

L'expressió anterior retorna **verdader** si hi ha només un atribut q que retorna l'expressió, i el seu valor és més gran que 10.Si més d'un q és retornat, succeeix un error.

6.7. Afegint Elements i Atributs [b.17]

6.7.1. Afegint elements HTML i Text

A continuació veurem un exemple de com afegir alguns elements HTML al resultat d'una consulta XQuery. Posarem aquest resultat en una llista HTML:

```
<html>
<body>
<h1>feed </h1>
<ul>
{
for $x in doc("feed.xml")/rss /channel/item
order by $x/title
return <li>{data($x/title)}. Tipus {data($x/@category)}</li>
}
</ul>
</body>
</html>
```

L'expressió XQuery anterior generarà el resultat següent:

```
<html>
<body>
<h1>feed</h1>
<ul>
<li>Pesach 5766.Tipus:sundry </li>
<li>Vacances. Tipus:sundry </li>
</ul>
</body>
</html>
```

6.7.2. Afegint atributs als elements HTML

Ara veurem com utilitzar l'atribut tipus com a atribut classe (class) en una llista HTML

```
<html>
<body>
<h1>feed</h1>
<ul>
 {
   for $x in doc("feed.xml")/rss /channel/item
   order by $x/title
   return
   <li class="{data($x/@category)}">{data($x/title)}
   </li>
  }
</ul>
</body>
</html>
```

L'expressió XQuery anterior generarà el següent resultat:

```
<html>
<body>
<h1>feed </h1>
<ul>
<li class="sundry">Pesach 5766 </li>
<li class="sundry">Vacances </li>
</ul>
</body>
</html>
```

7. BASES DE DADES XML NATIVES .EXIST

7.1. Introducció

Alguns autors defineixen un sistema de base de dades XML nativa només per la seva interfície amb l'usuari

-"Defineix un model [lògic] per un document XML... i emmagatzema i recupera documents segons aquest model... per exemple,pot ser construït damunt una base de dades relacional,jeràrquica, o orientada a objectes..." [b.1] http://www.rpbourret.com/xml/XMLAndDatabases.htm -,

algunes companyies porten aquest terme una mica més lluny fent que una base de dades XML nativa sigui construïda i dissenyada per la gestió de XML, i no només un sistema de base de dades per un model arbitrari de dades amb un estrat XML damunt.

A primera vista, això no com porta cap diferència a nivell de l'usuari, però una diferència fonamental es troba dins del sistema. XML és diferent d'altres models de dades ben coneguts (per exemple al relacional o l'orientat a objectes) en molts aspectes. Com a resultat d'això, la transformació de XML a un altre model de dades sempre origina "imperfeccions" , que porten a limitacions en funcionalitat i/o execució.

7.2. Un exemple concret: eXist [b.28]

eXist (http://exist.sourceforge.net/)es un esforç de la comunitat *Open Source* per desenvolupar un sistema de base de dades XML natiu (BD:XML), fortament integrat amb eines XML de desenvolupament existents, tals com Cocoon d'Apache. eXist cobreix la majoria de les característiques de les BD:XML natives així com altres tècniques avançades com la cerca de paraules clau en el text, consultes sobre la proximitat de termes i cerques de patrons basades en expressions regulars. La base de dades està completament escrita en Java i pot ser fàcilment instal·lada i executada com una aplicació dins d'un contenidor de *servlets* o directament incrustada dins d'una altre aplicació. Aquests dos últims casos és el que es coneix com una base de dades *embedded*: la base de dades no requereix un servidor separat, ja que tota la lògica de gestió de les dades està enllaçada dins l'espai d'adreces de l'aplicació. Generalment ofereixen una API de mètodes de classes o de crides de funcions que els desenvolupadors poden utilitzar per emmagatzemar, actualitzar o consultar registres. Les base de dades *embedded*, com a regla, no inclouen eines de consulta per usuaris finals.

Fonamentalment, són sistemes que s'utilitzen per la construcció d'aplicacions o de servidors que necessiten velocitat o fiabilitat en la gestió de les dades. L'única manera de guardar i treure dades és escriure codi que cridi al Api de la base de dades.

A continuació donarem algunes de les característiques clau de eXist.

7.2.1. Emmagatzemant dades XML sense validar

No es requereix que els documents tinguin associat un *schema* o un DTD. És a dir només cal que estiguin ben formats.

7.2.2. Col·leccions

Dins de la base de dades, els documents són gestionats en col·leccions jeràrquiques. Des del punt de vista d'un usuari, això és comparable a emmagatzemar arxius en un sistema d'arxius. Les col·leccions poden ser encapsulades arbitràriament. No estan lligades a un esquema

predeterminat, així el nombre de tipus de documents usats pels documents dins d'una col·lecció no està restringit. Dins d'una col·lecció podem barrejar document arbitraris.

Utilitzant XPath sintaxi amb extensions podem fer consultes de qualsevol part de la jerarquia o inclús de tots els documents dins la base de dades.

7.2.3. Processament de consultes basades en índexs

Per millorar la velocitat de processament de les consultes, eXist utilitza un *scheme* d'indexació numèrica per identificar els nodes XML en l'índex. La indexació s'aplica a tots els nodes del document ,incloent elements, atributs, text i comentaris. Contràriament a altres tipus d'implementacions , no és necessari crear índexs explícitament. Tots els índexs són gestionats pel motor de la base de dades.,De totes maneres, és possible restringir la indexació automàtica de tot el text a parts definides d'un document.

7.2.4. Extensions per a cerques textuals complertes

eXist proporciona un motor optimitzat de XQuery per processar eficientment consultes textuals complertes .Una estructura addicional d'índexs manté una rastre de ocurrències de paraules i assisteix al usuari en les consultes de contingut textual. Aquest tipus de consultes no són ben suportades per XPath.

7.2.5. XML estàndards

- XPath 2.0 i XQuery 1.0

7.3. Desenvolupament d'aplicacions

Les aplicacions poden accedir al servidor remot d'eXist a través de HTTP, XML *Remote Procedure Call* (XML-RPC), SOAP i de les interfícies *Web-based Distributed Authoring and Versioning* (WebDAV) .Els desenvolupadors que programen en Java poden utilitzar l'API XML:DB que eXist suporta: és una API proposada per *XML:BD initiative* que intenta estandarditzar interfícies de desenvolupament comunes d'accés als serveis de les bases de dades XML. Utilitzant el XML:DB API, es pot mantenir el codi sense canvis encara que intercanviïs de sistemes de bases de dades com Xindice (http://xml.apache.org/xindice/) i eXist. Parlarem d'aquesta API a l'apartat següent.

7.3.1. Programant aplicacions Java amb l'API XML:DB

La manera més senzilla per accedir a eXist des de les aplicacions Java és usar l'API XML:DB.

L'API XML:DB proporciona una interfície comuna per a les base de dades XML natives o *enabled* i suporta el desenvolupament d'aplicacions portables i reusables.

És un intent per provar d'estandarditzar un API comú per accedir als serveis de les bases de dades XML,comparables als JDBC o al *Open Database Connectivity* (ODBC) dels sistemes relacionals. L'API es troba construït al voltant de 4 conceptes centrals: *drivers*, col·leccions, recursos i serveis. Els *drivers* encapsulen tota la lògica d'accés a la base de dades. Són proporcionats pel fabricant de la BD i han de ser

registrats dins del gestor de la base de dades. Com en eXist, les col·leccions són contenidors jeràrquics, contenen altres col·lecions o recursos. Un recurs pot ser o bé un recurs XML o un objecte binari. Altres tipus de recursos poden ser afegits en futures versions de l'estàndard. Actualment eXist només suporta recursos XML. Finalment, els serveis poden ser sol·licitats per efectuar tasques especials com consultes a col·leccions amb XPath o gestionar les col·leccions.

Cada aplicació utilitzant l'API XML:DB ha d'obtenir primer un objecte col·lecció de la BD. Això es realitza fent una crida al mètode *static* **getCollection** de la classe **DatabaseManager.** Per localitzar la col·lecció específica el mètode espera una URI complerta com a paràmetre,la qual identifica la implementació de la base de dades, el nom de la col·lecció i opcionalment la localització del servidor de la BD a la xarxa.

Per exemple, la URI "xmldb:exist:///db/UOC" referència la col·lecció UOC d'una base de dades eXist que s'està executant en mode *embedded*. Internament eXist té dues implementacions de *driver* diferent:la primera parla amb un motor de base de dades utilitzant les crides XML-RPC; la segona té accés directe amb la instància local d'eXist. Quina implementació és seleccionada depèn de la URI passada al mètode **getCollection**. Per referenciar la col·lecció UOC en un servidor remot,per exemple Tomcat Web, usarem "xmldb:exist://localhost:8080/exist/xmlrpc/db/UOC".

El **DriverManager** manté una llista dels *drivers* disponibles i utilitza la ID de la base de dades especificada en la URI ("exist") per seleccionar la classe del *driver* correcta. Els *drivers* poden ser registrats per diferents bases de dades cridant el mètode **DatabaseManager.registerDatabase** al principi del programa. Anem a mostrar un exemple d'un fragment de codi que registra un *driver* per eXist

```
Class cl= Class.forname("org.exist.xmldb.DatabaseImpl");
Database driver=(database)cl.newInstance();
Databasemanager.registerDatabase(driver);
```

7.3.2. Accés a eXist mitjançat el conjunt de classes PHEXIST

PheXist és un conjunt de classes per fer consultes a la base de dades eXist.

Aquest conjunt de classes ,implemetades tant en PHP com en Perl, permeten fer consultes a la base de dades XML nativa utilitzant el llenguatge XQuery.

La connexió amb la XML:BD es realitza mitjançant la interfície SOAP.

Mètodes que podem utilitzar:

- **Constructor**(string User, string Password, string WSDL) : crea un nou client proxy SOAP.
- **connect**() : realitza la connexió amb la eXist XML:DB.
- **disconnect**() : desconnecta la sessió actual.
- **xquery**(string Query) : consulta la XML:DB utilitzant el llenguatge XQuery.

- **getError**() : torna una cadena de caràcters d'error (només implementat en PHP).
- **debug**(boolean Debug?) : per depurar les comunicacions entre la classe i el servidor, així com les sortides dels resultats XQuery (només implementat en PHP).

8. CONCLUSIONS

Fins ara les URI, per poder identificar els recursos, el llenguatge HTML i el protocol HTTP han estat els mecanismes per disposar la informació a la xarxa.

La web, tal i com es troba definida actualment (és a dir, amb els mecanismes mencionats), presenta seriosos problemes relacionats amb les possibilitats de gestió del coneixement: com trobem, extraiem, representem, interpretem i mantenim la informació a la web?

Una de les iniciatives per tractar de millorar aquests aspectes han estat els formats RSS i Atom.

RSS i Atom han permès el que coneixem com a sindicació del contingut web.

La sindicació web ofereix actualment moltes possibilitats: els weblogs ,el podcasting, l'agregació a publicacions de notícies.... en pocs anys s'ha convertit en una tecnologia molt utilitzada a la web i que s'està aplicant a més i nous camps, com per exemple al comerç electrònic.

Però la sindicació web és només el inici d'una veritable redefinició de la web tal i com la coneixem avui dia.

El W3C està portant a terme recerques per portar a terme el que ja es coneix com la web semàntica. Permetrà una usabilitat i aprofitament de la web, així com dels recursos que ofereix. Es tracta de fer accessible i que es pugui gestionar el contingut web sense intervenció humana, mitjançant agents automàtics.

Veritables cerques semàntiques no trigaran gaire en ser possibles, una millor gestió de la informació de la web i ,segurament una mica més tard ,el comerç electrònic mitjançant agents de software... qui sap quines coses més ens podrà oferir la web semàntica?

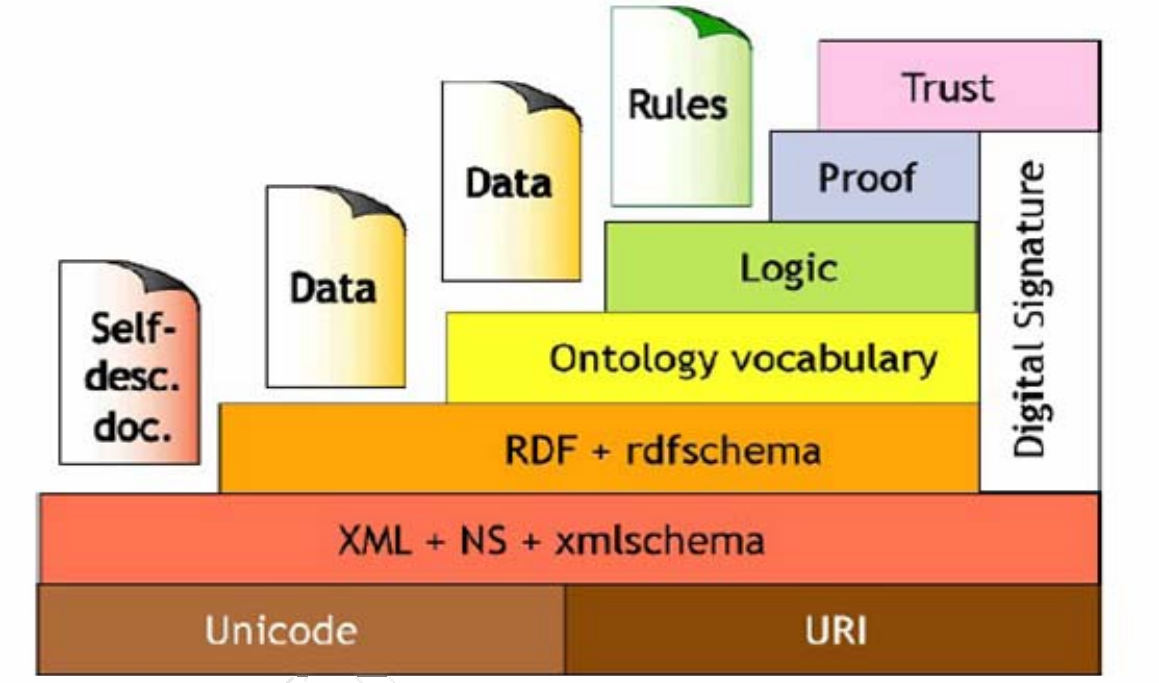
9. LÍNIES DE FUTUR

Aquesta memòria tracta abastament del fenomen de la sindicació web i, especialment, dels formats que la possibiliten, RSS i Atom.

RSS i Atom són formats de la família XML, del que també hem estudiat les seves característiques ,així com del seu llenguatge de consulta el XQuery.

XML és la base sintàctica de la web semàntica, que la formen diferents nivells com el *Resource Description Framework* (que defineix una infraestructura per a la descripció de recursos) o el *Web Ontology Language* (OWL): un llenguatge per definir ontologies a l'entorn Web.

Veiem aquí un esquema que ens mostra aquests nivells.



La web semàntica, que és el punt de partida d'aquest treball, és també el punt cap on es dirigeix: l'estudi dels conceptes relacionats amb la web semàntica, els llenguatges que trobem definits, el diferents nivells que la construeixen i quin és el seu significat...

Els molts reptes i noves tecnologies que ens aportarà els avenços en aquesta matèria són temes apassionants que han quedat fora d'aquest estudi, però que, sens dubte, serien la línia de treball a seguir del mateix.

L'aplicació desenvolupada, la web didàctica –limitada als formats RSS i Atom- seria enriquida amb tot aquest nou coneixement: una web que posaria a disposició de l'usuari la informació sobre la web semàntica i que seria actualitzada amb les novetats i nous coneixements sobre el tema.

És clar, aquesta web disposaria d'un *feed* perquè s'hi pogués agregar l'usuari visitant.

10.GLOSSARI

Atom: Atom és un format XML similar a RSS, és a dir, de sindicació. Va nèixer per resoldre la confusió creada per l'existència d'estàndars similars per a la sindicació (RSS i RDF) i crear una API i un format de sindicació més flexibles. No obstant això, hi ha qui opina que més que resoldre el problema, conviu amb els formats anteriors que volia eliminar. Encara està en desenvolupament i ha rebut diversos noms, entre ells Echo, però finalment va anomenar-se Atom. [b.24]

DTD/XML-Schema: Un document XML pot tenir associat un DTD o un XMLSchema. Aquests estableixen regles que ha de complir el document XML. Les dues diferències més importants entre un DTD o un XML-Schema és que l'últim permet de definir de forma més precisa les regles del XML i que l'XML-Schema és a la seva vegada un document XML. [b.16]

***eXtensible Markup Language* (XML):** L'XML, llenguatge d'etiquetatge extensible (*eXtensible Markup Language* en anglès) és un llenguatge informàtic d'etiquetatge que deriva del llenguatge SGML i permet representar i intercanviar informació entre ordinadors o programari, atès que organitza les dades de manera ordenada. [b.12]

RSS: El RSS és un format d'arxiu de la família XML desenvolupat específicament per a llocs de notícies i weblogs que s'actualitzen amb freqüència i per mitjà del qual es pot compartir la informació i usar-la en altres llocs web o programes. És en essència una "redifusió" de continguts. Per influència de la llengua anglesa s'acostuma a referir-s'hi utilitzant el barbarisme sindicació.

L'acrònim s'usa per als següents sistemes de comunicació estàndard:

- Rich Site Summary (RSS 0.91)
- RDF Site Summary (RSS 0.9 and 1.0)
- Really Simple Syndication (RSS 2.0)

Els programes que llegeixen i presenten fonts RSS de diferents procedències es denominen agregadors. [b.25]

SOAP: SOAP es un protocol dissenyat per intercanviar missatges en format XML en una xarxa d'ordinadors, normalment sobre el protocol HTTP. Principalment, **SOAP** és un protocol que es fa servir per accedir Serveis web.[b.27]

Weblog: Un *weblog* és un espai personal d'escriptura a Internet. Tot i les diverses formes d'anomenar-lo (weblog, blog, dip o bitàcola), la paraula correcta en català és bloc. Es pot definir com un diari *online*, un lloc web que una persona utilitza per a escriure periòdicament, en el qual tota l'escriptura i l'estil s'utilitza via web. Un weblog està dissenyat per a que, com a un diari, cada article (*post*) tingui data de publicació, de tal forma que la persona que escriu (*weblogger*) i les que llegeixen poden seguir un camí de tot el que s'ha publicat i editat.

Web semàntica: És un projecte que té com a objectiu crear un medi universal per al intercanvi d'informació significativa (semàntica), d'una forma comprensible per a les màquines, del contingut dels documents de la Web. Amb això es pretén ampliar la interoperabilitat dels sistemes informàtics i reduir la mediació dels operadors humans en els processos intel·ligents de flux d'informació. [b.5]

11.BIBLIOGRAFIA

b.1 Chaudri, Akmal B. ; Rashid, Awais ; Zicari, Roberto (2003)**.** *XML Data Management: Native XML and XML-Enabled Database Systems*. Ed. Addison-Wesley.

b.2 Ray, E.T (2001). *Learning XML.* O'Reilly.

b.3 Michael Brundage (2004) XQuery , *The XML Query Language*.

b.4 Shelley Powers (2005) *What are a Syndication Feeds* 0'Reilly Media

Recursos web:

- Web semàntica

b.5 http://www.w3.org/2001/sw/

b.6 G. Antoniou; F van Harmelen (2004) *A Semantic Web Primer*. Ed. The MIT Press.

b.7 www.ics.forth.gr/isl/swprimer/

- Aplicacions web

b.8 http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web

- Sindicació web

b.9 http://en.wikipedia.org/wiki/Web_syndication

b.10 http://en.wikipedia.org/wiki/Web_feed

- Blogs

b.11http://en.wikipedia.org/wiki/Weblog

- XML

b.12 http://www.w3.org/XML/

b.13 http://dat.etsit.upm.es/~abarbero/curso/xml/xmltutorial.html

b.14 http://en.wikipedia.org/wiki/XML

b.15 http://www.w3.org/XML/1999/XML-in-10-points.es.html

- XQuery

b.16 http://en.wikipedia.org/wiki/XQuery

b.17 http://www.w3schools.com/xquery/default.asp

- XPath

b.18 http://www.w3schools.com/xpath/default.asp

- formats RSS i Atom

b.19 http://en.wikipedia.org/wiki/Really_Simple_Syndication

b.20 http://www.w3schools.com/xquery/default.asp

b.21 http://my.netscape.com/publish/formats/rss-spec-0.91.html

b.22 http://web.resource.org/rss/1.0/

b.23 http://blogs.law.harvard.edu/tech/rss

b.24 http://atom-enabled.org/

b.25 http://ca.wikipedia.org/wiki/RSS

- World Wide Web Consortium

b.26 http://ca.wikipedia.org/wiki/World_Wide_Web_Consortium

- Soap

b.27 http://ca.wikipedia.org/wiki/SOAP

- eXist

b.28 http://www.exist-db.org/facts.html