

Estudi del model de representació XML/RDF.

Alumne: Pascual Sorando Pinilla
ETIG / ETIS

Consultor: Carlos Granell Canut

Gener 2004

Estudi del model de representació XML/RDF.

Objectius.

L'objectiu fonamental d'aquest treball és assolir les relacions que existeixen a nivell teòric entre XML (*eXtensible Markup Language*) i RDF (*Resource Description Language*).

Exposar el tractament de recursos en la xarxa i les limitacions de HTML.

Plantejar la necessitat de la descripció i registre de documents per al seu tractament mecànic.

Explicar les següents tecnologies de descripció de documents desenvolupades per W3C:

1. RDF presentat amb grafs.
2. RDF presentat amb tuples (subjecte-predicat-objecte).
3. Les DTD de XML
4. XML *Schema*
5. RDF/XML

Metodologia.

Trobar fonts de informació. El treball pretén presentar una recopilació de la informació referent a XML i RDF disponible a Internet i als llibres.

Presentació de resultats al tutor.

Corregir errors.

(Aquesta metodologia de treball amb documents que cal reeditar amb totes les correccions necessàries és la mateixa que aplica el W3C per a presentar les seves aportacions a la xarxa.)

Índex de Continguts

1. Introducció.
 - 1.0 Objectius del capítol.
 - 1.1 Situació actual de la xarxa Internet. (La informació, els organismes i les tecnologies).
 - 1.1.1 La informació
 - 1.1.1.1 Estructura
 - 1.1.1.2 Contingut
 - 1.1.2 Els organismes
 - 1.1.2.1 W3C
 - 1.1.3 Les tecnologies.
 - 1.2 Sintaxis de les URI.
 - 1.3 Necessitat de la semàntica.
 - 1.4 El model XML/RDF
 - 1.4.1 Justificació
 - 1.4.2 Els recursos
 - 1.4.3 La descripció de recursos i el registre.

2. XML
 - 2.0 Objectius del capítol.
 - 2.1. La sintaxis XML i els documents XML.
 - 2.1.1. La sintaxis bàsica
 - 2.1.2. Els DTD. – Declaració Tipus Document.
 - 2.1.3. Els documents XML.
 - 2.2. XML Schema i XML Namespace.
 - 2.2.1. XML Namespace
 - 2.2.2. XML Schema
 - 2.2.2.1. El document xsd
 - 2.2.2.2. El document xml
 - 2.3. Avantatges i limitacions de XML.

3. RDF – Resource Description Framework.
 - 3.0 Objectius del capítol.
 - 3.1- Model semàntic. RDF i la Informació.
 - 3.1.1.- Les adreces dels recursos.
 - 3.1.2.- Generació de sentències.
 - 3.1.3.- Els vocabularis RDF.
 - 3.2.- Les formes de representació de sentències RDF.
 - 3.2.1.- La representació gràfica.
 - 3.2.2.- La representació amb t-uples.
 - 3.2.3.- La representació amb text XML (RDF/XML).
 - 3.3- Propietats fonamentals de RDF.

4. Els documents RDF/XML
 - 4.1 Relacions entre XML i RDF. ¿com aprofita RDF el llenguatge XML?
 - 4.2. La representació amb llenguatge XML .Els elements xml
 - 4.2.1 Els elements de xmlns:rdf

- 4.2.2 Els elements de xmlns:rdfs
- 4.2.3 Els elements de xmlns:xsd
- 4.2.4 Elements definits en altres documents : Els elements de xmlns:dc
- 4.3 Procediment per crear documents RDF/XML.
- 4.3.1 Simplificacions i anidament d'elements.
- 4.4 Gramàtica RDF/XML.

- 5. Glossari.
- 6. Bibliografia.

Resum
Summary

Capítol 1. Introducció

En línies generals, XML i RDF són dues tecnologies complementaries utilitzades per fer Internet molt més intel·ligent.

En primer lloc, es tracta de realitzar un estudi sobre aquestes dues tecnologies, per a arribar a comprendre en què consisteixen i descriure les seves limitacions i avantatges. Després, s'analitzen les diferències entre el model sintàctic que proporciona XML i el model semàntic RDF.

Una vegada tractats els temes introductoris, cal investigar com es possible relacionar-les, és a dir, s'arriba al concepte de “*serialization*”. En general, aquest concepte es refereix a la representació d'un origen en un altre format que l'original per emmagatzemar-lo a memòria, transportar-lo o qualsevol altre motiu. Cal tenir en compte que RDF és una capa abstracta sota XML i existeixen moltes maneres d'emmagatzemar o transportar RDF. Es tracta, doncs de profunditzar en XML com a representació de RDF, tenint en compte que la sintaxi XML és una de les possibles sintaxis per RDF. També convé saber que la combinació dels dos, models representats en RDF i XML, s'anomena representació XML/RDF.

1.0. Objectius del capítol

Es pretén fer una presentació d'alguns aspectes del model HTML d'Internet que poden ser interessants des de el punt de vista de la **descripció de recursos** amb RDF, en particular el mecanisme d'identificació (URL).

No s'entrarà en la descripció dels elements dels llenguatges de marca (SGML i HTML) encara que es necessari conèixer-los per entendre les seves limitacions.

1.1. Situació actual de la xarxa Internet. (La informació, els organismes i les tecnologies).

1.1.1. La informació

Internet va nàixer com un mitjà de transferència d'informació entre dos interlocutors. Avui dia ha evolucionat i s'ha complicat.

Internet permet la comunicació entre desconeguts, amb aplicació d'una determinada tecnologia, mes ben dit, permet la difusió de dades entre una persona a l'origen i una altra a la destinació. Cal suposar que les dades es transformaran en informació i en coneixement.

1.1.1.1. Estructura de la informació

Si considerem la informació des de un punt de vista tècnic, podem veure que té una determinada estructura lògica i física i que utilitza uns determinats suports materials.

L'estructura física és el format. Normalment es presenta en format textual amb ajuda de multitud d'imatges.

D'entrada, es pot afirmar que durant un temps l'interès ha estat acaparat per la presentació de la informació i que el contingut ha estat deixat en segon pla. I encara que la xarxa no seria possible sense el suport tècnic, és necessari ser conscients que la veritable importància de la xarxa rau al processament racional de la informació.

HTML és la base de l'estructura de la informació actual a Internet.

A la xarxa es poden trobar molts tipus de fitxers, de text (.txt, .doc, .html, .htm, .pdf), sons (.wav, .midi), d'imatges estàtiques (.bmp, .gif, .mpg) o dinàmiques (.avi), però la xarxa basa la seva importància en els documents textuais (en particular en llenguatges de marca com és el cas de html i xml derivats del sgml).

Les pàgines web amb html permeten integrar tot tipus d'informació, motiu pel qual, aparentment, es configura com el suport documental ideal per als documents interactius.

Una pàgina web té una estructura superficial molt simple, però de fet, és força complexa si la comparem amb els mitjans tradicionals d'impressió de documents en paper.

Una pàgina HTML és un suport dinàmic, amb el contingut i el format separables, que permet fer modificacions amb molt poc esforç. En ocasions el document ni tan sols existeix, i és el propi client, qui sense saber-lo genera el contingut i la composició, amb l'ajut d'una base de dades. (CGI)

La pàgina web és un fitxer només de text. Les imatges no es troben a la pàgina. (Cal dir que la informació de les imatges no és accessible als buscadors.)

La pàgina i les imatges son considerades **recursos** perquè tenen el seu propi URL.

Un element molt important de l'estructura de la informació és la seva localització. Qualsevol informació té associat un **URI** (identificador uniform de recurs – Uniform Resource Identifier) , es a dir, una adreça.

Ver RFC 2396

Un recurs es defineix com qualsevol element amb una adreça URI.

Tots els noms d'adreces formen part d'una estructura jeràrquica de dominis de la forma

Subdomini2.subdomini1.domini.

El nivell superior dels dominis pot fer referència a una determinada classe d'organització (.com, .edu, .gov, .mil, .org, .net) o a un país. Aquesta jerarquia és la que garanteix que cap nom no està repetit.

La forma general d'una URI (URI absoluta) és

Esquema: identificador

Mentre que les URI relatives son únicament l'identificador.

Les URI poden ser localitzadors (URL) si especifiquen com accedir al recurs, i un nom (URN) si identifica el recurs amb un conjunt d'atributs.

Segons l'**esquema**, aquestes són algunes de les sintaxis possibles dels URL:

<ftp://direcció-servidor: directori>

[ftp://\[usuari\[:contrasenya\]@\]servidor/camí](ftp://[usuari[:contrasenya]@]servidor/camí)

<News:identificador>

(identificador pot ser un grup de notícies Usenet o un identificador de missatge)

<mailto:adreça>

[http://servidor\[:port\]/camí ;paràmetre ?consulta #fragment](http://servidor[:port]/camí ;paràmetre ?consulta #fragment)

El port només és necessari quan la informació no es troba disponible al port que es fa servir per defecte per al servei.

Exemples de recursos:

<ftp://mrcnext.cso.uiuc.edu/>

<ftp://ftp.uoc.es/pub/doc/README>

<news:soc.culture.spain>

<news:comp.infosystems.www.misc>

<telnet://wiretap.spies.com>

<mailto:U.N.Estudiant@campus.uoc.es>

<http://www.acme.com/buscador/busca.cgi?nom=Internet>

<http://www.acme.com/doc/ajuda.html#buscador>

Exemple de recurs amb URI relativa

Ajuda.html#buscador

Exemple de transformació de URI relativa a URI absoluta

URI base:

<http://www.uoc.edu/extern/ct/home/home.html>

URIs relatives

URIs absolutes

/accesset98/

http://www.uoc.edu/accesset98/

altres/pagina.html

http://www.uoc.edu/extern/ct/home/altres/pagina.html

../../logo.jpeg

http://www.uoc.edu/extern/logo.jpeg

#final

http://www.uoc.edu/extern/ct/home/home.html#final

Les URIs relatives s'interpreten en funció de l'adreça base, conservant la jerarquia que marquen les barres.

Funcionament d'aquesta estructura.

Quan l'usuari escriu en el seu navegador una direcció web (URL), el navegador crea una connexió amb el servidor i pregunta per la pàgina especificada.

El servidor envia una còpia del fitxer en qüestió a l'ordinador de l'usuari. Quan el navegador rep el fitxer, l'analitza i comença a presentar la informació.

Amb la informació textual no tindrà cap problema. Però per aconseguir les imatges, cal iniciar noves connexions.

El hipertext té un funcionament semblant. Clicar un enllaç té el mateix efecte que sol·licitar una pàgina (introduir un URL) al navegador. En aquest cas és un URL predeterminat.

Cal suposar que es tracta d'una estructura estable i permanent (encara que no es així).

Hi ha altres mètodes per saltar a una altra pàgina com ara el mètode GET d'un formulari o comandes de JavaScript.

Un mecanisme diferent és la consulta de bases de dades. El client sol·licita una pàgina web (una URI) al servidor, i el servidor (després de consultar una base de dades) genera i retransmet la pàgina. La pàgina no existeix realment en el servidor, es tracta de la resposta d'una aplicació tipus CGI.

1.1.1.2. Els continguts

Els documents de Internet poden ser llegits de la mateixa forma que un llibre, (una seqüència de dades lineals), però fonamentalment estan preparats per a ser aprofitats com elements hipertextuals (dades no lineals).

Les dades de Internet són molt heterogènies. L'anàlisi de les dades ens pot permetre extreure informació o determinar que un recurs és un document de valor. I finalment es pot esperar que la informació es convertirà en coneixement.

Es obligat apuntar que Internet no es va crear en un principi per atendre la publicació i recuperació organitzada d'informació motiu pel qual en l'actualitat és una gran quantitat d'informació textual indefinida i poc estructurada.
La recuperació d'informació provoca en els usuaris, de vegades desbordament i de vegades desorientació degut a canvis de context.

HTML aporta molt poc ajut per a la recuperació d'informació. Només algunes metadades (*meta keywords* = paraules clau). Aquesta carència s'ha intentat superar amb tecnologia de cerca i recuperació de documents basada en mètodes de "força bruta".

La gran virtut de la xarxa de ser un entorn obert, heterogeni i canviant, sense límits a les variacions que pugui patir un document, per canvis d'organització, canvis de presentació o canvis de contingut, produeix un resultat desconcertant. Es pot trobar variabilitat espacial (canvi d'ubicació) i variabilitat temporal (temps de vida del document especialment curts).

1.1.2. Els organismes

Organització de Internet.

Internet es una agrupació de persones i organitzacions al voltant d'un objectiu comú: Desenvolupar tecnologies que permetin millorar el funcionament de la xarxa. Hi ha molts organismes públics, d'universitats i privats treballant, com ara

- CERN Centre Europeu d'Investigació Nuclear
- NCSA National Center for Supercomputing Applications (U.Illinois)
- NIC Network Information Center
- NSF National Science Foundation
- W3C World Wide Web Consortium

1.1.2.1. W3C : World Wide Web Consortium

El W3C (Consorti per a la World Wide Web) fou fundat a l'octubre de 1994.
És un consorci industrial internacional constituït per Organitzacions membres, sense ànim de lucre, que treballen amb la comunitat internacional per desenvolupar **especificacions** i programes informàtics de referència, que es distribueixen gratuïtament.

W3C crea els estàndards per a la Web. Desenvolupa especificacions, directrius, software, i eines. Els seus principis operatius són: Accés universal (internacionalització, independència de dispositiu, accessibilitat), Web semàntica (RDF, XML, XML Schema, XML Signatures), confiança (responsabilitat de les publicacions), software obert, evolutiu, descentralitzat.

Les limitacions de HTML 4.0 han portat a W3C a crear un nou metallenguatge, el XML, que aprofita els avantatges de HTML .

XML Working Groups.

W3C està organitzat en grups de treball:

- I. XML Coordination Group
- II. XML Core Working Group: és el encarregat de desenvolupar i mantenir les especificacions del XML, Namespaces, XML Information Set i Xinclude.
- III. XSL Working Group: és responsable del XSL – Extensible Stylesheet Language, amb inclusió de XSLT - XSL Transformations y XSL/FO – XSL Formatting Objects.
- IV. XML Linking Working Group : va treballar en Xlink – XML Linking Language i Xpointer – XML Pointer Language.
- V. XML Query Working Group : està treballant en XML Query Language, jper a facilitar la recuperació de dades de documents XML de la xarxa. (Xquery i Xpath).
- VI. XML Schema Working Group : W3C XML Schemas ofereix mecanismes de definició i descripció de estructura, contingut i en certa mesura semàntica dels documents XML.

Especificacions:

La pàgina del XML Core WG ens parla dels següents documents:

- Extensible Markup Language XML 1.0 (Second Edition)
- XML 1.1
- Namespaces in XML
- Namespaces in XML 1.1
- XML Inclusions (XInclude)
- XML Information Set
- XML Fragment Interchange
- XML Base
- Associating Stylesheets with XML

Altres Working Groups:

- I. XML Encryption WG que ha generat els documents XML Encryption Requirements, XML Encryption Syntax and Processing , Decryption Transform for XML Signature, Additional XML Security URIs (Informational), application/xenc+xml Media Type Registration .
- II. XML Key Management Working Group (XKMS - XML Key Management Specification) que ha publicat els documents XML Key ManagementRequirements , i XML Key Management Specification .
- III. XML Signature WG treballa en la creació d'una sintaxi per representar la signatura dels recursos (URIs). Ha publicat els documents Signature Syntax and Processing , Canonical XML, Exclusive Canonical XML, XPath Filter, Additional XML Security URIs, XML Signature Requirements

Ver W3C World Wide Web Consortium. Leading the Web to Its Full Potential... http://www.w3.org/
--

Els documents del W3C tenen una capçalera a on s'indica l'estat del document. Els documents són revisats per membres del W3C i aprovats pel Director com Recomanació del W3C. Són documents estables que poden ser utilitzats com material de referència normativa en altres documents. La voluntat del W3C és promoure la difusió general.

1.1.3. Les tecnologies.

Les tecnologies d'Internet són els protocols, les aplicacions per editar, navegar i localitzar informació. I els llenguatges en que estan escrites , en particular Java.

Els protocols que es fan anar són:

DNS. És el servei de noms de domini. Una base de dades distribuïda i descentralitzada.

Tots els noms de la xarxa estan organitzats d'una manera jeràrquica, formant un sistema de dominis.

Ver RFC 1034 i RFC 1035

FTP: servei de transmissió de fitxers. (any 1985)

Ver RFC 959

SMTP i POP3: serveis de correu electrònic. Amb adreces amb el format [usuari@domini](#)

Ver RFC 822

NNTP: servei de notícies.

Ver RFC 977

HTTP : servei de worl wide web (WWW). (1993)

La edició de documents no presenta cap dificultat. Qualsevol editor pot servir.

Els navegadors són aplicacions que permeten visualitzar un document textual. Incorporen elements molt elementals de navegació (botons back/forward) i bookmarks per generar directoris de recursos.

Els buscadors són eines afegides a Internet per controlar grans volums de dades. Són bases de dades de recursos dissenyades de forma més o menys lògica. Yahoo! Està organitzat com un Directori de recursos i retorna consultes classificades segons una jerarquia, per àrees i subàrees temàtiques, mentre que altres es limiten a "llegir" documents, baixant a través dels hiperenllaços. Les dades que recullen solen ésser les metadades, o les paraules més freqüents del text (encara que no sempre les paraules més freqüents són les de major significat). Es necessari actualitzar aquestes bases de dades constantment o la variabilitat de la xarxa les faria obsoletes en molt poc temps.

Els mètodes de programació que es fan anar són Java, JavaScript i DHTML.

Java és el que millor adaptació té a les necessitats de la xarxa. Hi ha moltes aplicacions de Java a problemes concrets com ara:

- JSP Java Server Pages
- JDBC – Java DataBase Connectivity

També es fan anar Perl i C per escriure aplicacions CGI.

1.2. La localització dels recursos de la Web. La sintaxis de les URI.

La Identificació Uniforme de Recursos és una manera fàcil i extensible de identificar un recurs.

Es una cadena compacta de caràcters per identificar un recurs físic o abstracte.

Una URI pot ser una URL (localitzador) o un nom (URN).

Sintaxi per URI:

Parts d'un URI:

```
<first>/<second>;<third>?<fourth>  
<scheme>://<authority><path>?<query>
```

URL per Internet:

```
<userinfo>@<host>:<port>  
URI-reference = [ absoluteURI | relativeURI ] [ "#" fragment ]
```

Els caràcters URI (uric) poden ser reservats, no reservats o codis d'escapament.

```
uric = reserved / unreserved / escaped
```

Els caràcters no reservats són lletres, dígit i alguns símbols:

```
unreserved = ALPHA / DIGIT / mark  
mark = "-" | "_" | "." | "!" | "~" | "*" | "" | "(" | ")"
```

Caràcters reservats:

```
 ";" | "/" | "?" | ":" | "@" | "&" | "=" | "+" | "$" | "," | "#" | "[" | "]"
```

Els codis d'escapament:

```
escaped = "%" HEXDIG HEXDIG
```

Exemple:

"%20"
és el codi d'escapament del octet en binari "00100000" (ABNF: %x20), que correspon amb el caràcter espai (SP).

La forma general d'un URI es pot expressar així:

Esquema: identificador

Aquesta es la forma corresponent a un URI absolut. La paraula que hi ha abans del separador “:” és l’esquema, que determina la sintaxi de l’identificador. Aquesta sintaxi també pot permetre l’especificació d’URI relatius, en què s’omet l’esquema i el separador “:”

La definició d’URI es va publicar al document RFC 2396 l’any 1998.(obsolet)

Ver URI RFC 1630, RFC 2396, T. Berners-Lee : Uniform Resource Identifier (URI): Generic Syntax draft-fielding-uri-rfc2396bis-0x http://gbiv.com/protocols/uri/rev-2002/rfc2396bis.html URN RFC 2141
--

1.3. Motivació. Justificació de la necessitat de la semàntica.

La web semàntica és una àrea amb futur que proposa noves tècniques per a la representació del coneixement que facilitin la localització, compartició e integració de recursos. Aquestes tècniques es basen a la introducció de coneixement semàntic explícit que descrigui i estructurari la informació i els serveis disponibles, de forma que es pugui processar automàticament per un programa.

El XML està acceptat com l’estàndard per al intercanvi de dades en la xarxa. XML permet als autors crear les seves pròpies marques amb contingut semàntic. XML accepta qualsevol nova etiqueta que calgui definir. I els autors poden agafar qualsevol paraula i donar-li un significat qualsevol. (Es a dir, que la principal virtut de XML és també el seu punt feble).

Per exemple, un autor pot crear l’etiqueta <Títol> que sembla tenir algun significat. Però, per a una aplicació informàtica, no té més significat que l’etiqueta de HTML <H1>. Es a dir, l’ordinador no té coneixement de conceptes ni de relacions de conceptes que els humans tenim com a bàsics. Es pot fer creure a l’ordinador que Títol vol dir Autor, o qualsevol altra cosa.

Malauradament, XML tampoc permet parlar de màquines intel·ligents, però pot oferir molta ajuda per a predir quina informació hi ha entre les etiquetes, encara que aquestes etiquetes li semblin sense sentit.

Ver http://www.semanticweb.org/knowmarkup.html
--

Es pot esperar que amb un esforç relatiu es pugui transformar la xarxa de hiperenllaços de dades a relacions d'informacions.

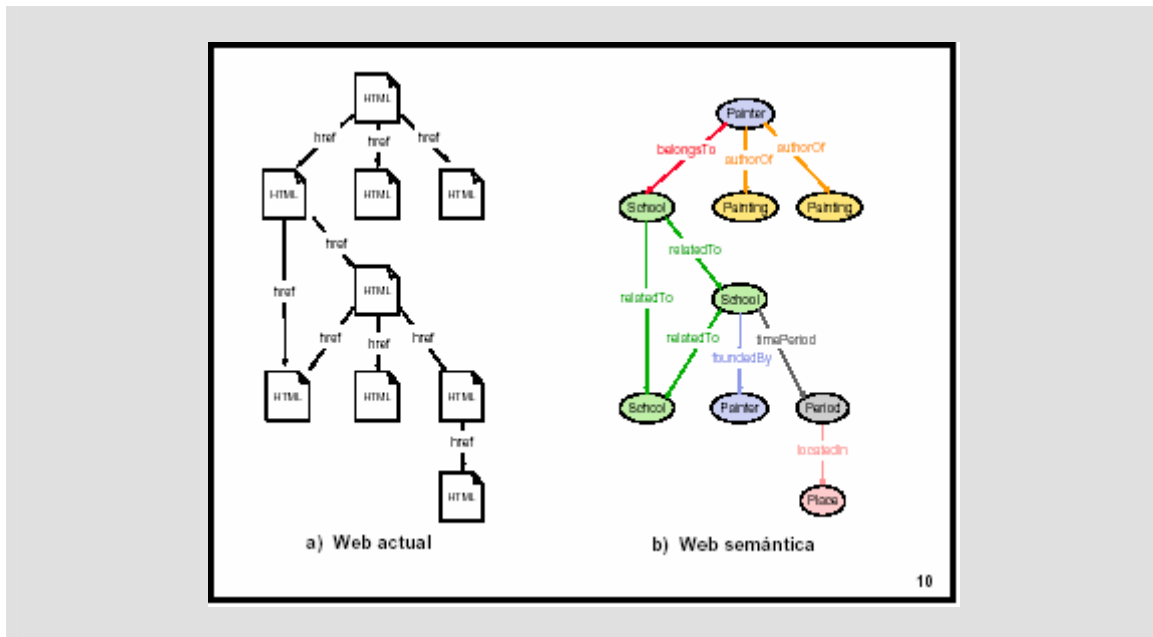


Figura extreta de:
Francisco Saiz , Pablo Castells : La web semántica. Tecnologías y aplicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Curso 2003 – 2004

RDF és un model que només utilitza elements que la màquina entén (les URI), però depèn dels recursos creats amb altres sistemes que han de ser estables i accessibles.
RDF pot reflectir de forma definida allò que hi ha i es veu. Permet gestionar recursos aliens i generar classificacions a partir de recursos sense etiquetes de contingut.

1.4. El model XML/RDF 1.4.1. Justificació

XML/RDF és un sistema de descripció de documents que ens permet fer registre de documents, gestió i avaluació de recursos.
Per tal de registrar-los, cal que els documents existeixen i siguin visibles. La gestió consisteix bàsicament en la recuperació de documents i generació d'informes, i la avaluació de recursos en catalogació, validació etc., des d'un punt de vista actual (anàlisi sincrònica) o amb visió històrica (anàlisi diacrònica).

1.4.2. Els recursos

Importància objectiva dels Documents: pel seu contingut. Integritat. Autenticitat.

Importància subjectiva : la que el propietari vulgui donar-los.

Exemples de recursos:

UOC. Universitat Oberta de Catalunya <http://www.uoc.edu/web/cat/index.html>
David Domingo: "...periodisme digital..." <http://www.uoc.edu/dt/20230/index.html>
Projecte Internet Catalunya - Empresa <http://www.uoc.edu/in3/pic/cat/pic2.html>
PIC-E : Informe Cap 4 http://www.uoc.edu/in3/pic/cat/pdf/PIC_empresa_0_5.pdf
Rosa Borge:"...administració electrònica" <http://www.uoc.edu/dt/20358/index.html>

Comentaris: UOC. Universitat Oberta de Catalunya és una plana amb un recull de recursos d'actualitat. Hi ha substitució de recursos molt sovint.(Recurs 1)

L'article de David Domingo es referent al periodisme digital. Les planes dels diaris són una mostra de recursos amb una vida molt curta. (En canvi tenen una llarga vida com documents d'hemeroteca). (Recurs 2)

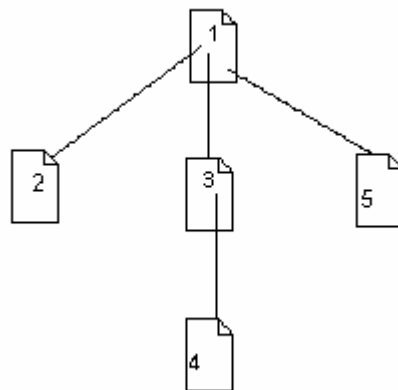
Projecte Internet Catalunya – Empresa té un resum del contingut i enllaços a documents pdf. (Els buscadors poden tenir problemes per trobar documents pdf). (Recurs 3)

El capítol 4 de l'informe del Projecte Internet Catalunya – Empresa és un document pdf. (Recurs 4)

L'article de Rosa Borge i Agustí Cerrillo ens permet parlar de dades amb valor, es a dir, documents. Cal preservar la integritat i l'autenticitat. (Recurs 5).

Cal remarcar que aquestos documents que trobem en un mateix espai, no tenen cap relació semàntica entre ells.

Estructura física (hiperenllaços) de l'exemple anterior:



1.4.3. La descripció de recursos i el registre.

Un registre de recursos es pot fer amb referències no declaratives (directes) amb HTML o XML, i amb referències declaratives (indirectes) amb RDF.

Les referències directes poden contenir les METADADES del full html, un RESUM del contingut i els HIPERENLLAÇOS.

Les referències indirectes només fan us d'hiperenllaços. Per tal d'utilitzar referències indirectes és necessari l'existència dels recursos (HTML, XML, pdf, etc).

La representació d'un recurs en HTML es la següent:

Llenguatge humà: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"> <HTML> <BODY> Aprenent a utilitzar les eines del periodisme digital. David Domingo: </BODY> </HTML>	Llenguatge màquina: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"> <HTML> <BODY> - Seqüència ASCII - </BODY> </HTML>
---	---

Comentari: La màquina no té cap informació referent al contingut de l'hiperenllaç. En conclusió: **HTML no es pot aprofitar per fer registre de recursos.**

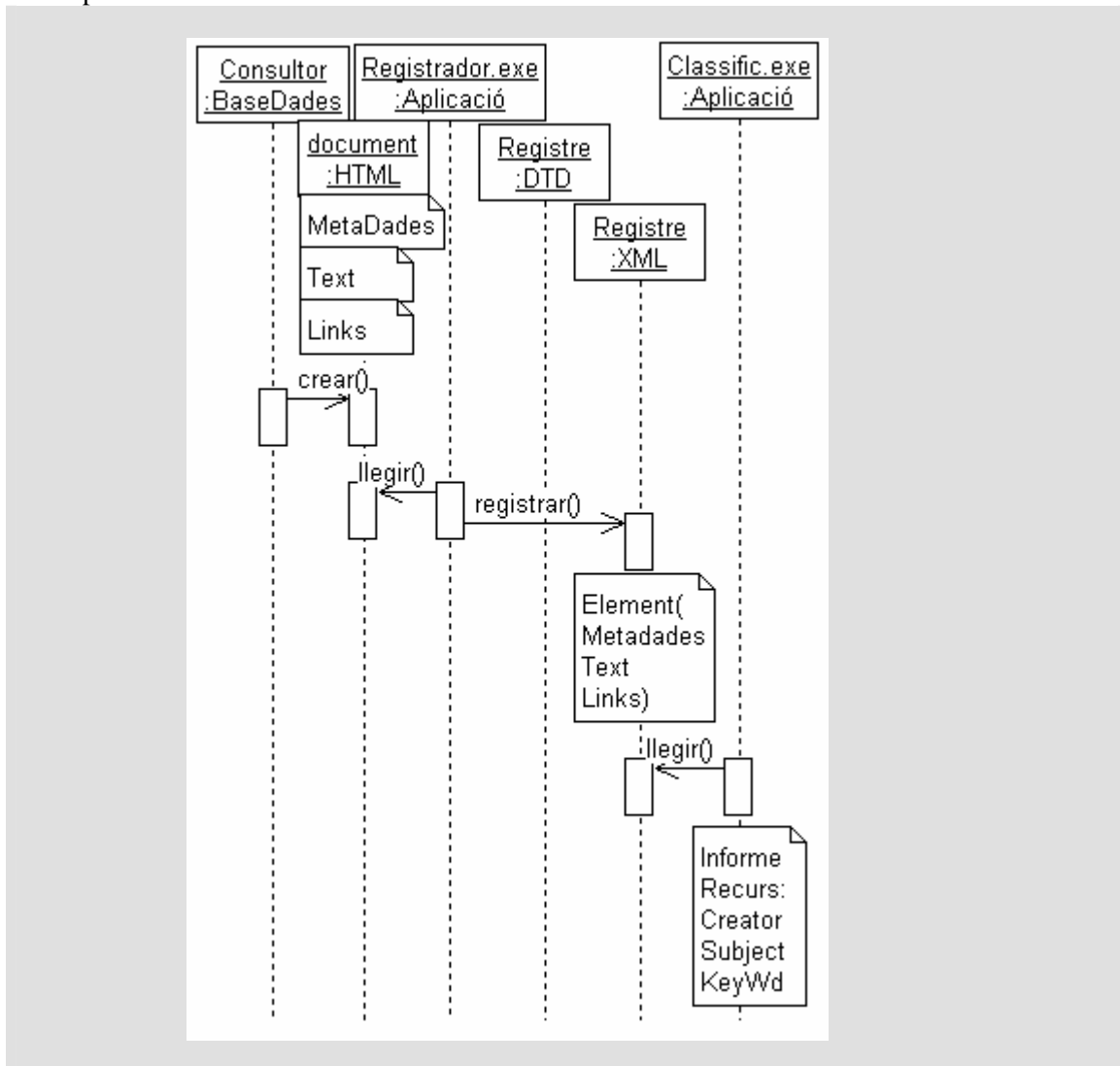
Exemple de registre de recursos fet amb XML:

Llenguatge humà: <?XML ENCODING="UTF-8" ?> <!DOCTYPE nom_DTD SYSTEM "Registre.dtd"> <Registre> <Element> <Creator>David Domingo</Creator><Subject>Eines del periodisme digital</Subject> </Element> </Registre>	Llenguatge màquina: <?XML ENCODING="UTF-8" ?> <!DOCTYPE nom_DTD SYSTEM "Registre.dtd"> <Registre><Element> <Creator>Seqüència 1</Creator><Subject>Seqüència 2</Subject> </Element> </Registre>
--	--

Comentari: XML afegeix significat al llenguatge màquina. Ens parla d'un creador del document i d'un tema més o menys concret.

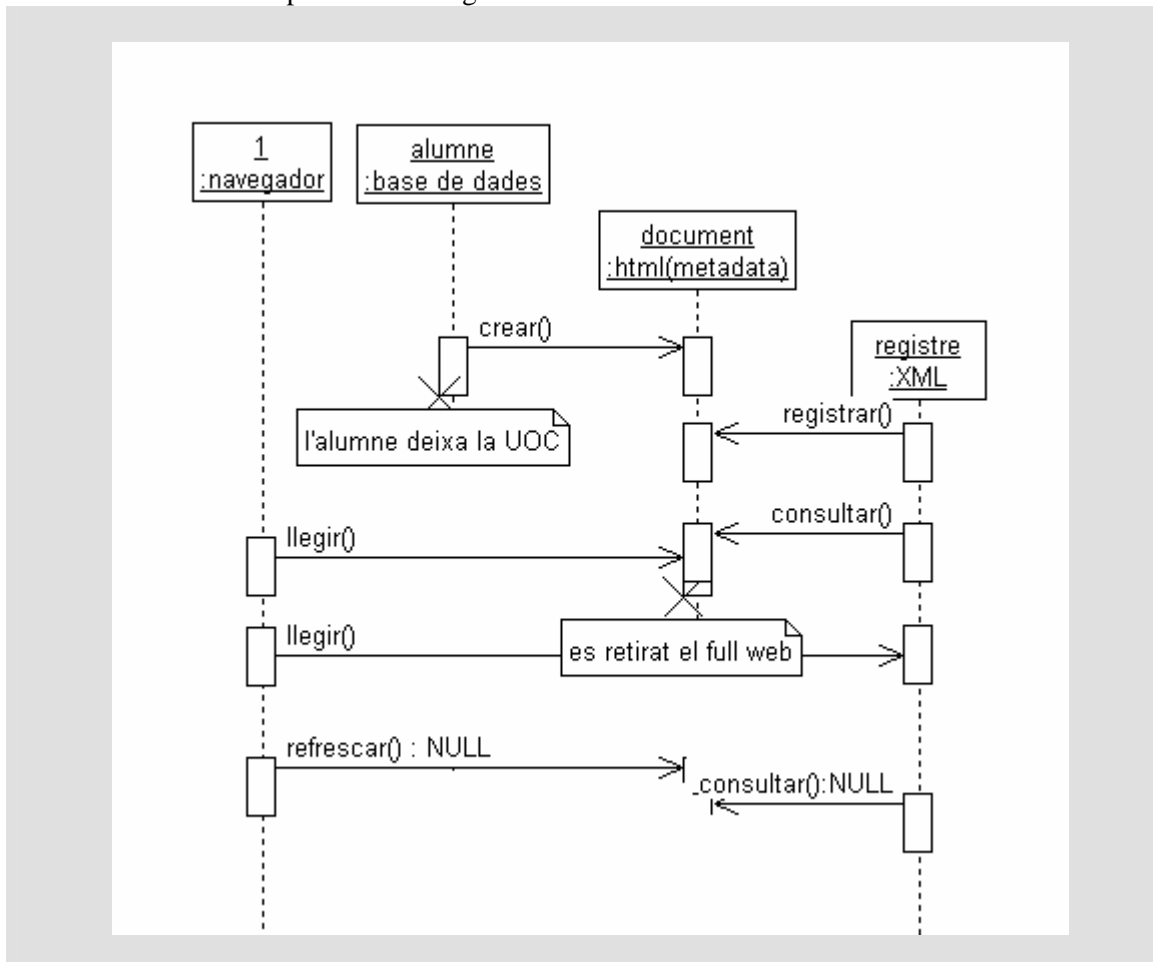
Ver DTD més endavant.

Exemple d'anàlisi sincrònica:



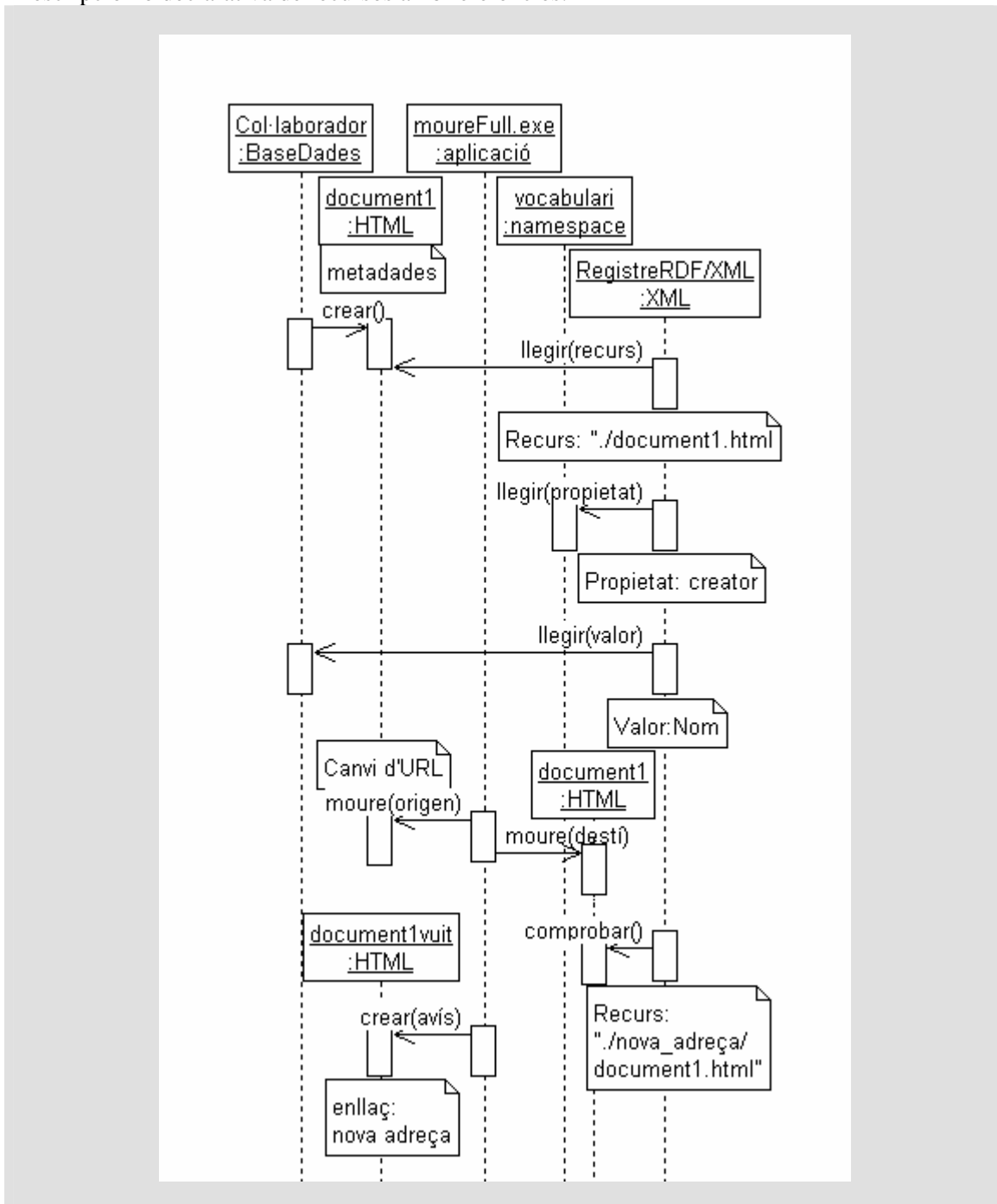
Comentari: Registrador.exe, RegistreXML i Classificador.exe depenen de la existència i integritat de RegistreDTD. (Registrador.exe, i Classificador.exe són aplicacions fictícies per representar la capacitat de mecanització del model).

Anàlisi diacrònic d'un problema de registre:



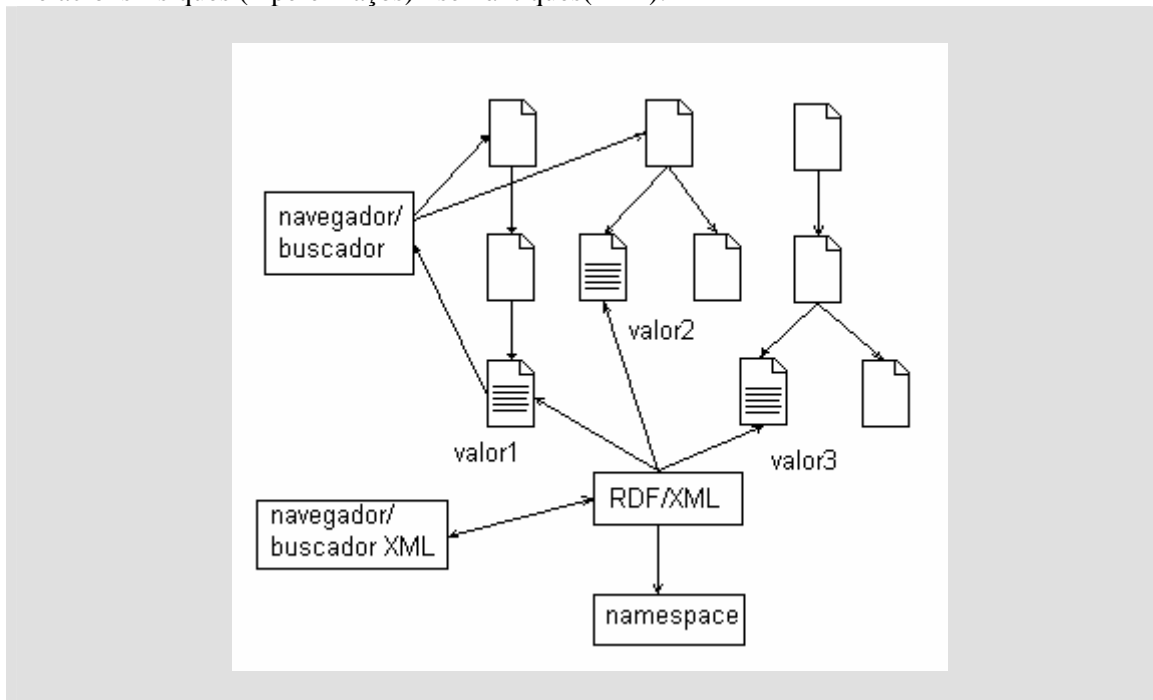
Comentari: Quan un full es retirat del servidor, els navegadors eliminen el recurs de les seves bases de dades, en canvi, en un registre (des de dins del domini) ens pot quedar constància que el recurs va existir.

Descripció no declarativa de recursos amb referències:



Comentari: El fet d'assignar una nova adreça al recurs ens pot fer creure que tenim dos recursos. En realitat només hi ha un recurs. Aquest fet es pot posar de relleu si afegim un identificador numèric o textual al recurs, que no canviaria quan modifiquem l'adreça.

Relacions físiques (hiperenllaços) i semàntiques(RDF):



Comentari: el gràfic vol comparar les relacions físiques entre fulls HTML, i les relacions semàntiques que genera un registre RDF/XML.

RDF permet l'accés directe a recursos molt concrets i sense relació física. Evita "soroll", es a dir, genera cerques d'informació molt més precises.

Capítol 2. XML - eXtensible Markup Language

XML permet la existència de documents amb qualsevol tipus d'informació. Pot existir un llenguatge escrit en XML per qualsevol col·lectiu d'usuaris. XML té dos metallenguatges amb els que definir els llenguatges que ens permetran escriure documents XML: el DTD i el XML Schema.

Objectius del capítol.

El present capítol pretén fer una presentació de XML. És un resum de la sintaxi, que permeti fer-se una idea dels sistemes que XML utilitza per tal de definir i generar documents descriptius de recursos.

Primer es fa la presentació del sistema DTD de descripció de recursos concrets i després el sistema Schema per a situacions més generals.

Per entendre les capacitats que aporta XML és convenient tenir coneixement dels llenguatges de marca SGML i HTML.

2.1. La sintaxis XML i els documents XML.

La sintaxis de XML és molt semblant a la de SGML i de HTML, però és molt estricta.

Qualsevol error ortogràfic invalidarà el nostre document. Un document ben format respecta les normes bàsiques de XML per al disseny de documents, i un document vàlid respecta les normes dictades per la seva DTD.

2.1.1. La sintaxis bàsica

Els documents XML només contenen text escrit amb caràcters ASCII.

La marca es crea amb els caràcters reservats: <, >, &, “, ‘, (ver entitats)

El text inclòs dintre dels caràcters menor que “<” i major que “>” o entre els signes “&” i “;” és el marcat. Són les parts del document que entenen els processadors de XML

El text marcat amb “<...>” es coneix com “Etiqueta”. La etiqueta està formada per “<”, un nom_element_identificador i “>”. Hi ha la etiqueta inicial i la final. Totes dues porten el mateix nom_element. I en mig de les dues etiquetes hi el contingut que pot ser text ASCII o altres etiquetes. A vegades no hi ha contingut (són les etiquetes buides).

Les etiquetes inicials comencen en un signe “<” i acaben en “>” mentre que les etiquetes finals comencen per “</” i acaben en “>”.

Els identificadors es poden concretar amb atributs. Són un par atribut=”valor”. Els atributs no es poden repetir dintre d’una etiqueta.

```
<element_de_marca atribut=”valor”>text marcat </element_de_marca>
```

Les etiquetes tenen una jerarquia en forma d'arbre que cal definir amb molta cura. Es concreta amb els signes : +, ?, *, (...)*; i paraules : ANY,

Les Entitats són abreviatures que el processador presenta amb el text íntegre.
Hi ha quatre tipus: interna, de text externa, binària externa i entitat de caràcter o numèrica.
XML especifica cinc entitats predefinides:

- & per a &
- < per a <
- > per a >
- ' per a '
- " per a "

Les diferències amb SGML i HTML són que en XML cal crear les nostres pròpies etiquetes , en XML la sintaxi és estricta, no es poden oblidar les cometes i cal distingir entre minúscules i majúscules.

En XML és obligatori tancar tots els elements.

Definició de models de contingut per a elements.

La capacitat de anidats elements és clau per al èxit d'un llenguatge de marca. Un element aïllat pot proporcionar només una part de la descripció d'un bloc de continguts mentre que els elements anidats afegeixen moltes possibilitats, Però les relacions entre elements no només porten avantatges, sinó també molts problemes.

El disseny curós dels models de contingut pot garantir que els creadors de contingut afegiran els elements necessaris, i s'evitarà la creació de combinacions problemàtiques.

Cal identificar les combinacions clau d'elements i els elements incompatibles.

Els DTD. – Declaració Tipus Document.

DTD Externes.

Totes les estructures es dissenyen per tal que descriguin minuciosament la marca que es podrà usar als documents. Tots els aspectes de la marca s'han de especificar a la DTD: els elements, els atributs que poden usar els elements, els tipus de valors, els valors predeterminats, etc.

Declaració de elements:

```
<! ELEMENT nom_element (llista d'elements que cal contenir en ordre fix , amb les opcions, separats per | ) >
```

(la llista d'elements ha quedat definida quan s'ha fet el model de contingut per als elements.)

Un element pot contenir altres elements, text ASCII o trobar-se buit.

La llista d'elements que un element pot contenir es una seqüència en l'ordre especificat. El nombre de vegades que pot aparèixer un element s'indica amb els signes següents:

“+” si cal que l’element aparegui una vegada o més.
“?” si l’element pot aparèixer una o zero vegades.
“*” si cal que l’element aparegui zero o més vegades.
“|” si cal triar un element d’una llista.
Es poden combinar de formes complexes afegint parèntesis.

Exemples

```
<!ELEMENT empleat (nom+, cognom, fill*)>  
<!ELEMENT animal (mamifer|ovipar)>  
<!ELEMENT div1 (head, (p | list | note) *, div2)>
```

El contingut de text ASCII s’indica amb #PCDATA, i els elements que no contenen res porten la paraula EMPTY.

```
<!ELEMENT comentari (#PCDATA)>  
<!ELEMENT nota EMPTY>
```

Exemples de DTD de registre de recursos : Registre.dtd

```
<!ELEMENT Article (Title*, Creator*, Subject*, Date*, Identifier+ )>  
<!ELEMENT Title(#PCDATA)>  
<!ELEMENT Creator(#PCDATA)>  
<!ELEMENT Subject(#PCDATA)>  
<!ELEMENT Date(#PCDATA)>  
<!ELEMENT Identifier(#PCDATA)>
```

Quan un element té un contingut mixt, amb text i elements, llavors es pot restringir el tipus d’elements però no l’ordre en que apareixen.

Exemple

```
<!ELEMENT p (#PCDATA|a|ul|b|i|em)*>
```

Els element poden tenir definits una sèrie d’**atributs** per tal de concretar el significat. A vegades farem us dels atributs per reduir la llista dels elements.

Declaració de llistes d’atributs:

```
<!ATTLIST nom_element nom_atribut (valors possibles separats per |) "valor per defecte"  
#required | fixed | implied >
```

Cal afegir una línia per cada nou atribut. (També és poden definir més d’un atribut per línia).

El valor dels atributs pot ser text (CDATA), pot ser una identificació, o pot prendre valors predefinitos.

Les **entitats** ens permetran escriure els signes reservats, crear abreviatures, o fer associacions amb objectes de fora del document.

Exemples

```
<!ENTITY XML "Extensible Markup Language">  
<!ENTITY imagen1 "/graphics/imagen1.gif" NDATA GIF SYSTEM >  
<!ENTITY tiff SYSTEM "path/photoshop.exe">
```

Un valor especial dels atributs és el nom d'un fitxer gràfic que ha estat definit com entitat.

Quan volen definir una entitat per ser utilitzada dintre del DTD emprarem la següent notació per definir-la:

```
<!ENTITY % nom_entitat_DTD "valor que representa">
```

I la cridarem escrivint : %nom_entitat_DTD; (entitat paramétrica).

XML aprofita les entitats numèriques o de caràcter dels esquemes de codificació: UTF-8, UTF-16, ISO-10646-UCS-2 ... (es pot aconseguir informació de les llistes d'entitats a <http://www.unicode.org>). ISO-Latin-1 és l'esquema predeterminat de XML.

DTD Internes en documents XML

Els elements d'un document XML es poden definir dintre del mateix document amb la instrucció de declaració amb la DTD entre corchetes

```
<!DOCTYPE nom_DTD [ <!ELEMENT nou_element EMPTY> ] >.
```

Aquest sistema només es útil en documents aïllats, o per afegir petites modificacions a la DTD general.

Les DTD internes també s'aprofiten per declarar les entitats dels esquemes de codificació amb una sola declaració situada al pròleg del document XML:

```
<!ENTITY %HTMLlat1 public "ISO 8879-1986//ENTITIES Added Latin 1//EN//XML">
```

Els documents XML.

Els documents XML comencen per un pròleg amb instruccions. El pròleg conté la declaració de XML amb la versió, i la declaració externa de la DTD i/o la declaració interna de la DTD. El pròleg també porta instruccions especials que cal executar per les aplicacions que les processen (instruccions de procés).

Declaració de procés de XML:

```
<?XML VERSION="1.0" RMD="ALL | INTERNAL | NONE" ?>
```

Declaració de codificació de caràcters:

```
<?XML ENCODING="UTF-8" ?>
```

```
( <?XML ENCODING="UTF-8 | UTF-6 | ISO-10646-UCS-2 | ISO-10646-UCS-4 | ISO-8859-1  
| ISO-2022-JP | Shift_JIS | EUC-JP " ?> )
```

Després cal afegir la instrucció de declaració de DTD:

```
<!DOCTYPE nom_DTD SYSTEM "nom_DTD.dtd">
```

i les modificacions:

```
<!DOCTYPE nom_DTD [...dtd interna ] >
```

Les instruccions de declaració es poden utilitzar per escriure comentaris invisibles iguals a HTML:

```
<!-- text del comentari -->
```

i si ens cal visualitzar el contingut de les etiquetes podem recórrer a :

```
<! [CDATA[ text del comentari de etiqueta]]>
```

Les etiquetes de XML funcionen igual que les de HTML, només que és obligatori que qualsevol etiqueta inicial tingui una etiqueta final. Les etiquetes buides es poden escriure amb una notació alternativa : <NomEtiqueta /> que porta el tancament incorporat.

Exemple de XML:

```
<?xml version="1.0"?>
<!DOCTYPE missatge [
  <!ELEMENT missatge (#PCDATA)>
]>
<!-- això és un comentari -->
<missatge>Hello, world!</missatge>
```

Exemple de Registre de Recursos amb XML:

```
<?XML VERSION="1.0" RMD="ALL" ENCODING="UTF-8"?>
<!DOCTYPE Registre SYSTEM "Registre.dtd">
<! - Exemple de creació de metadades amb XML -->
<Registre>
<Article>
<Title>UOC Universitat Oberta de Catalunya</Title>
<Creator>UOC</Creator>
<Subject></Subject>
<Date>17/12/2003</Date>
<Identifier>http://www.uoc.edu/web/cat/index.html</Identifier>
</Article>
<Article>
<Title>Aprentent a utilitzar les eines del periodisme digital.</Title>
<Creator>David Domingo</Creator>
<Subject>ensenyament virtual</Subject>
<Date>maig 2003</Date>
<Identifier>http://www.uoc.edu/dt/20230/index.html</Identifier>
</Article>
<Article>
<Title>Projecte Internet Catalunya - Empresa</Title>
<Creator>Projecte Internet Catalunya – Empresa
</Creator><Subject>TIC</Subject><Date>Decembre-2003</Date>
<Identifier>http://www.uoc.edu/in3/pic/cat/pic2.html</Identifier>
</Article>
<Article>
<Title>Les transformacions dels elements de
valor.</Title><Creator></Creator><Subject></Subject><Date></Date>
<Identifier>http://www.uoc.edu/in3/pic/cat/pdf/PIC\_empresa\_0\_5.pdf</Identifier>
</Article>
<Article>
<Title>L'administració electrònica: terra promesa o cursa d'obstacles sense
resultats?</Title><Creator>Rosa Borge</Creator><Subject></Subject><Date>novembre-
2003</Date><Identifier>http://www.uoc.edu/dt/20358/index.html</Identifier>
</Article>
</Registre>
```

Construcció de vincles amb XML.

Hi ha dos tipus bàsics de vincles a XML: els vincles simples i els EXTENDIDOS. Els vincles EXTENDIDOS poden ser vincles fora de línia o vincles multidireccionals.

Creació d'un vincle simple:

```
<element href="URL" xml:link="simple">.....</element>
```

Exemple:

```
<site href="http://www.htmlhelp.com" xml:link="simple"> htmlhelp.com </site>
```

La transformació d'un element en un hiperenllaç es fa amb els atributs següents:

- href = "URL"
- xml:link = "simple | document | extended | group | locator "
- role = "informació per a la màquina"
- title = "informació per a l'usuari"
- actuate = "auto | user"
- show = "embed | new | replace "

xml:link descriu el tipus de enllaç que estem creant. Role proporciona un mètode per passar informació a la aplicació (és opcional). Actuate ofereix la possibilitat d'especificar la manera d'activació de l'enllaç. El mode auto permet que els continguts del recurs s'insereixin automàticament al recurs local.

L'atribut SHOW = embed | new | replace ens permeten definir la presentació del recurs dintre de la finestra actual, una nova finestra o substituint la pàgina actual per una de nova.

Creació d'un vincle estès:

```
<element1 xml:link="extended">  
  <element2 xml:link="localitzador1" href="URL">...</element2>  
  <element2 xml:link="localitzador2" href="URL"> ...</element2>  
  <element2 xml:link="localitzador3" href="URL">... </element2>  
</element1>
```

els vincles estesos tenen els atributs ROLE, ACTUATE, SHOW , TITLE, INLINE=true / false CONTENT-ROLE, CONTENT-TITLE.

Ver XML namespaces. xml:link

Els XLINK és complementen amb els Xpointer.

Tot i que el navegador ens mostrarà el lloc assenyalat per el punter # , cal tenir clar que s'ha descarregat tota la pàgina a efectes de treball de aplicacions.

Xpointer ens permet crear vincles específics a elements identificats per un nom amb la barra vertical:

```
<link xml:link="simple" href="|abstract">
```

En aquest cas l'element té el nom "abstract".

Els fulls d'estil – XSL.

Els fulls d'estil de XML són el complement per transformar la informació d'un document XML en un full HTML.

```
<xsl:stylesheet><!-- Identifica la informació ->  
<xsl:template match = "body"> <! – identifica l'element sobre el que s'aplica la regla d'estil - >  
<fo:block font-size = "24 pt"> <! - fa un salt de línia i mida a 24->  
<process-children/><! - l'acció es transmet als fills ->  
</fo:block>  
</xsl:template>  
</xsl:stylesheet>
```

2.2. XML Namespace i XML Schema .

XML Schema és l'altre metallenguatge amb el que definir documents XML.

Té l'avantatge enfront DTD de poder treballar amb diferents definicions de document i de manera oberta.

Si DTD recordava molt l'estructura d'un fitxer, XML Schema ens fa pensar en l'estil de les Bases de Dades Relacionals.

Per exemple, HTML es definit en només un lloc [per un W3C WG] . Això pot ser definit amb DTD. El RDF, per altra banda, està específicament dissenyat per a ser un model per a diferents equips que fan servir un conjunt d'elements comuns.

XML Namespace

Un XML namespace es una col·lecció de noms, identificats per referències a URIs que són emprats als documents XML com elements i atributs.

Els namespaces permeten la reutilització de marques, en lloc de reinventar-les. A més, la referència a un URL produeix identificadors que són universalment únics.

Exemple 1:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  
<xs:element name="book">
```

La línia `<xs:element name="">` conté el URL <http://www.w3.org/2001/XMLSchema:element> name

Exemple 2:

```
<?xml version="1.0"?>  
<!-- both namespace prefixes are available throughout -->  
<bk:book xmlns:bk='urn:loc.gov:books'  
  xmlns:isbn='urn:ISBN:0-395-36341-6'>  
  <bk:title>Cheaper by the Dozen</bk:title>  
  <isbn:number>1568491379</isbn:number>  
</bk:book>
```

Es un document amb dos namespaces i amb elements d'ambdós namespaces:

- bk:title que equival a urn:loc.gov:books:title
- isbn:number que equival a urn:ISBN:0-395-36341-6:number

XML Schema

Qualsevol document XML es compon de elements i atributs. I la seva justificació és una DTD generada per algun grup concret. De la mateixa manera un XML-Schema és un document que descriu el format vàlid per a documents XML. Aquesta definició inclou quins elements son (i quals no) permesos a qualsevol punt, quins atributs cal afegir a cada element, el número d'aparicions de cada element, etc.

Però la identificació de l'origen del XML Schema és molt més flexible. Només cal dir la URL on és pot trobar, (o les URLs quan hi ha més d'un XML-Schema)

2.2.2.1. El document xsd

Els documents xsd (*schema*) comencen amb:

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

Definició d'un element que conté altres elements (sense repetir):

```
<xs:element name="nom_element_complexe" [minOccurs="1" maxOccurs="1"]>
  <xs:complexType>
    <xs:sequence>
      .../...
    </xs:sequence>
    .../...
  </xs:complexType>
</xs:element>
```

Definició d'un element que conté altres elements (amb repeticions a voluntat):

```
<xs:element name="nom_element_complexe" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      .../...
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Definició d'un element que no conté altres elements:

```
<xs:element name="nom_element_simple" type="xs:string" minOccurs="0"
maxOccurs="unbounded" />
```

Definició d'un element (complexType) amb atribut:

```
<xs:element name="nom_element_complexe">
  <xs:complexType>
    <xs:attribute name="nom_atribut" type="xs:string"/>
  </xs:complexType>
</xs:element>
```

Els elements que contenen subelements o porten atributs tenen tipus complex, mentre que els elements que contenen números, o cadenes o dates, però no contenen cap subelement són de tipus simple. Els atributs són sempre de tipus simple.

Etiqueta	Contingut	Tipus
Xs:schema	Xs:Element	Xmlns:xs , xmlns:xsd
Xs:element	Xs:ComplexType,	Name, type, minOccurs, maxOccurs, ref,
Xs:complexType	Xs:Sequence	Name,
Xs:sequence	Xs:Element	
Xs:attribute		Name, type, fixed, use
Xs:choice		
Xs:all		
Xsd:annotation	Xsd:documentation	
Xsd:documentation		Xml:lang="en"
Xsd:restriction		Base,
Xsd:maxExclusive		Value
Xsd:simpleType		Name
Xsd:pattern		Value

(Es pot definir xmlns:xs i xmlns:xsd)

Tipus	Tipus	Valors:
	Name	"nom_element" "nom_atribut"
	Type	"xs:string" "xs:date" "xsd:decimal" "xsd:NMTOKEN" "Items"
	MinOccurs	"0" "1" per defecte
	MaxOccurs	"unbounded" "1" per defecte
	Ref	"{Element_existent}"
	Fixed	"US"
	Base	"xsd:positiveInteger" "xsd:string"
	Value	"100" "\d{3}-[A-Z]{2}"
	Use	"required" "optional" "prohibited"

Type pot tenir valor definits en el mateix document xsd.

Ver Table 2. Simple Types Built In to XML Schema
 David C. Fallside : Schema Part 0: Primer W3C Recommendation, 2 May 2001
<http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/primer.html>

Exemple de *xml-schema*:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="article">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="title" type="xsd:string"/>
        <xsd:element name="creator" type="xsd:string"/>
        <xsd:element name="subject" type="xsd:string"/>
        <xsd:element name="date" type="xsd:string"/>
        <xsd:element name="identifier" minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xs:schema>
```

2.2.2.2. El document xml

```
<?xml version="1.0" encoding="UTF-8"?>
<nom_element_complexe atribut="id">
  <nom_element_simple1>cadena_ASCII</nom_element_simple1>
  <nom_element_simple2>cadena_ASCII</nom_element_simple2>
  <nom_element_complexe2>
    <nom_element_simple1>cadena_ASCII</nom_element_simple1>
    <nom_element_simple3>cadena_ASCII</nom_element_simple3>
    <nom_element_data>Data</nom_element_data>
    <nom_element_simple4>cadena_ASCII</nom_element_simple4>
  </nom_element_complexe2>
</nom_element_complexe>
```

Els elements que podem trobar en un document xml han estat definits en el document xsd corresponent. És obligatori conservar el nom, els atributs definits i l'ordre d'aparició en que han estat declarats.

2.3 Avantatges i limitacions de XML.

L'aprofitament de XML per a descripció de recursos.

XML permet fer registre de documents de manera molt semblant als procediments administratius que es fan servir a qualsevol empresa.

Només cal definir els termes que farem servir, o millor, buscar un vocabulari que ja estigui definit, i emprar-lo.

XML té, per una part, els elements de registre definits, (que són en bona part subjectius) i les cadenes de text en format ASCII (que són del tot subjectius a la persona que fa el registre), i per altra part els hiperenllaços (que són objectius perquè només depenen de l'existència del recurs que apunten).

Un document de registre amb XML que no tingués text lliure seria una aproximació a RDF, però caldria afegir el namespace de les etiquetes i que la informació surtis dels hiperenllaços.

La metainformació en XML, pot aparèixer dintre del mateix recurs o en un document XML independent, en el primer cas només ens interessa fer ressaltar la informació de les metadades del document. Quan generem un document nou amb les metadades d'altres recursos es pot guardar la informació encara que el document original desaparegui.

XML aporta la seva adaptació total a la xarxa, però no es cap garantia contra els mals usos de registre.

Exemple de descripció de recursos amb XML

```
<?xml version="1.0" encoding="UTF-8"?>
<article>
  <title>cadena_ASCII</title>
  <creator>cadena_ASCII</creator>
  <subject>cadena_ASCII</subject>
  <date>data</date>
  <identifier> URI</identifier>
</article>
```


Capítol 3 RDF

Objectius.

RDF/XML és la representació escrita de les idees desenvolupades amb RDF. És del tot necessari conèixer els fonaments de RDF.

El present capítol pretén donar una idea bàsica de la descripció de recursos en general, i dels documents web de la xarxa Internet en particular, des del punt de vista de RDF. (Model semàntic de RDF).

Cal tenir coneixements d'URI

T. Berners Lee : Uniform Resource Identifier
(URI) : Generic Syntax
<http://gbiv.com/protocols/uri/rev-2002/rfc2396bis.html>
RFC2396.txt

Cal tenir coneixements d'XML

Es pretén descriure la utilització que RDF fa de les URIs
Com RDF afronta la descripció de les informacions que genera (els grafos i les tuples) (La representació de les metadades).
Com RDF genera eines de treball (els vocabularis)
Com RDF genera un entorn de transmissió de dades. (RDF/XML)

RDF – *Resource Description Framework*

Origen

El RDF va sorgir l'agost de 1997 al Consorci Web W3C, amb el recolzament de protagonistes molt influents – Netscape, Microsoft – i motors de cerca. Es nodreix dels treballs de diversos col·lectius i per sobre de tot de la comunitat bibliotecària a l'entorn del Dublin Core (DC) que és un dels models de metainformació que primer a adoptat la sintaxi de RDF.

3.1. Model semàntic.

Resource Description Framework (RDF) i la Informació.

RDF és un llenguatge que identifica els Identificadors Uniformes de Recursos (*Uniform Resource Identifier*, URI) amb la informació.

RDF (INFORMACIO = URI)

La informació, des del punt de vista de RDF, és la associació de característiques definides dels recursos amb la seva ubicació. Interpreta les relacions binàries com parells ' propietat ' : ' valor ' i les expressa en forma de sentència.

RDF intenta trobar les característiques genèriques (propietats) que es poden reconèixer en molts individus i que són més o menys immutables.

En qualsevol cas, RDF considera la **ubicació** del **recurs** com la **propietat fonamental** i que l'identifica.

Un valor, en general, és una manifestació concreta de una propietat. Des del punt de vista de RDF els 'valors' també gaudeixen de la propietat fonamental de la ubicació. Es consideren objectes situats en un lloc definit.

En termes informàtics, 'propietats' i 'valors' tenen una URI assignada. (Amb major precisió direm que tenen una URIfref assignada).

Ver Capítol Inicial on s'explica que són les <i>Uniform Resource Identifiers</i> (URI)

RDF tracta la informació en forma de sentències formades per un subjecte, un predicat i un objecte. Pretenen descriure una informació molt concreta i indivisible (un àtom d'informació) que anomenem 'Metadata'.

Les metadates fan referència a un recurs concret, són creades per l'interès d'un agent generador (pot ser una persona qualificada o no, una organització, etc.), i són mecanitzables. Encara que poden ser escrites per una persona, l'ideal és que les generi un ordinador per tal de ser consumides per un altre ordinador.

Les sentències RDF , que descriuen metadates, estan formades, en general, per tres URIs, que fan funció de subjecte, de predicat i d'objecte.

3.1.1 Les adreces del Recursos.

En general, una URI és una representació simplificada d'un recurs. De fet la URI no és el recurs, però el representa.

Son cadenes de text amb un format definit:

```
Servei://nom_host.domini: port / path_directori / nom_document. Classe # nom_rekurs
```

El servei pot ser un protocol : http, ftp, etc

El nom del recurs es una cadena formada per caràcters acceptats. És opcional.
Les URI amb nom_recurs referenciat són les URIref.

Qualsevol element etiquetat , o tros de text pot tenir la seva URIref corresponent.

Es pot observar que dues URIs diferents poden emprar la mateixa paraula per identificar el seu recurs apuntat, i és molt probable que en cadascuna de les dues URI, aquesta paraula tingui significats diferents.

Ver els namespaces de XML en un paràgraf anterior.

Les URIref tenen una forma abreujada amb un nom qualificat (*QName*) i un nom local, i una forma desenvolupada amb un *namespace* i un nom local :

Forma abreujada: Prefix : nom_local
Forma desenvolupada: namespace / nom_local

Dues URIref , URIref1 i URIref2, són iguals sí i només sí les dues cadenes de caràcters són iguals, en totes les posicions de la cadena.
Per aquest motiu, no té importància que els noms locals estiguin repetits.

Tota la força de RDF és el tractament d'URIs. Però les URIs tenen una debilitat: estan exposades a modificació. Són molts els motius que poden portar a la modificació de la ubicació d'un recurs.

Aquest problema ha estat tractat de diferents maneres: hi ha autors que parlen de compromís de no modificar-les, altres parlen de URIs 'fortes', que poden suportar el pas del temps sense modificacions, i n'hi que parlen de URIs amb molt poc contingut semàntic , només l'any, el mes i el dia (Cool URIs)

Cool URIs don't change
<http://www.w3.org/Provider/Style/URI.html>
Introduction to PURLs
<http://purl.oclc.org/docs/inet96.html>

3.1.2 Generació de sentències RDF.

Ja hem vist que els elements mínims que utilitza RDF són URIs.

Per generar sentències RDF el primer que cal és seleccionar un recurs. Només fa falta conèixer la seva localització a Internet, es a dir, conèixer la seva URIref.

Un recurs és qualsevol objecte (un full HTML, un gràfic GIF, un full d'estil CSS, etc). Però no només els fitxers que formen les pàgines web són recursos. De fet, RDF considera un 'recurs' qualsevol cosa virtual o física que pugui tenir una referència.

Aquesta virtut de generalitzar el concepte de recurs pot tornar-se un problema en ocasions, quan hem de dir quina és la URIref que millor representa un objecte físic.

Una vegada relacionat un recurs amb una URIref, ens cal treballar amb les propietats. Necessitem trobar un 'vocabulari' RDF amb la col·lecció de propietats que utilitzarem per descriure el recurs seleccionat.

Es pot agafar Dublin Core com a model de vocabulari RDF.

Ver Dublin Core:
xmlns:dc
<http://purl.org/dc/elements/1.1/>

Altres direccions de Dublin Core

<http://es.dublincore.org/documents/2002/05/15/dcq-rdf-xml/index.shtml>
<http://www.bib.uc3m.es/~mendez/publicaciones/7jc99/rd f.htm>
<http://dublincore.org/documents/2002/07/31/dcmes-xml/dcmes-xml-dtd.dtd>

Si no trobem cap 'vocabulari RDF' aleshores l'haurem de crear nosaltres estudiant les propietats del recurs. Les propietats es poden classificar de moltes maneres: bàsiques i secundàries, concretes i abstractes, etc. Nosaltres podem estudiar totes aquestes propietats fins a crear un vocabulari RDF adient al recurs.

Un vocabulari RDF és un conjunt de parells <propietat definida – ubicació>.

D'aquesta manera, només cal recordar la ubicació (URI) de la propietat per poder entendre de què estem parlant exactament quan diem que un recurs té una determinada propietat amb un determinat valor.

Si mirem el Dublin Core trobarem 15 propietats definides al document
xmlns:dc=<http://purl.org/dc/elements/1.1/>

Una vegada definits el recurs i seleccionada una propietat definida en un vocabulari RDF ens cal seleccionar el valor associat que completa la sentència.

Hi dues maneres en que trobem els valors associats, la primera en forma de URIref, i la segona en forma de Literals.

Ver Literals en el capítol de XML

Un 'literal' és una seqüència de text, es a dir, fem una definició amb paraules del recurs. Quan definim el 'valor' associat amb un literal la significació queda limitada a allò que aporten les paraules emprades. (Cal tenir present que aquesta definició pot ser ambigua o no.) Un Literal pot anar acompanyat per un Tipus definit per tal de reduir l'ambigüitat.

El valor associat es pot presentar amb una URIref, (no donem un valor explícit sinó que diem on es pot trobar un valor adient). De fet, amb aquest sistema el valor es pot processar com un Recurs, i permet que es pugui afegir altra significació vinculada.

La selecció de Recurs i Valor ha de evitar que el propi Recurs sigui emprat com a valor, es a dir, cal evitar generar bucles en que un recurs pugui aparèixer al mateix temps com a subjecte i com objecte. Les aplicacions que processen metadades es trobarien amb un bucle incòmode de processar .

3.1.3 Els vocabularis RDF.

Els vocabularis RDF són aplicacions pràctiques del RDF a problemes més o menys concrets. Un vocabulari RDF es un conjunt de tuplas (propietat, localització, definició) que es poden utilitzar com a predicats en sentències RDF.

Es del tot imprescindible que les URIref que permeten la localització del vocabulari no siguin modificades mai.

El vocabulari es pot presentar de moltes maneres, en forma de document HTML amb *links* locals, o en forma de DTD de XML.

DTD : Ver capítol de XML.

Qualsevol pot crear un vocabulari, però el valor radica en la acceptació que tingui en el conjunt dels usuaris. Es per això que sembla millor buscar-los, que no escriure'ls. Quant més gran sigui l'ús d'un vocabulari, més gran és la probabilitat que sobrevisqui.

Es a dir que un document RDF a de tenir contingut semàntic, contingut sintàctic i informació de l'entitat generadora, mentre que el format és indiferent.

3.2 Les formes de representació de sentències RDF.

3.2.1 La representació gràfica.

Amb RDF cadascuna de les metadades es pot representar amb un grafo , amb una tupla, o amb un document amb text XML.

RDF aprofita el model matemàtic de **grafs dirigits** i jerarquitzats per generar representacions gràfiques de les sentències.

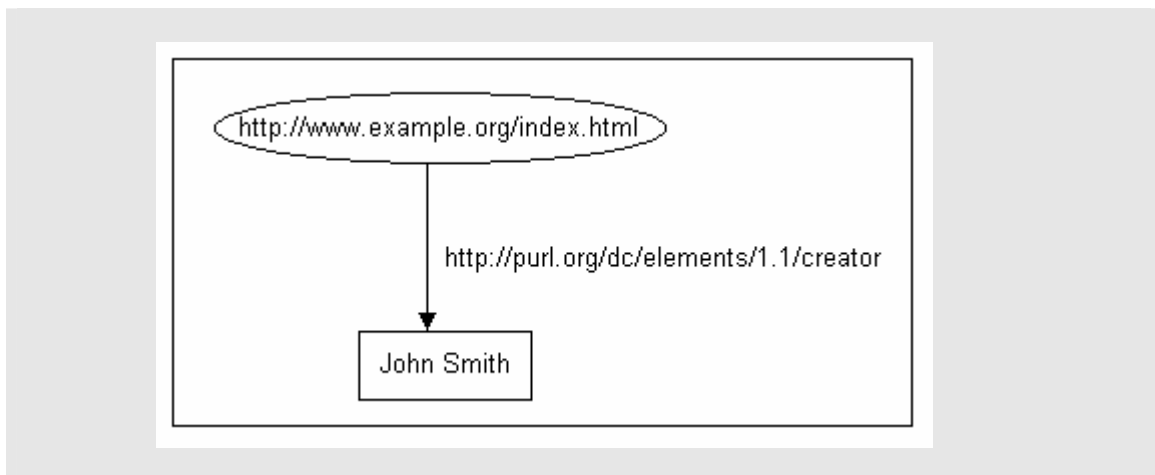
Els grafos estan compostos per nodes i arcs etiquetats.

Els nodes tenen forma d'el·lipse quan l'etiqueta és una URI i forma de rectangle quan és un Literal.

RDF fa correspondre un recurs a cadascun dels nodes. Podem trobar nodes amb contingut, i nodes en blanc. Un node amb contingut té una etiqueta amb una cadena de caràcters vàlids que representa un recurs definit amb URI. Un node en blanc representa un recurs del que no coneixem la URI, o no volem indicar-la (recurs indefinit).

Un arc etiquetat correspon a una propietat definida en un document RDF.

Exemple de graf RDF.:



Indicacions per crear grafos RDF.

- Totes les sentències RDF tenen un graf associat.
- Els grafos amb el mateix Subjecte es transformen en un sol graf amb l'etiqueta del Node subjecte igual al Recurs, i amb els corresponents arcs etiquetats i nodes objecte. (És una simplificació en que passem de tenir varis grafos a només un.)
- Si tenim dues sentències en que el node subjecte de la segona coincideix amb el node objecte de la primera, aleshores les dues sentències es poden transformar en un sol graf de tres nodes i dos arcs de la forma: node , arc, node_objecte = node_subjecte, arc, node. (És una simplificació en que passem de tenir dos grafos a només un.)

- Una variant de la simplificació anterior és la fusió de dos grafs en que el node_objecte=node_subjecte es un node en blanc. (És la expressió d'una referència indirecta a un recurs.)
- Transformació d'una relació n-aria en relacions binàries. Les estructures de dades se poden representar de dues formes:
 - Assignant un identificador a l'estructura
 - Utilitzant un node en blanc (que actua com a node subjecte del qual surten els arcs etiquetats cap als valors components de l'estructura.
- Es convenient substituir Literals per URIs (per tal de incrementar el contingut, si es pot evitar la complicació del model.)
- El model permet crear conjunts de recursos.
- El model permet crear seqüències de recursos.
- El model permet crear col·leccions de recursos.
- El model permet crear metadades amb referències.

3.2.2 Representació amb tuples.

La segona manera de representar una sentència RDF és amb una tupla.

Una tupla RDF és una seqüència subjecte, predicat, objecte presentada en aquest ordre.

```
<URI subjecte> <URI predicat> <URI objecte>
```

Les tuples les podem trobar en forma abreujades amb un QName.

La presentació de sentències en forma de tupla té alguna semblança amb les taules relacionals. Les taules relacionals es poden transformar en tuples si som conscients de que les tuples només poden ser relacions binàries (subjecte – objecte).

Les taules relacionals amb estructures de dades representen relacions n-aries. Cal transformar-les en taules binàries abans de fer la transformació final a tuples RDF.

El primer element de la tupla, el subjecte, sempre és una URI, que representa un recurs web o la situació de la representació textual de un recurs físic.

El segon element, el predicat, és una propietat definida en algun dels documents RDF de definició de propietats. Cal mencionar rdf, rdfs, dc, etc.

El tercer element, l'objecte, és el valor de la propietat associada al recurs, i pot ser una URIref o un Literal.

Els Literals RDF de tipus definits són els mateixos que els de XML.

3.2.3 La representació amb text XML (RDF/XML).

La tercera forma de representació de sentències RDF és amb codi XML.

El document RDF escrit en XML ha de conservar tot el significat de les metadades generades.

La representació RDF/XML és mecanitzable i transportable, i el codi es obert.

El text XML es pot trobar dintre del recurs, o en un document fora del recurs. Aquesta característica permet diverses relacions entre el document i la informació que volem guardar del recurs. Podem trobar-nos les següents situacions:

1. el recurs conté les metadades XML
2. les metadades són fora del recurs. Les metadades apunten al recurs.
3. les metadades són fora del recurs. El recurs enllaça les metadades externes.
4. les metadades són fora del recurs. No hi cap enllaç físic entre metadades i recurs.

A continuació es presenta l'aspecte de una sentència en totes les maneres mencionades.

Exemple de representacions de sentències RDF:

sentència

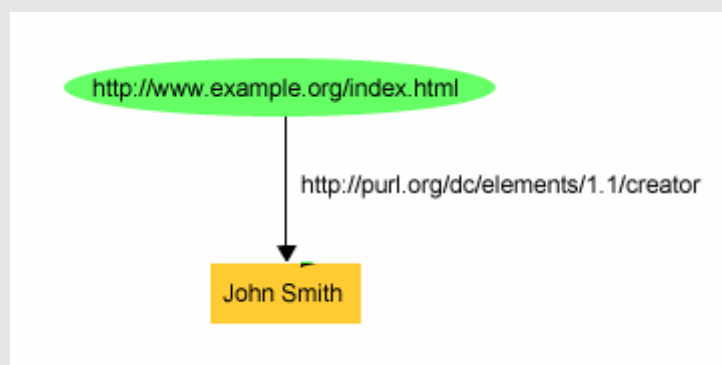
`http://www.example.org/index.html has a creator whose value is John Smith`

taula

Nom_Taula: `http://purl.org/dc/elements/1.1/creator`

http://www.example.org/index.html	John Smith
---	------------

Grafo



tupla

`<http://www.example.org/index.html> <http://purl.org/dc/elements/1.1/creator> "John Smith" .`

(versió abreujada)

`<http://www.example.org/index.html><dc:creator> "John Smith"`

Codi RDF/XML

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">

  <rdf:Description rdf:about="http://www.example.org/index.html">
    <dc:creator >John Smith</dc:creator>
  </rdf:Description>

</rdf:RDF>
```

Comentari:

quan intentem consultar la adreça <http://www.example.org/> ens apareix la següent informació:

You have reached this web page by typing "example.com", "example.net", or "example.org" into your web browser. These domain names are reserved for use in documentation and are not available for registration. See RFC 2606, Section 3.

Es a dir, el recurs presentat és del tot fictici.

3.3 Propietats fonamentals de RDF.

RDF és molt simple, el seu plantejament és la utilització de les URI i res més, però aquesta simplicitat fa que sigui utilitzable en qualsevol situació i per qualsevol persona. És modular, les tuplas accepten la propietat associativa, i els nodes permeten generar arbres que poden ampliar-se i afegir més informació. És extensible, perquè RDF pot aprofitar nous vocabularis. RDF no queda bloquejat en els límits d'una branca del coneixement. També es pot dir que no requereix gaire consens. Només cal triar el millor. És obert, el seu codi pot ser llegit per qualsevol aplicació perquè està escrit en XML. I per tant, també és transportable. Pot aprofitar la xarxa per transmetre's. Per últim, també es auto referenciable, es a dir, es poden generar metadades de metadades.

El fet que RDF es transmet en documents XML pot fer pensar que és el mateix, però hi ha una diferència bàsica. RDF té un objectiu molt clar i específic, està pensat per processar metadades, mentre que XML té un aprofitament general per tractar qualsevol tipus de dades.

La definició del llenguatge XML fa que hi hagi altres diferències. XML es va pensar inicialment com un sistema tancat, amb totes les seves possibilitats definides en una DTD i amb un ordre estricte dels elements dintre dels documents. Després s'ha afegit l'esquema amb *namespaces* que fa que funcionalment s'hi assembli a RDF.

RDF es pot utilitzar amb l'objectiu de crear un mapa lògic i comprensible. Utilitza elements de XML, que hagin estat inclosos dins un vocabulari RDF la qual cosa permet de utilitzar eines informàtiques en un entorn mecanitzat. D'aquesta manera RDF proporciona uns resultats acceptables en els seus resultats, amb menys "soroll" i sense fer us de mètodes de "força bruta".

Les aplicacions de RDF són:

- Catalogació de recursos.
- Emmagatzematge d'informació.
- Descripció de recursos.
- Generació de mapes lògics de llocs web.
- Recuperació de informació. Orientació i gestió d'informació.

El resultat de treballar amb RDF hauria de portar un coneixement de què hi ha a la xarxa, quin aspecte té i on s'hi troba, com evolucionen els recursos...

El fet de existir mapes lògics pot estalviar molts esforços de consulta de pàgines, perquè es tindrà un coneixement de què hi ha al final del camí abans d'anar-hi.

RDF pot resultar poc pràctic en algunes situacions. Per exemple en el control de pàgines amb URIs contextuais, que apunten cap a recursos que canvien tots el dies, o en casos de URIs que segons el *browser* poden apuntar a un recurs o un altre.

Capítol 4. RDF/XML

A partir dels coneixement del tractament de la informació que fa RDF i de la base sintàctica i gramatical de XML es fa una presentació de la generació de documents amb continguts de metadades.

RDF/XML és més complicat que XML. Requereix més esforç per aplicar-lo, però té moltes més possibilitats.

Es pretén fer una presentació dels vocabularis bàsics que formen RDF/XML, la forma d'utilització, i els documents on son definits.

Els documents RDF/XML.

Un document RDF/XML és una llista de descripcions. Cada descripció fa referència a només un recurs, i conté una llista de propietats. Els valors que prenen aquestes propietats poden ser URIs, literals o altres descripcions.

Això es tradueix en RDF/XML en seqüències d'elements `rdf:Description`. Aquests elements poden tenir l'atribut `rdf:about` o `rdf:ID` però no tots dos.

Els elements `rdf:Description` han de contenir un `rdf:Property`.

Els `rdf:Property` són recursos emprats com a Predicat. Cal remarcar que les propietats (*Property*) de RDF no tenen consideració d'atribut, sinó que són objectes independents, i que poden ser descrits com recursos.

Convencionalment es parla de elements nodes i de elements *Property*. A l'exemple `rdf:Description` es el element node (emprat tres vegades per als tres nodes) i `ex:editor` i `ex:homePage` són els dos elements *Property*

```
<rdf:Description>
  <ex:editor>
    <rdf:Description>
      <ex:homePage>
        <rdf:Description>
          </rdf:Description>
        </ex:homePage>
      </rdf:Description>
    </ex:editor>
  </rdf:Description>
```

Les estructures de dades han tingut una ampliació amb `rdfs`. Les Classes (`rdfs:Class`) són recursos definits que denoten un conjunt de recursos, i tenen la propietat `rdf:type`. (Fan funció de `rdf:Property`). Les Classes estan estructurades en una jerarquia denotada per `rdfs:subClassOf`. El element arrel és `rdf:Resource`.

Els `rdfs:Containers` son col·leccions de recursos. (En ocasions poden generar bucles que no és poden processar informàticament).

Relacions entre XML i RDF.

¿Com aprofita RDF el llenguatge XML?

RDF aprofita XML per codificar i transmetre metadata. La forma específica de XML per aquest fi es coneix com RDF/XML.

RDF/XML aprofita elements, atributs, entitats, tipus definits i namespaces de XML. Des de el punt de vista de RDF, un element és una etiqueta que conté una URI. El mateix passa amb els atributs.

Els documents RDF/XML comencen amb la etiqueta `<?xml version="1.0"?>` o `<?xml version="1.0" encoding="ISO-8859-1"?>`

També porten la capçalera de definició :

```
<!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
```

o també

```
<!DOCTYPE rdf:RDF [
```

```
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
```

```
<!ENTITY rdfs "http://www.w3.org/TR/2000/01/rdf-schema#">
```

```
<!ENTITY refOnto "http://www.fing.edu.uy/cursos/inco/grupos/csi/esp/Cursos/cursos_posg/WebBD2002/esquemaOntologia#">
```

```
]>
```

Els tipus definits que es poden utilitzar amb RDF/XML són els mateixos que els definits a XML. (xsd:date i xsd:integer). De totes maneres hi algunes excepcions.(ver <http://www.w3.org/TR/rdf-primer/#ref-rdf-semantic>s)

El sistema de namespaces de RDF/XML també és el mateix que a XML.

Elements de xml utilitzats a RDF/XML:

Atribut	Xml:lang	xml:lang = "en"
rdf:parseType="Literal"	Xml:size	Size="10"
rdf:parseType="Literal"	Xml:id	Id="23"
	Xml:base	

Els elements RDF/XML.

¿on es poden trobar?

Els elements RDF/XML estan definits en diversos documents. Els més importants són els que trobem a XML, i xsd, i el propis de RDF rdf, rdfs. També cal mencionar dublin core, dc.

Les adreces on estan definits són les següents:

- rdf = <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
- rdfs=<http://www.w3.org/TR/2000/01/rdf-schema#>
- dc = <http://purl.org/dc/elements/1.1/>
- owl = <http://www.w3.org/2002/07/owl>
- xsd = <http://www.w3.org/2001/XMLSchema>

I s'indiquen d'aquesta manera dintre del document :

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
```

Els elements de `xmlns:rdf=`<http://www.w3.org/1999/02/22-rdf-syntax-ns#>

CoreSyntax	Rdf:RDF	Identifica el codi RDF
Element	Rdf:Description	És l'element contenidor bàsic.
CoreSyntax	Rdf:about	Acompanya a rdf:Description
OldTerm	Rdf:aboutEach	
OldTerm	Rdf:aboutEachPrefix	
Element	Rdf:Alt	Contenidor de recursos que són alternativa un a l'altre.
Element	Rdf:Bag	Contenidor de recursos que formen un conjunt. Pot substituir a rdf:Description amb rdf:ID, (mai rdf:about). <pre><rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"> <rdf:Bag ID="mybag"> <rdf:li resource="http://rama.cpe.fr/index.html"/> <rdf:li resource="mailto:champin@cpe.fr"/> <rdf:li> literal element </rdf:li> </rdf:Bag></pre>
Atribut de xmlschema	Rdf:datatype	<pre><ex:size rdf:datatype="http://www.w3.org/2001/XMLSchema#int">123 </ex:size></pre>
Collection	Rdf:first	
CoreSyntax	Rdf:ID	Permet identificar els nodes en blanc. Permet identificar una metadada : <code>rdf:ID="tupla1"</code> (reification)
Element	Rdf:li	Es pot trobar dintre de rdf:Bag i representa un element del grup. En els grafs apareix com <code>rdf:_n</code> ,

Collection	Rdf:List	
Collection	Rdf:nil	
Atribut	Rdf:nodeID	
Reification	Rdf:object	
	Rdf:parseType	= "Collection"
Reification	Rdf:predicate	
Classe	Rdf:Property	Les Properties són recursos emprats com predicats
Atribut	Rdf:resource	
Collection	Rdf:rest	
	Rdf:Seq	És un contenidor de recursos ordenats.
Reification	Rdf:Statement	Permet generar metadades d'altres metadades.
Reification	Rdf:subject	
Property	Rdf:type	Informa sobre rdf:Bag, rdf:Alt i rdf:Seq
Utility property	Rdf:value	
Classe	Rdf:XMLLiteral	
	Rdf:_n	Contenidor amb rdf:alt

Els elements de `xmlns:rdfs=http://www.w3.org/TR/2000/01/rdf-schema#`

Classe	Rdfs:Class	<code><rdfs:Class rdf:ID="RECURSOS_TEORICOS"> <rdfs:comment> material teorico del curso, por ej: Clases dictadas </rdfs:comment> <rdfs:label> RECURSOS TEORICOS </rdfs:label> <rdfs:subClassOf rdf:resource="#MATERIALES"/> </rdfs:Class></code>
Property	Rdfs:comment	<code><rdfs:comment> Curso de Sistemas Operativos </rdfs:comment> Anotación para el autor o el lector.</code>
	Rdfs:Container	Són col·leccions de recursos.
Classe	Rdfs:Datatype	
Property	Rdfs:domain	<code><rdfs:domain rdf:resource="#CLASE"/></code>
Utility property	Rdfs:isDefinedBy	
Property	Rdfs:label	<code><rdfs:label> CURSO </rdfs:label> Es la interpretació humana del recurs.</code>
Classe	Rdfs:Literal	
	Rdfs:member	
Property	Rdfs:range	<code><rdfs:range rdf:resource="#GENERALIDADES"/></code>
Classe	Rdfs:Resource	Classe principal. Totes les coses que es descriuin amb expressions RDF s'anomenen recursos.
Utility property	Rdfs:seeAlso	<code><rdfs:seeAlso rdf:resource="http://www.w3.org/1999/.status/PR-rdf-</code>

		schema-19990303/status"/>
Property	Rdfs:subClassOf	<rdfs:subClassOf rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema#Resource"/> </rdfs:Class>
Property	Rdfs:subPropertyOf	
	Rdfs:vocabulary	

Els elements de `xmlns:xsd` = <http://www.w3.org/2001/XMLSchema#>

	Xsd:date	Tipus definit
	Xsd:integer	Tipus definit, "string"^^xsd:integer

Ver David C. Fallside : XML Schema Part 0:
 Primer <http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/>

Elements definitos en altres documents:

Elements definitos en Dublin Core:

Contingut	Dc:Title	Títol: el nom donat a un recurs (¿nom?)
Propietat intel.	Dc:Creator	L'entitat responsable de la elaboració del contingut. (Persona, organització, servei...)
Contingut	Dc:Subject	Matèria: el tema del contingut del recurs (tema: clau, codi)
Contingut	Dc:Description	Descripció del contingut.
Propietat intel.	Dc:Publisher	L'entitat responsable de fer accessible el recurs.
Propietat intel.	Dc:Contributor	Contribucions eventuais.
Versió/accés	Dc>Date	Una data associada amb un esdeveniment del recurs. (creació, publicació, modificació, ...)
Contingut	Dc:Type	Naturalesa o gènere del contingut. (categories, nivells, classificacions...)
Versió	Dc:Format	Manifestació física del recurs.
Versió	Dc:Identifier	Una referència unívoca i no ambigua del recurs en el seu context.(URI, URL, ISBN, ...)
Contingut	Dc:Source	Font. Referència a un recurs previ del qual deriva l'actual.
Versió	Dc:Language	Llengua del contingut del recurs.
Contingut	Dc:Relation	Referència a un recurs emparentat. (conforme a un sistema formal d'identificació.)

Contingut	Dc:Coverage	Espai, temps, jurisdicció, ...
Propietat intel.	Dc:Rights	Drets sobre el recurs.

Exemple d'utilització dels vocabularis rdf, rdfs i dc :

Ver.: Eva M^a Méndez Rodríguez
RDF: UN MODELO DE METADATOS
FLEXIBLE ...
<http://www.bib.uc3m.es/~mendez/publicaciones/7jc99/rdf.htm>

```
<rdf:RDF>
<rdf:Description about=""
  xmlns:ddc="http://purl.org/net/ddc#"
  dc>Title="Resource Description Framework (RDF) Schema Specification"
  dc:Description="The Resource Description Framework (RDF) is a foundation for
processing metadata; it provides interoperability between applications that exchange machine-
understandable information on the Web. RDF emphasizes facilities to enable automated
processing of Web resources."
  dc:Publisher="World Wide Web Consortium"
  dc>Date="1999-03-03"
  dc:Format="text/html"
  dc:Type="technical specification"
  dc:Language="en">
<dc:Subject resource="http://purl.org/net/ddc/025.30285"/>
<dc:Subject resource="http://purl.org/net/ddc/025.316"/>
<dc:Subject ddc:Class="025.302855741"
  ddc:Heading="Applications of computer file organization and access methods"/>
<dc:Creator>
  <rdf:Bag rdf:_1="Dan Brickley"
    rdf:_2="R.V. Guha" /></dc:Creator>
<rdfs:seeAlso rdf:resource="http://www.w3.org/1999/.status/PR-rdf-schema-
19990303/status"/>
</rdf:Description>
</rdf:RDF>
```

Comentaris: Apareix una referència a una ampliació del Dublin Core en forma de namespace :
xmlns:ddc=<http://purl.org/net/ddc#> amb l'element ddc:Heading
També es poden trobar més elements de Dublin Core al namespace
xmlns:dcterms=<http://purl.org/dc/terms/>
La utilització de 'resource' (en lloc de rdf:resource) vol dir que la capçalera inclou la definició
de rdf.

Procediment per crear documents RDF/XML.

Ver
<http://hipatia.uc3m.es/documents/2002/07/31/dcmes-xml/>

Declaració de XML.

```
<?xml version="1.0"?>
```

Referència de la DTD de XML corresponent.

```
<!DOCTYPE rdf:RDF PUBLIC "-//DUBLIN CORE//DCMES DTD 2002/07/31//EN"  
"http://dublincore.org/documents/2002/07/31/dcmes-xml/dcmes-xml-dtd.dtd">
```

```
<!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
```

Comentari: consultar DTD i ENTITY en XML.

Declaració d'ús de continguts RDF.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
xmlns:dc="http://purl.org/dc/elements/1.1/"  
xmlns:externs="http://www.example.org/terms/">  
  
</rdf:RDF>
```

(comentari: rdf:RDF permet a les aplicacions reconèixer text RDF/XML)
(comentari: ver namespaces de XML)
(comentari: els xmlns més freqüents són : rdf, rdfs, dc,)

Descripció dels recursos i la informació.

Exemple de recurs:

```
<rdf:Description rdf:about="http://example.org/">  
...  
</rdf:Description>
```

(comentari: rdf:Description és un contenidor de recurs)

(comentari: cada Recurs pot incorporar un , més d'un, o cap identificador del recurs)

Exemple de recurs amb informació en forma de URIref(node, arc, node)

```
<rdf:Description rdf:about="http://example.org/">  
  <http://purl.org/dc/elements/1.1/creator>  
    <http://www.example.org/staffid/85740>  
  </http://www.example.org/staffid/85740>  
  <http://purl.org/dc/elements/1.1/creator>  
</rdf:Description>
```

Exemple de recurs amb informació en forma de Literal (node, arc, node-literal)

```
<rdf:Description rdf:about="http://example.org/">  
  <dc:title>My Home Page</dc:title>  
</rdf:Description>
```

(comentari: combinacions d'elements i atributs dels diferents namespaces)

(comentari: tots els elements definits per DC són repetibles i optatius. Ver DTD en <http://hipatia.uc3m.es/documents/2002/07/31/dcmes-xml/>)

(comentari: els elements poden estar continguts dintre d'altres elements. La jerarquia debe respondre fidelment al graf que descriu. ver XML)

Exemple de recurs amb informació en forma de Literal de tipus definit (node, arc , node-literal):

```
<rdf:Description rdf:about="http://www.example.org/index.html">  
  <externs:creation-date  
rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1999-08-16  
  </externs:creation-date>  
</rdf:Description>
```

Exemple de recurs amb informació en forma de Literal de tipus definit emprant una Entitat definida en la capçalera (node, arco, node-literal):

```
<rdf:Description rdf:about="http://www.example.org/index.html">  
  <externs:creation-date rdf:datatype="&xsd:date">1999-08-16  
  </externs:creation-date>  
</rdf:Description>
```

(comentari: ver XML)

Exemple de recurs amb informació presentada en una etiqueta no contenidora.(node , arco, node)

```
<rdf:Description rdf:about="http://www.example.org/index.html">  
  <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>  
</rdf:Description>
```

Exemple de relació transitiva amb el node central en blanc (node, arco, node en blanc, arco, node) en etiqueta no contenidora:

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">  
  <externs:editor rdf:nodeID="abc"/>  
</rdf:Description>  
<rdf:Description rdf:nodeID="abc">  
  <externs:homePage rdf:resource="http://purl.org/net/dajobe"/>  
</rdf:Description>
```

Exemple de jerarquia de etiquetes. (node arrel, arc, node fill, arc, node fill,):

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">  
  <ex:editor>  
    <rdf:Description>  
      <ex:homePage>  
        <rdf:Description rdf:about="http://purl.org/net/dajobe/">  
        </rdf:Description>  
      </ex:homePage>  
    </rdf:Description>  
  </ex:editor>  
</rdf:Description>
```

Tancament del document:

```
</rdf:RDF>
```

Consultar Gramàtica RDF/XML més avall.
Ver :
<http://www.w3.org/TR/2001/WD-rdf-syntax-grammar-20010906/>

Simplificacions i anidament d'elements.

Exemple de transformació d'elements en atributs:

```
<rdf:RDF>
  <rdf:Description about="http://www.w3.org">
    <s:Publisher>World Wide Web Consortium</s:Publisher>
    <s:Title>W3C Home Page</s:Title>
    <s:Date>1998-10-03T02:27</s:Date>
  </rdf:Description>
</rdf:RDF>

<rdf:RDF>
  <rdf:Description about="http://www.w3.org"
    s:Publisher="World Wide Web Consortium"
    s:Title="W3C Home Page"
    s:Date="1998-10-03T02:27"/>
</rdf:RDF>
```

Comentari: Les dues formes són equivalents a efectes de RDF, però el resultat presentat pels browsers pot ser diferent.

Exemple d'anidament de sentències

```
<rdf:RDF>
  <rdf:Description about="http://www.w3.org/Home/Lassila">
    <s:Creator>
      <rdf:Description about="http://www.w3.org/staffId/85740">
        <v:Name>Ora Lassila</v:Name>
        <v:Email>lassila@w3.org</v:Email>
      </rdf:Description>
    </s:Creator>
  </rdf:Description>
</rdf:RDF>
```

Comentari: remarcar les etiquetes rdf:Description contingudes dintre del mateix element rdf:Description.

Exemple de jerarquia de etiquetes: (node arrel, arc editor [, node , arc homePage, [node,] arc fullName [node literal],], arc title [node literal]). Hi ha dues *Property* de node arrel.

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
  <ex:editor>
    <rdf:Description>
      <ex:homePage>
        <rdf:Description rdf:about="http://purl.org/net/dajobe/">
          </rdf:Description>
        </ex:homePage>
        <ex:fullName>DaveBeckett</ex:fullName>
      </rdf:Description>
    </ex:editor>
  <dc:title>RDF/XML Syntax Specification (Revised)</dc:title>
</rdf:Description>
```

Exemple de arc etiquetat amb atributs. (node amb atribut, arc amb atribut,

```
<rdf:Description ex:fullName="Dave Beckett">
  <ex:homePage rdf:resource="http://purl.org/net/dajobe/" />
</rdf:Description>

<rdf:Description ex:fullName="Dave Beckett">
  <ex:homePage>
    < rdf:Description rdf:resource="http://purl.org/net/dajobe/" />
    < /rdf:Description>
  </ex:homePage>
</rdf:Description>
```

Comentari: els grafs generats serien diferents.

Exemple de arc etiquetat amb atributs. (node amb atribut, arc amb atribut, (el recurs objecte és un node en blanc amb identificador)

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar"
  dc:title="RDF/XML Syntax Specification (Revised)">
  <ex:editor rdf:nodeID="abc" />
</rdf:Description>
```

Exemple de node amb Identificador (¿node en blanc?) Amb un recurs (node objecte) presentat com atribut i arc amb atribut (node objecte presentat com atribut)

```
<rdf:Description rdf:nodeID="abc" ex:fullName="Dave Beckett">
  <ex:homePage rdf:resource="http://purl.org/net/dajobe/" />
</rdf:Description>
```

Exemple amb ommissió de rdf:Description:

```
<ex:editor rdf:parseType="Resource">
  <ex:fullName>Dave Beckett</ex:fullName>
  <ex:homePage rdf:resource="http://purl.org/net/dajobe/">
</ex:editor>

<ex:editor rdf:parseType="Resource">
  <rdf:Description >
    <ex:fullName>Dave Beckett</ex:fullName>
  </rdf:Description >
  <rdf:Description >
    <ex:homePage rdf:resource="http://purl.org/net/dajobe/">
  </rdf:Description >
</ex:editor>
```

Comentari : els dos documents contenen representacions formalment diferents, encara que la informació és la mateixa.

Exemple de predicat rdf:type, arc etiquetat amb rdf:type

```
<rdf:Description rdf:about="http://example.org/thing">
  <rdf:type rdf:resource="http://example.org/stuff/1.0/Document"/>
</rdf:Description>
```

Exemple de element typed node: (és el mateix que el anterior)

```
<ex:Document rdf:about="http://example.org/thing"></ex:Document>
```

Comentari: dubte ¿el node subjecte està representat per rdf:about, l'arc etiquetat per ex:Document i el node objecte per ?

Gramàtica RDF/XML.

```
[6.1] RDF ::= [ '<rdf:RDF>' ] obj* [ '</rdf:RDF>' ]
```

[6.2]	obj	::=	description container
[6.3]	description	::=	'<rdf:Description' idAboutAttr ? bagIdAttr ? propAttr * '>'
			'<rdf:Description' idAboutAttr ? bagIdAttr ? propAttr * '>' propertyElt *
			'</rdf:Description>'
[6.4]	container	::=	sequence bag alternative
[6.5]	idAboutAttr	::=	idAttr aboutAttr aboutEachAttr
[6.6]	idAttr	::=	' ID="' IDsymbol '''
[6.7]	aboutAttr	::=	' about="' URI-reference '''
[6.8]	aboutEachAttr	::=	' aboutEach="' URI-reference ''' ' aboutEachPrefix="' string '''
[6.9]	bagIdAttr	::=	' bagID="' IDsymbol '''
[6.10]	propAttr	::=	typeAttr propName '=' string ''' (with embedded quotes escaped)
[6.11]	typeAttr	::=	' type="' URI-reference '''
[6.12]	propertyElt	::=	'<' propName idAttr ? '>' value '</' propName '>'
			'<' propName idAttr ? parseLiteral '>' literal '</' propName '>'
			'<' propName idAttr ? parseResource propertyElt * '</' propName '>'
			'<' propName idRefAttr ? bagIdAttr ? propAttr * '>'
[6.13]	typedNode	::=	'<' typeName idAboutAttr ? bagIdAttr ? propAttr * '>'
			'<' typeName idAboutAttr ? bagIdAttr ? propertyElt * '</' typeName '>'
[6.14]	propName	::=	Qname
[6.15]	typeName	::=	Qname
[6.16]	idRefAttr	::=	idAttr resourceAttr
[6.17]	value	::=	obj string
[6.18]	resourceAttr	::=	' resource="' URI-reference '''
[6.19]	Qname	::=	[NSprefix ':'] name
[6.20]	URI-reference	::=	string , interpreted per [URI]
[6.21]	IDsymbol	::=	(any legal XML name symbol)
[6.22]	name	::=	(any legal XML name symbol)
[6.23]	NSprefix	::=	(any legal XML namespace prefix)
[6.24]	string	::=	(any XML text , with "<", ">", and "&" escaped)
[6.25]	sequence	::=	'<rdf:Seq' idAttr ? '>' member * '</rdf:Seq>'
			'<rdf:Seq' idAttr ? memberAttr * '>' member * '</rdf:Seq>'
[6.26]	bag	::=	'<rdf:Bag' idAttr ? '>' member * '</rdf:Bag>'
			'<rdf:Bag' idAttr ? memberAttr * '>' member * '</rdf:Bag>'
[6.27]	alternative	::=	'<rdf:Alt' idAttr ? '>' member + '</rdf:Alt>'
			'<rdf:Alt' idAttr ? memberAttr ? '>' referencedItem inlineItem
[6.28]	member	::=	referencedItem inlineItem
[6.29]	referencedItem	::=	'<rdf:li' resourceAttr '>'
[6.30]	inlineItem	::=	'<rdf:li' '>' value '</rdf:li>' '<rdf:li' parseLiteral '>' literal

```

</rdf:li>'
| '<rdf:li' parseResource '>'
propertyElt* </rdf:li>'
[6.31] memberAttr ::= ' rdf:_n="' string "'" (where n is an
integer)
[6.32] parseLiteral ::= ' parseType="Literal"'
[6.33] parseResource ::= ' parseType="Resource"'
[6.34] literal ::= (any well-formed XML)
    
```

Aquesta taula ens permet de generar documents RDF/XML. De moment es pot mirar de fer l'anàlisi d'un exemple:

[6.1] RDF	::= ['<rdf:RDF>'] obj *	
	['</rdf:RDF>']	
[6.2] obj	::= description	<rdf:Description ex:fullName="Dave Beckett"> <ex:homePage rdf:resource="http://purl.org/net/dajobe/"> </rdf:Description>
[6.3] description	::= '<rdf:Description'	<rdf:Description ex:fullName="Dave Beckett">
	propAttr * '>' propertyElt *	<ex:homePage
	'</rdf:Description>'	rdf:resource="http://purl.org/net/dajobe/"> </rdf:Description>
[6.10] propAttr	::= propName '=' string ''	ex:fullName="Dave Beckett"
[6.12] propertyElt	::=	<ex:homePage
	'<' propName idRefAttr ? '/>'	rdf:resource="http://purl.org/net/dajobe/">
[6.16] idRefAttr	::= resourceAttr	rdf:resource="http://purl.org/net/dajobe/"
[6.18] resourceAttr	::= ' resource="' URI-reference ''	rdf:resource="http://purl.org/net/dajobe/"

5. Glossari.

A

Arc: és un element dels grafs que identifica la relació que té un element amb un altre.

Atribut: es la descripció de una propietat d'un element.

Atribut fixe: és un atribut amb un valor fixe, que no es pot modificar.

Atribut implícit: és un atribut amb valor desconegut.

Atribut requerit: és un atribut que ha d'apareixer de manera obligada.

C

CDATA: character data. És el text no marcat. Paraula reservada de XML

CGI: comon gateway interface. Interface de pasarel·la comú. Define un método estándar de intercambiar datos entre una página web y una aplicación.

CNIDR: Agencia de localització i recuperació d'informació en la xarxa.

Comentari: text marcat amb <! - ->

CSS : Cascading Style Sheets. CSS és un estàndard del W3C.

D

Dades: són el contingut de un arxiu informàtic.

Delimitadors de marca: "<", ">", "</", "&", ";", "<!", "<?",

DTD: Document Type Declaration. És la capçalera del document i permet controlar errors.

Conté la declaració dels elements, dels atributs i de les entitats.

E

Element: és un parell de marques d'inici i final amb un significat definit.

Element vinculant: és la etiqueta <A> o qualsevol altre definit com a tal.

EMPTY: paraula reservada de DTD per que la etiqueta és buida.

Enllaç: és el mecanisme que permet vincular documents.

Entitat: és una abreviatura definida en una DTD.

ESNET: xarxa del departament d'energia dels USA.

Esquema: part de les URI que indica el camí on es troba el document.

Etiquetat : anotacions a un text.

Etiquetat procedimental: són anotacions de com cal fer un tractament de text.

Etiquetat descriptiu: són anotacions que aclaren la composició del text.

H

HTML: HyperText Markup Language. És un llenguatge de anotació.

I

IAB: Internet Architecture Board, genera los informes Request For Comments RFC

ID : paraula reservada de DTD per indicar un identificador únic.

Identificador: és un objecte que pot actuar com referència d'un recurs amb identitat. En el cas de URI, el objecte és una seqüència de caràcters amb una sintaxi definida. (2)

IDREF: paraula reservada de DTD per indicar una referència a un identificador.

IETF: Internet Engineering Task Force és un comitè de científics per Internet.

Instruccions de procés: tipus de marca amb ordres que cal executar.
ISO : Organització Internacional per a la Normalització.
#IMPLIED: paraula clau de XML que vol dir que l'element és opcional.

L

Llenguatge de marca: utilitza un sistema de marca definit. SGML, XML i HTML.
Localitzador: és el URI (URL o URN) d'un recurs.

M

Marca: és una anotació que dona informació de la naturalesa d'un text.
Markup: són les etiquetes de començament i final, entitats, comentaris i instruccions de procesament.
Metadata: informació que facilita la manipulació d'altra informació.

N

Namespace: és una referència a un document amb definicions de metadades.
NDATA: paraula reservada de DTD per fer la declaració d'entitats binàries, fitxers gràfics...
NMTOKEN: paraula reservada de DTD per indicar que el contingut de XML són números.
Node: és la representació d'un recurs en RDF.
Nom qualificat: QName
NSC : Namespace Constraint: És una norma que els documents han de seguir.
NSF: National Science Foundation, manté la xarxa NSFNET.
NSINET: xarxa de la NASA.

P

Parser: programa informàtic que pot llegir documents marcats.
#PCDATA : el contingut d'un element. És una paraula reservada de SGML.
Property Element:

R

RDF: Resource Description Language. DTD de XML que fa la descripció de tot tipus de recurs per a la seva posterior catalogació, busqueda y referència.
RDF graph: és un grafo que representa una sentència RDF.
Recurs: és qualsevol cosa que tingui identitat.
Recurs estable: és el recurs amb la propietat de ser el mateix en qualsevol context.
RFC : Request for Comments.

S

SGML: Standard Generalized Markup Language.

T

Tupla: conjunt de tres elements de la forma <URI recurs><URI propietat><URIvalor>

U

URI: Uniform Resource Identifiers , Identificadors Uniformes de Recursos.
URL: Uniform Resource Locator. És un URI.
URN: Uniform Resource Name. És un URI.

V

Vincle: és la connexió entre dos o més recursos o seccions de recursos. (Enllaç)

W

W3C: World Wide Web Consortium. És un consorci empresarial per al desenvolupament de la xarxa Internet.

X

XML: eXtensible Markup Language. És un metallenguatge.

XML/RDF: representació transportable d'un esquema RDF.

XML namespace: és una referència a un document de vocabulari XML.

XML-Schema: és un sistema de definició de llenguatges XML.

6. Bibliografia.

Dave Beckett : RDF/XML Syntax Specification (Revised). 2003

<http://www.w3.org/TR/rdf-syntax-grammar/>

Dave Beckett: Refactoring RDF/XML Syntax W3C Working Draft 06 September 2001

<http://www.w3.org/TR/2001/WD-rdf-syntax-grammar-20010906/>

Dave Beckett : Expressing Simple Dublin Core in RDF/XML

<http://hipatia.uc3m.es/documents/2002/07/31/dcmes-xml/>

T. Berner – Lee : Uniform Resource Identifier (URI): Generic Syntax.

<http://gbiv.com/protocols/uri/rev-2002/rfc2396bis.html>

Tim Bray (i altres) : Extensible Markup Language (XML) 1.0 - El lenguaje extensible de marcas (XML) 1.0 Recomendación de la W3C - Febrero 1998 Traducció al castellà: Fabio Arciniegas .

<http://www.sidar.org/recur/desdi/traduc/es/xml/xml1/index.html>

Tim Bray (I altres) : Extensible Markup Language (XML) 1.0 (Second Edition)

W3C Recommendation 6 October 2000

<http://www.w3.org/TR/REC-xml>

Tim Bray (I altres) :Extensible Markup Language (XML) 1.0 (Second Edition) Review Version
W3C Recommendation 6 October 2000

<http://www.w3.org/TR/2000/REC-xml-20001006-review.html>

Tim Bray (i altres) : Namespaces in XML World Wide Web Consortium 14-January-1999

<http://www.w3.org/TR/1999/REC-xml-names-19990114>

<http://www.w3.org/TR/1999/REC-xml-names-19990114/xml-names.xml>

<http://www.w3.org/TR/1999/REC-xml-names-19990114/Overview.html>

Dan Brickley : RDF Vocabulary Description Language 1.0: RDF Schema

W3C Working Draft 05 September 2003

<http://www.w3.org/TR/2003/WD-rdf-schema-20030905/>

(Hi ha una traducció al castellà per Eva Méndez – SIDAR)

James Clark (i altres) : Lenguaje de Caminos XML (XPath) Versión 1.0 Recomendación del W3C 16 Noviembre 1999

<http://www.w3.org/TR/1999/REC-xpath-19991116>
<http://www.sidar.org/recur/desdi/traduc/es/xml/xpath.html>

Steve DeRose (i altres) : XML Linking Language (XLink) Version 1.0 W3C Recommendation
27 June 2001 <http://www.w3.org/TR/2000/REC-xlink-20010627/>

David C. Fallside: XML Schema Part 0: Primer W3C Recommendation, 2 May 2001
<http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/primer.html>

Patrick Hayes: RDF Semantics W3C Working Draft 5 September 2003.
<http://www.w3.org/TR/2003/WD-rdf-mt-20030905/>

Graham Klyne & Jeremy J.Carroll : Resource Description Framework (RDF):
Concepts and Abstract Syntax W3C Working Draft 05 September 2003
<http://www.w3.org/TR/2003/WD-rdf-concepts-20030905/>

Stefan Kokkellink : Expressing Qualified Dublin Core in RDF / XML
<http://es.dublincore.org/documents/2002/05/15/dcq-rdf-xml/index.shtml>

Ora Lassila : Resource Description Framework (RDF) Model and Syntax Specification
W3C Recommendation 22 February 1999
<http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>

Jonathan Marsh: XML Base W3C Recommendation 27 June 2001
<http://www.w3.org/TR/2001/REC-xmlbase-20010627/>

Lluïsa Núñez : Metadades - Informe per al CBUC

Henry S. Thompson (i altres) : XML Schema Part 1: Structures W3C Recommendation 2 May
2001 <http://www.w3.org/TR/xmlschema-1/>

URI RFC2396
Naming and Addressing: URIs, URLs, ... <http://www.w3.org/Addressing/>

Garret Wilson : XML Package (XPackage) 1.0 Editor's Working Draft 6 March 2003
<http://www.xpackage.org/specification/xpackage-draft-20030306.html>

Resum

El present treball tracta de les relacions entre XML i RDF com a mitjans de descripció de recursos web.

Planteja la necessitat de la descripció i registre de documents per al seu tractament mecànic.

XML aporta una solució que s'ajusta al mode de treball tradicional de les persones mentre que la solució de RDF està pensada des d'un punt de vista de la màquina ja que només fa servir les URI dels recursos.

El treball consta d'una introducció general i tres capítols de XML, RDF i RDF/XML. La introducció fa una explicació de les URI, la informació i el W3C, i el model XML/RDF de descripció de recursos.

El capítol de XML, està centrat en la sintaxi dels llenguatges de marca per tal de crear DTD externes i internes, XML-Schema (documents xsd) i documents xml. També menciona els xml-namespaces i xlink.

El capítol de RDF tracta del model semàntic de presentació de informació de RDF, els recursos i les adreces dels recursos, i una explicació de com crear sentències ajustades al model RDF. També explica les maneres de representació de sentències RDF: els grafos, les tuples i la serialització amb RDF/XML.

El capítol de RDF/XML fa una descripció dels vocabularis disponibles: rdf, rdfs, xsd i dc. Després dona una guia del procediment de creació de documents i de les relacions entre els elements, i les simplificacions per tal de transformar els grafos en text xml. Acaba amb un quadre de la gramàtica i una aplicació pràctica.

Summary

This work is about relations between XML and RDF as a description tools for web resources.

It presents the opportunity of description and register of documents for mechanic process.

XML gives a solution that fits the traditional human working way, while RDF is intended for mechanical work because only URI are needed.

This work has four parts. A general introduction and three chapters: XML, RDF and RDF/XML.

In the first part there is one explanation about URI, the knowledge and W3C, and the XML/RDF model of resources description.

The chapter on XML, is about the markup language intended for DTD , XML-Schema (xsd documents) and xml documents writing. Xml-namespaces and xlink are also mentioned.

The chapter on RDF is about the semantic model of information representation of RDF, the resources address. How to make RDF sentences is explained. The representation of sentences with graphs, tuples and serialization with RDF/XML close chapter 3.

Chapter 4 is about XML/RDF. It explains the vocabulary we afford: rdf, rdfs, xsd and dc. Then it offers one guidance for documents writing, relations between elements and simplifications for transforming graphs to xml text. It ends with a grammar schema and how to use it.