

## Tehnologii Web avansate – WEB-ul semantic

### CUPRINS

1. Introducere
2. Un scurt istoric
3. Ce este Web-ul semantic?
4. Ce este o căutare semantică?
5. Ce este un browser semantic?
6. Structura Web-ului semantic
7. Resource Description Framework (RDF)
8. Introducere în RDFS și OWL
9. Interogare folosind SPARQL
10. Concluzii
11. Bibliografie

Anexa



## 1. Introducere

De multe ori suntem în situația de a face cumpărături și accesăm paginile web ale comercianților cu amănuntul, comparând prețurile, timpul și ratele de expediere. Putem căuta, de asemenea, un site care va compara prețurile și opțiunile de transport de la mai mulți comercianți. De fiecare dată, noi trebuie să facem o analiză pe baza informațiilor primite, apoi să luăm decizia de cumpărare și să plasăm singuri comanda.

Cu Webul Semantic, avem o altă opțiune. Putem introduce preferințele noastre folosind un agent computerizat, care ar putea căuta pe Web și el să găsească cea mai bună opțiune pentru noi și să plaseze comanda. Agentul ar putea apoi să deschidă software-ul de finanțare personală pe computerul personal și să înregistreze suma pe care am cheltuit-o și ar putea marca data la care produsele ar putea să sosească, într-un calendar. Agentul ar învăța, de asemenea, obiceiurile și preferințele noastre, așadar dacă am avea o experiență neplăcută cumpărând de la un anumit site, ar ști să nu mai folosească site-ul respectiv.

## 2. Un scurt istoric

Acum câțiva ani în urmă dacă foloseam sintagma: "*Am aflat despre asta pe Web*", toată lumea știa despre ce vorbești. Prin intermediul hiperlinkului sau prin utilizarea unui motor de căutare Web, puteam găsi un site Web cu informațiile pe care le-am dorit. Astăzi oamenii pot vorbi despre World Wide Web, Web 2.0 sau Web Semantic. Există o mulțime de concepții greșite despre asemănările și diferențele dintre ele, precum și despre tehnologiile inerente fiecărei suprapunerii, dacă ele se suprapun sau nu. Cea mai greșită concepție este că termenii Semantic Web și Web 2.0 înseamnă același lucru.

Sir Timothy John Berners-Lee este creditat cu invenția World Wide Web-ului, când în 1980 a realizat un proiect bazat pe conceptul de hipertext, pentru a facilita schimbul de informații și actualizarea informațiilor între cercetători. În 1994 a fondat Consorțiul World Wide Web (W3C). Tot Berners-Lee este, de asemenea, motivația din spatele ideii unui Web Semantic, în care ideea era să transforme Web-ul într-un singur depozit de informații în loc de o vastă colecție de site-uri Web și de pagini.

Două perioade de timp pot fi luate ca reper:

**1991 – 2003** – apare și se dezvoltă WWW-ul (World Wide Web), care este un sistem global de documente hipertext interconectate logic (prin hiperlegături, sau hiperlinks). Tot acum apare și familia de protocoale TCP/IP, ce facilitează comunicarea dintre sistemele de calcul din internet. WWW este referit retroactiv drept Web 1.0. Acest nume desemnează starea de fapt a lucrurilor dinaintea apariției Web 2.0. Web 1.0 se caracterizează prin pagini statice – HTML – folosirea de frameset-uri, guestbookuri online, message boards și altele.

**2003 – prezent** – maturizarea Web 2.0, aducând în plus aplicații web ce facilitează răspândirea de informații în mod interactiv, interoperabilitate, design având ca țintă nevoile utilizatorului și colaborarea între oameni. Servicii specifice Web 2.0: comunități bazate pe web (Twitter / Facebook, Hi5), aplicații web (e-applications), rețele sociale, siteuri de sharing video sau foto, pagini wiki, bloguri etc.

Web 2.0 este vorba despre oameni, este a doua generație a World Wide Web-ului și se axează pe capacitatea oamenilor de a colabora și de a face schimb de informații online. În cazul în care Web-ul conține pagini HTML statice, Web 2.0 este dinamic, deoarece servește aplicațiilor utilizatorilor și oferă comunicări deschise cu accent pe comunitățile Web. Web 2.0, deoarece se concentrează pe oameni și comunicații, cuprinde un număr mare de tehnologii și standarde: AJAX, Ruby, XHTML, SOAP și multe altele. Aici tehnologia este mai puțin importantă pentru oameni – lor nu le pasă de standardele și tehnologiile care rulează aceste aplicații, ci doar folosesc rezultatul final, care este interacțiunea socială într-o aplicație atractivă și ușor de utilizat.

Viziunea despre Web-ul Semantic sau Web 3.0, așa cum afirmă Sir Tim Berners-Lee, este să ca el să devină o extensie a actualului web, în care informațiile au un înțeles bine definit și este înțeles și procesat de computere. Tim Berners-Lee descrie webul semantic: „Oamenii mă întreabă ce este Webul 3.0. Cred că dacă adaugi grafică vectorizată – tot conținutul să fie dinamic, interactiv și atragător – peste Web 2.0 și oferi acces la Web Semantic peste un spațiu enorm de date, obții acces la o resursă de date incredibilă” (Tim Berners-Lee, 2006).

### 3. Ce este web-ul semantic?

**Definiție** - un termen folosit pentru a descrie aplicațiile bazate pe Web care încorporează principiile sau tehnologiile W3C Semantic Web, cum ar fi RDF, OWL și alte standarde de metadate. Web-ul Semantic este o extensie a Web-ului actual ce permite descrierea formală a resurselor existente pe Internet (pagini Web, documente text și multimedia, baze de date, servicii etc).

Dacă Web 2.0 se concentrează pe oameni, Webul Semantic se concentrează pe mașini. O extensie a site-ului Web real, oferă o modalitate mai ușoară de a găsi, împărtăși, reutiliza și combina informațiile. El se bazează pe informații care pot fi citite de mașină și se bazează pe capacitatea tehnologiei XML de a defini scheme personalizate de etichetare și abordare flexibilă a RDF (Resource Description Framework) pentru reprezentarea datelor. Webul Semantic oferă formate comune pentru schimbul de date (unde pe Web există doar un schimb de documente).

Web-ul necesită un operator uman, folosind sisteme informatice pentru a îndeplini sarcinile necesare pentru găsirea, căutarea și agregarea informațiilor sale. Este imposibil ca un computer să facă aceste sarcini fără îndrumare umană, deoarece paginile Web sunt concepute special pentru cititorii umani. Webul Semantic este un proiect care își propune să schimbe acest lucru prin prezentarea datelor de pe paginile Web într-un mod care este înțeles de computere, permițând mașinilor să facă căutarea, agregarea și combinarea informațiilor Web - fără un operator uman. Pe scurt, webul semantic ar trebui să permită ca datele, localizate oriunde pe web, să fie accesibile și înțelese atât de oameni, cât și de calculatoare.

Web-ul semantic poate fi considerat o viziune pentru viitor în care datele ar putea fi interpretate rapid de către mașini, permițându-le să efectueze numeroase sarcini plictisitoare legate de descoperirea, amestecarea și luarea de decizii asupra informațiilor disponibile pe Web. Webul Semantic este un proces care permite mașinilor să înțeleagă rapid și să reacționeze

la cererile umane complicate, în funcție de semnificația lor. Acest tip de înțelegere impune ca sursele de informare corespunzătoare să fie structurate semantic, ceea ce este o sarcină dificilă.

A fost nevoie de ani de zile pentru a pune piesele laolaltă care cuprind Web-ul Semantic, inclusiv standardizarea RDF, lansarea W3C a Web Ontology Language (OWL) și standardizarea pe SPARQL, care adaugă capabilități de interogare RDF. Deci, cu standardele și limbile existente, putem vedea tehnologiile de tip Semantic Web folosite de adoptoarele timpurii.

Cine folosește tehnologia de tip Semantic Web? Tehnologiile Web semantice sunt populare în domenii precum cercetarea și științele vieții, unde pot ajuta cercetătorii prin agregarea datelor despre diferite medicamente și boli care au nume multiple în diferite părți ale lumii. Pe Web, Twine oferă o aplicație de rețea bazată pe cunoaștere, care a fost construită cu tehnologii Web semantice.

Serviciul de televiziune online Joost utilizează de asemenea tehnologia Semantic pe backend. Aici tehnologia Semantic este utilizată pentru a ajuta utilizatorii Joost să înțeleagă relațiile dintre fragmente de conținut, permițându-le să găsească tipurile de conținut pe care le doresc cel mai mult. Oracle oferă o vizualizare Semantic Web a Oracle Technology Network, numită OTN Semantic Web.

#### 4. Ce este o căutare semantică?

Căutarea semantică este o tehnică de căutare a datelor în care o interogare de căutare are ca scop nu numai să găsească cuvinte cheie, ci și să determine intenția și semnificația contextuală a cuvintelor pe care o persoană o folosește pentru căutare. Căutarea semantică oferă rezultate de căutare mai relevante, evaluând și înțelegând fraza de căutare și găsiind cele mai relevante rezultate într-un site web, în baza de date sau în orice alt depozit de date.

Căutarea semantică funcționează pe principiile semanticii limbajului. Spre deosebire de algoritmi de căutare tipici, căutarea semantică se bazează pe contextul, substanța, intenția și conceptul expresiei căutate. Căutarea semantică include, de asemenea, locația, sinonimele unui termen, tendințele actuale, variațiile de cuvinte și alte elemente de limbă naturală ca parte a căutării. Conceptele de căutare semantică sunt derivate din algoritmi și metodologii de căutare diferite, incluzând cartografierea cuvintelor-cheie, modelele grafice și logica fuzzy.

Ideea de căutare în Web Semantic poate fi explicată foarte ușor printr-un exemplu.

Să presupunem că un utilizator dorește să afle informații despre o entitate software, din domeniul Inteligenței Artificiale, care acționează în scopul atingerii unui anumit obiectiv. După cum se observă, sunt mai multe criterii de căutare.

Prin executarea unei căutări a sintagmei *agent software* într-unul dintre motoarele de căutare cunoscute (Google, Bing, Ask, Yahoo etc), se returnează documentele în care au fost identificate cuvintele introduse în interogare. Dezavantajul motoarelor de căutare actuale este că funcționează pe bază de string matching - cu alte cuvinte identifică cuvinte specificate în query (în cazul nostru *agent software*) într-o mulțime de documente țintă, astfel că sunt returnate documentele în care se găsește cel puțin unul dintre cele două cuvinte din exemplul nostru. Totuși subiectul acestor documente s-ar putea să nu fie cel de interes pentru utilizator.

Prin urmare utilizatorul este nevoit să parcurgă și să sorteze manual rezultatele pentru a extrage doar acele documente care îl interesează. În schimb, un motor de căutare semantic ar lua în considerare întreaga sintagmă, prin urmare conceptul de agent software și nu stringul corespondent și ar returna doar rezultatele care sunt relevante pentru utilizator. Rezultatul unei

astfel de interogări semantice ar fi o descriere a conceptului de *agent software* în termeni de proprietăți pe care acest concept le are și eventual o listă de exemple. Putem duce acest raționament și mai departe și să ne imaginăm un motor de căutare semantic care să răspundă la întrebări de genul: Care sunt universitățile din regiune în a căror curiculă sunt incluse cursuri despre agenți software ?

## 5. Ce este un browser semantic?

Un browser web semantic este un tip de browser care poate naviga pe Web-ul semantic, o extensie a World Wide Web care lucrează pentru a lega diferite date și surse de date. Browserele Web semantice pot fi separate sau făcute ca extensii ușoare la browserele Web standard, oferind posibilitatea de a naviga pe Web semantic. Browserele de tip semantic sunt, de asemenea, cunoscute sub numele de browsere hyperdata. Browserele semantice sunt capabile să ofere căutări inteligente, ce folosesc principiile sinonimiei și ale stemmingului, clusterizări semantice ale răspunsurilor, capacitatea de a face sugestii de căutare sau sumarizări automate.

Browserul semantic permite unui utilizator să navigheze pe o rețea de date interconectate în același mod în care un browser Web standard permite utilizatorului să viziteze site-urile text standard prin link-uri. Cu toate acestea, în loc de formatul documentului HTML, browserele Web semantice utilizează Formatul de descriere a resurselor (RDF), care se bazează foarte mult pe metadata pentru a defini semantica, semnificațiile și definițiile legăturilor de date. Din acest motiv, browserele web semantice sunt numite și browsere hyperdata, care sunt similare cu browserele hipertext.

Exemple de browsere web semantice:

- Ripple
- Zitgist
- Marbles
- Elda
- Facet

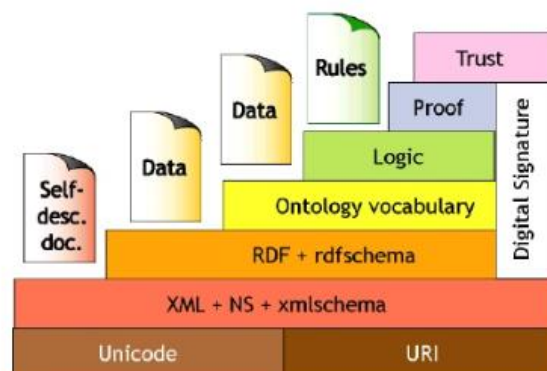
## 6. Structura web-ul semantic

Deodată a devenit șic să eticheteze fiecare aplicație nouă ca fiind oarecum "semantică", dar ce înseamnă aceasta într-adevăr? Sunt toate aplicațiile "semantice" parte din "Web-ul semantic"? Care sunt criteriile pentru a diferenția termenul de "semantic" versus "Semantic Web"?

Orice aplicație care poate înțelege într-o oarecare măsură un limbaj ar putea fi etichetată ca "semantică". Deci chiar și Google este într-o oarecare măsură o aplicație semantică ținând cont de acest criteriu. Desigur, unele aplicații sunt mult mai semantice decât altele. Powerset este mai semantic decât Google, de exemplu, deoarece înțelege limbajul natural, nu doar cuvintele cheie.

Pentru ca o aplicație să fie considerată parte a "Web-ului Semantic", ea trebuie să se sprijine pe un set de standarde deschise definite de W3C, inclusiv cel puțin RDF și, posibil, și OWL și SPARQL. Acestea sunt tehnologiile folosite de Web-ul Semantic. Orice aplicație care publică cel puțin datele RDF poate fi considerată atât ca fiind semantică, cât și ca parte a Webului Semantic.

Pentru a clarifica conceptul prezentat, vom face afirmația că Web-ul semantic adoptă o viziune stratificată:



Niveluri de specificare a Web-ului semantic [Berners-Lee, 2000]

Pentru a explica aceste concepte, să luăm un exemplu din viața noastră – *Ștefan cel Mare și Petru Rareș sunt ambii domnitori și există o relație între ei*. Știm că un tată este un tip de dată părinte și că cunoaștem ce înseamnă și noțiunea de fiu. Dar un calculator nu se poate gândi la asta fără ajutor. Pentru a permite unui calculator să înțeleagă ce înseamnă această propoziție, va trebui să adăugăm informații care pot fi stocate de computer, care descriu cine sunt cele două personaje amintite și care este relația lor. Aceasta începe cu două instrumente - limbajul eXtensible Markup Language (XML) și Resource Description Framework (RDF).

La baza piramidei găsim standardul Unicode folosit în reprezentarea și manipularea textului în diferite limbi precum și standardul de construcție a URI-urilor, util pentru identificarea resursele publicate pe Web (documente multimedia, pagini Web, bloguri etc.).

Stocarea informațiilor semantice se poate face și sub formă de Uniform Resource Identifier (URI), adică un string de identificare ce poate fi de natura unei locații (URL) sau unui nume (URN). URN-ul este asemănător în concepție cu numele unei persoane, pe când URL-ul este asemănător cu adresa sa. Un exemplu elocvent este cazul cărților, cu catalogarea în funcție de ISBN.

În prezent, standardul pentru descrierea și transferul de date pe Web este XML. Extensible Markup Language (XML) este un meta-limbaj de marcare recomandat de Consorțiul Web pentru crearea de alte limbaje de marcare, cum ar fi XHTML, RDF, RSS, MathML, SVG, OWL etc. Aceste limbaje formează familia de limbaje XML. Stratul (XML) folosește scheme XML și se asigură că există o sintaxă comună utilizată în web-ul semantic.

XML este un limbaj de markup general pentru documente care conțin informații structurate. XML este un limbaj de markup la fel ca limbajul de hipertext (HTML). HTML guvernează aspectul informațiilor pe care le privim pe Web. XML completează (dar nu înlocuiește) codul HTML, adăugând etichete care descriu datele. Aceste etichete sunt invizibile pentru persoanele care citesc documentul, dar sunt vizibile pentru computere. Etichetele sunt deja utilizate pe Web și ele sunt citite de programele care colectează datele pentru motoarele de căutare.

Spre deosebire de HTML, care utilizează etichete standard ca <b> pentru caractere aldine și <u> pentru subliniere, XML nu are etichete standard. Acest lucru este util - permite dezvoltatorilor să creeze etichete unice pentru scopuri specifice. Dar înseamnă că un browser nu știe automat ce înseamnă tag-ul. Un spațiu de nume XML este în esență un document care



spune aplicațiilor semnificația tuturor etichetelor dintr-un alt document. Creatorul unui document XML declară spațiul de nume la începutul documentului cu o linie de cod. În exemplul nostru, declarația noastră privind spațiul de nume ar arăta astfel:

```
<rdf: RDF xmlns: hsw = https: //www.howstuffworks.com/example/RDF/relationship;
```

Această linie de cod spune computerului: "Toate etichetele pe care le vedeți începând cu 'hsw' utilizează vocabularul găsit în acest document. Puteți căuta orice etichetă care începe cu 'hsw' aici." Astfel, oamenii pot crea etichetele XML de care au nevoie pentru un document fără a intra în conflict cu alte documente XML pe Web.

XML este deci un model de stocare a datelor nestructurate și semi-structurate în cadrul bazelor de date native XML. Totuși limbajul XML oferă unele limitări. Printre acestea, putem aminti lipsa unei semantici formale a Schemelor XML, care face greoaie comunicarea între aplicații sau servicii care nu partajează aceeași schemă sau aceeași interpretare a unei scheme. Din momentul ce, folosind numai marcatori XML, nu putem descrie resursele, un prim pas este asocierea de metadate, adică date privitoare la formate.

Din dorința de a adăuga un anumit nivel de semantică limbajului XML s-a născut limbajul RDF, un limbaj de descriere a resurselor folosind structuri asemănătoare grafurilor. Pentru a organiza resursele și proprietățile într-o ierarhie de tipuri se folosește RDF Schema.

Limbajul OWL extinde limbajul RDF(S) introducând o serie de constructori mai avansați care permit descrieri mai expresive decât cele posibile folosind triplete RDF(S). OWL permite de asemenea definirea de constrângeri asupra proprietăților cum ar fi constrângerile de cardinalitate, restricțiile de valori sau impunerea unor caracteristici predefinite pentru proprietăți (i.e. tranzitivitate, simetrie etc.). De remarcat este faptul că limbajele RDF(S) și OWL au la bază teoria logicilor de descriere, ceea ce garantează o interpretare semantică non-ambiguă a declarațiilor făcute în aceste limbaje. Mai mult, folosind motoarele de inferență specifice logicilor de descriere, cunoștințe noi pot fi derivate automat din descrieri RDF(S) sau OWL.

## 7. Resource Description Framework (RDF)

RDF este un cadru general pentru descrierea metadatelor unui site web sau a informațiilor despre informațiile de pe site. El oferă interoperabilitate între aplicații care fac schimb de informații pe internet. Informațiile despre detaliile RDF, cum ar fi sitemap-ul site-ului, datele despre momentul în care au fost făcute actualizări, cuvintele cheie pe care motoarele de căutare le caută și drepturile de proprietate intelectuală ale paginii web.

Un exemplu de cod scris în XML, folosind RDF:

```
01. <?xml version="1.0" encoding="UTF-8"?>
02.
03. <rdf:RDF
04.   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
05.   xmlns:dc="http://purl.org/dc/elements/1.1/"
06.   xmlns:region="http://www.country-regions.fake/">
07.
08.   <rdf:Description rdf:about="http://en.wikipedia.org/wiki/Oxford">
09.     <dc:title>Oxford</dc:title>
10.     <dc:coverage>Oxfordshire</dc:coverage>
11.     <dc:publisher>Wikipedia</dc:publisher>
12.     <region:population>10000</region:population>
13.     <region:principaltown rdf:resource="http://www.country-regions.fake/oxford"/>
14.   </rdf:Description>
15.
16. </rdf:RDF>
```

Dezvoltat sub îndrumarea consorțiului World Wide Web, RDF a fost conceput pentru a permite dezvoltatorilor să construiască motoare de căutare care se bazează pe metadate și să permită utilizatorilor de Internet să împărtășească mai ușor informații despre site-urile web. RDF se bazează pe XML ca o sintaxă de schimb, creând un sistem ontologic pentru schimbul de informații pe Web. Influența RDF se poate întrevădea în *structurarea inteligentă a documentelor* (realizată prin intermediul limbajelor bazate pe XML), în *reprezentarea cunoștințelor* (*knowledge representation*), în *standardizarea Web-ului*.

Pentru a facilita definirea datelor RDF, va fi necesar un sistem de clase similar celui din programarea orientată-obiect. O colecție de clase (dezvoltată pentru un anumit scop specific) se numește *schemă*. Clasele sunt organizate ierarhic oferind extensibilitatea prin rafinarea subclaselor. Astfel, pentru crearea unei noi scheme, putem pleca de la o schemă de bază (privită ca o clasă abstractă în termenii programării obiectuale).

Datele folosite de aplicațiile web sunt stocate în baze de date relaționale. Rândurile descriu un obiect, coloanele reprezintă proprietăți ale obiectelor, celula reprezintă o valoare a unei proprietăți a unui obiect.

CNP	Nume_Prenume	Ocupatie	Localitate
1880211..	Popescu Ion	Student	Bucuresti
1871121..	Ionescu Mihai	Student	Cluj
1871002..	Ene Marius	Student	Timisoara
1880203..	Ionita George	Student	Iasi

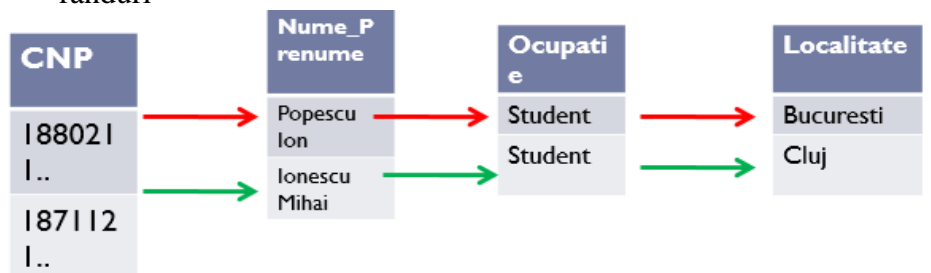
Cum putem să distribuim datele pe Web?

*Varianta 1:*

- Separăm rânduri diferite pe mai multe calculatoare
- Trebuie asigurată gestionarea distribuției informației
- Trebuie să asigurăm un model relațional comun

*Varianta 2:*

- Distribuim datele pe coloane
- Trebuie asigurat un mod prin care să facem legătura între datele de pe aceleași rânduri



*Varianta 3:*

- Distribuim informații la nivel de celulă



- Fiecare calculator reține un număr de celule și un mod de a identifica obiectul la care se referă informațiile
- Nevoia de a identifica în mod unic
  - Obiectele descrise (rândurile)
  - Proprietățile descrise (coloanele)
  - Valorile efective

Modelul RDF se referă la modalitatea 3 de distribuire a datelor. El permite specificarea resurselor de tip tripletă (coloană, rând, valoare), unde coloana/rând/valoare sunt identificate prin URI-uri (URI = Uniform Resource Identifier).

Există următoarele asocieri:

- ▶ Resursă  $\approx$  Subiect  $\approx$  Rând
- ▶ Predicat  $\approx$  Coloană  $\approx$  Proprietate

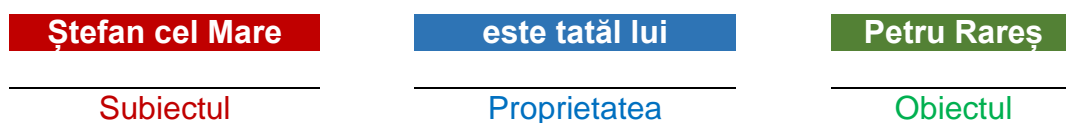
Modelul formal de bază l-a constituit logica cu predicate, relațiile dintre resurse fiind modelate prin intermediul predicatelor. Conceptul central se axează asupra *claselor* (*classes* sau *frames*) având asociate diverse proprietăți denumite *atribute*. Atributele pot fi aplicate numai asupra claselor pentru care au fost definite. Putem asocia același nume de atribut unor clase diferite, impunând anumite *restricții* ale valorilor atributelor considerate (*range restrictions*). O clasă oferă un context pentru modelarea unui aspect specific al unui *domeniu* (*domain*).

[Horrocks *et al.*, 2000, Davies *et al.*, 2003].

Modelul de bază se construiește cu ajutorul următoarelor tipuri de obiecte:

- *Resurse* - Datele descrise de expresiile RDF sunt denumite *resurse*. O resursă poate fi o pagină Web completă, o parte a unei pagini Web (un element specific HTML sau XML prezent în sursa documentului, *de ex.* o imagine) sau un obiect care nu este direct accesibil via Web (o carte tipărită). Resursele sunt specificate de URI-uri plus un identificator de legătură, opțional.
- *Proprietăți* - O *proprietate* reprezintă un aspect specific, o caracteristică, un atribut sau o relație pentru a descrie o resursă. Fiecare proprietate posedă o semantică, un set de valori permise, o mulțime de tipuri de resurse pe care le descrie și un set de relații (interdependente) cu alte proprietăți.
- *Declarații* - O anumită resursă împreună cu o proprietate a sa având asociată o valoare formează o *declarație*. Putem privi declarația ca un triplu  $\{ \text{subiect}, \text{predicat}, \text{obiect} \}$ . Obiectul declarației (valoarea proprietății) poate desemna o altă resursă (specificată de un URI) sau un *literal* (tip primitiv de dată sau șir de caractere, conform specificațiilor XML). În modelul RDF, un *literal* poate conține marcate XML care însă nu vor fi evaluate (analizate) de procesorul RDF.

RDF însuși servește ca o descriere a unei diagrame formata de triplete. Oricine poate defini vocabularul de termeni utilizați pentru o descriere mai detaliată.

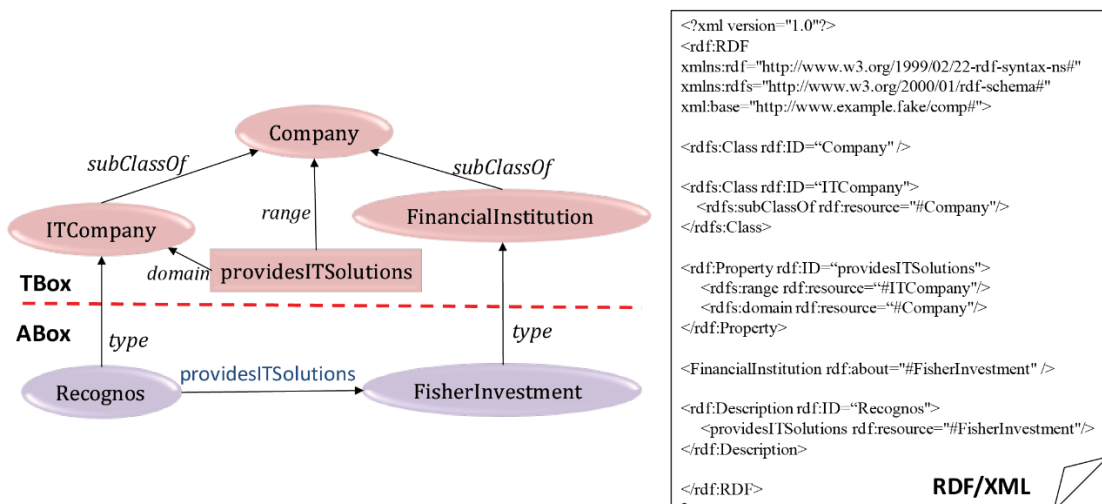


Până acum, în acest exemplu, calculatorul știe că există două obiecte în această propoziție și că există o relație între ele.

Pentru a face acest lucru, RDF folosește triple scrise ca etichete XML pentru a exprima aceste informații ca un grafic. Aceste triple constau dintr-un subiect, proprietate și obiect, care sunt ca subiect, verb și obiect direct al unei propoziții. (Unele surse le numesc subiectul, predicatul și obiectul.) RDF există deja pe Web - de exemplu, face parte din crearea fluxului RSS.

În mod tradițional datele pot fi reprezentate ierarhic într-o structură XML, sau relațional într-o bază de date. Web-ul Semantic introduce o nouă organizare a datelor sub forma grafurilor RDF (Resource Description Framework). RDF este un model simplu de descriere a resurselor de pe Web (documente multimedia, bloguri, pagini Web, servicii Web, baze de date, etc) și a meta-datelor asociate cu acestea din urmă cum ar fi titlul, autorul, data publicării etc. RDF descrie o resursă prin prisma proprietăților pe care aceasta le are.

O colecție de propoziții în format RDF reprezintă intrinsec un graf orientat și etichetat. De aceea, un model de date în format RDF este mai potrivit pentru reprezentarea cunoștințelor decât un model relațional sau alte obiecte ontologice. În practică, însă, datele sub forma RDF sunt de fapt stocate sub formă de baze de date relaționale sau reprezentări native numite Triplestores sau Quadstores în cazul în care pentru fiecare RDF se păstrează și contextul (de exemplu numele grafului).



Exemplu de graf RDF(S) simplu ce descrie două tipuri de companii între care pot exista relații de tipul furnizare de servicii IT. Graful este descris folosind sintaxa grafică (informală) și sintaxa RDF/XML.

Vocabularul RDF, așa cum apare în specificațiile limbajului, este format din:

- rdf:type: un predicat este folosit pentru a afirma că o resursă este o instanță a unei clase
- rdf:XMLLiteral: clasa obiectului scris sub formă literală
- rdf:Property: clasa de proprietăți
- rdf:Alt, rdf:Bag, rdf:Seq: – sunt, în ordine, container de alternative, container de obiecte neordonate, container de obiecte ordonate
- rdf:List: – clasa pentru obiecte de tipul listă RDF
- rdf:nil: – o instanță a rdf:List ce reprezintă o listă vidă.
- rdf:Statement, rdf:subject, rdf:predicate, rdf:object – folosite pentru ipotetizare sau reformulare.

*Definiție - Ce înseamnă o bază de date RDF?*

Nu are sens să adăugăm semantică datelor dacă nu oferă beneficii semnificative. Să ilustrăm folosind un exemplu simplu.

Două site-uri web au fost construite independent unul față de altul. Un site găzduiește informații despre filmele curente și istorice premiate cu Oscar; celălalt este o bază vastă de biografii ale actorilor și actrițelor de la Hollywood. Ambele conțin informații complementare în bazele de date ale site-urilor web. Ce se întâmplă în cazul în care schimbul de informații între aceste site-uri s-ar putea întâmpla fără utilizarea semanticii?

Resursele de descriere a bazei de date (RDF) sunt motoare care standardizează limbajul de interogare SPARQL. Aceste baze de date necesită un limbaj de interogare mai avansat decât SQL, pentru a face posibilă interogarea semantică a datelor, pentru a apropia lumea de conceptul de web semantic. SPARQL nu este numai eficient în interogări semantice, ci și în interfața cu datele. Bazele de date RDF pot face setarea procesării și, în același timp, procesarea graficelor.

Bazele de date RDF funcționează asupra conceptului de formulare a declarațiilor referitoare la resurse, în special la resursele web, ca expresii de subiect-predicate-obiecte. Aceste expresii se numesc triple în nomenclatura RDF. Subiectul semnifică resursa, iar predicatul semnifică trăsături ale resursei și definește o relație între obiect și subiect.

Bazele de date RDF sunt o soluție NoSQL construită pe un model de date uniform și simplu. NoSQL este un model de bază de date definit slab, care este non-relațional, open source și scalabil orizontal. Bazele de date RDF oferă diferite avantaje care constau în portabilitatea datelor, fiind dovada viitoare și nu necesită blocarea produsului.

## **8. Introducere în RDFS și OWL**

Un alt obstacol pentru Webul Semantic este că computerele nu au un fel de vocabular la fel ca oamenii. Oamenii folosesc limbajul întreaga viață, deci este probabil ușor să înțelegem conexiunile între diferite cuvinte și concepte și să deducem semnificații bazate pe contexte. Din păcate, un calculator nu poate primi un dicționar, un almanah și un set de enciclopedii și să poată să învețe toate acestea singur. Pentru a înțelege ce înseamnă cuvintele și care sunt relațiile dintre cuvinte, calculatorul trebuie să aibă documente care descriu toate cuvintele și logica pentru a face conexiunile necesare.

În Webul Semantic, acest lucru se implementează cu ajutorul schemei sau a ontologiei. Acestea sunt două instrumente asociate pentru a ajuta un calculator să înțeleagă vocabularul uman. O ontologie este pur și simplu un vocabular care descrie obiectele și modul în care se relaționează unul cu celălalt. O schemă este o metodă de organizare a informațiilor. Ca și în cazul etichetelor RDF, accesul la scheme și ontologii este inclus în documente ca metadate, iar creatorul unui document trebuie să declare care ontologii sunt menționate la începutul documentului.

Nu există o distincție clară între ceea ce se numește "vocabular" și "ontologie". Tendința este de a folosi cuvântul "ontologie" pentru o colecție mai complexă și, eventual, destul de formală de termeni, în timp ce "vocabularul" este folosit atunci când un astfel de formalism strict nu este neapărat utilizat sau doar într-un sens foarte slab.

Definiții mai actuale afirmă că o ontologie reprezintă un catalog al conceptelor existente într-un domeniu. Ontologia conține predicate, semantica termenilor și a conceptelor, precum și relațiile dintre acestea (Sowa 2001). Ontologie este formată din clase (concepte) și sloturi (roluri sau proprietăți) și restricții pe aceste sloturi. O bază de cunoștințe este formată dintr-o ontologie împreună cu un set de instanțe ale claselor.

Printre primele definiții utilizate pentru definirea Ontologiei se numără cea a lui (Gruber, 1993) "o ontologie este o specificare explicită a unei conceptualizări" și (Borst et al., 1997) "Ontologiile sunt definite ca a formalizarea unei concepții comune". Ulterior, (Staab și Studer, 2004) au explicat ce acești termeni înseamnă că "conceptualizarea se referă la un model abstract al unor fenomene din lume prin identificarea conceptelor relevante ale fenomenului respectiv. Explicit înseamnă că tipul de conceptele utilizate și constrângerile legate de utilizarea lor sunt definite în mod explicit. Formal se referă la faptul că ontologia trebuie să poată fi citită de mașină."

Ontologiile sunt create peste structura RDF care oferă doar o modalitate de a descrie Web-ul resurse prin crearea de metadata. Ontologiile oferă o descriere suplimentară a structurii conceptelor și relațiile dintre concepte, declarații de clase și subclase, tipuri și cardinalitate pentru proprietățile conceptelor. Aceasta înseamnă că ontologiile se bazează pe concepte specifice de logică descriptivă (DL).

Cunoscând ce este o ontologie ne putem întreba cum putem să o creăm.

Cu ajutorul RDF pot fi descrise mai multe tipuri de resurse predefinite dar pot fi definite și tipuri noi de resurse cu ajutorul unei extensii numite RDF Schema. RDF Schema (RDFS) permite definirea unor clase, instanțe și proprietăți utilizând sintaxa RDF. De asemenea, permite definirea unor relații între resurse prin oferirea posibilității de a defini subclase, subproprietăți, precum și domeniu și co-domeniu pentru proprietăți.

Limbajul descrierii vocabularului RDF 1.0 (sau RDF Schema sau RDFS) este un limbaj care poate fi utilizată pentru a defini vocabularul (adică termenii) pentru a fi utilizat într-un RDF grafic. Limbajul descrierii vocabularului RDF 1.0 este un limbaj de definire a ontologiei (simplu, comparativ cu OWL; putem doar să definim taxonomii și face unele inferențe de bază despre ele).

Conceptele RDFS sunt ele însele furnizate sub forma unui vocabular RDF; care este, ca un set specializat de resurse RDF predefinite cu ajutorul acestor semnificații speciale. Resursele din vocabularul RDF Schema au URIs cu prefixul `http://www.w3.org/2000/01/rdf-schema#` (asociat cu prefixul `QName rdfs:`). Ceea ce trebuie de reținut e că RDFS este RDF! Acesta este un punct cheie. Nu există o sintaxă specială, nimic nou pe care trebuie să-l învățăm. RDFS este exprimată ca RDF.

Să luăm un exemplu:

@prefix csipeople: <<http://www.cambridgesemantics.com/people/about/>>.

@prefix csi: <<http://www.cambridgesemantics.com/>>.

@prefix rdfs: <<http://www.w3.org/2000/01/rdf-schema#>>.

@prefix rdf: <<http://www.w3.org/1999/02/22-rdf-syntax-ns#>> .

@prefix foaf: <<http://xmlns.com/foaf/0.1/>> .

@prefix xsd: <<http://www.w3.org/2001/XMLSchema#>> .

csipeople:David

rdf:type foaf:Person ;

rdfs:label "David Tester" ;

rdfs:SeeAlso <<http://www.linkedin.com/pub/david-tester/33/737/2aa>>

## OWL

OWL reprezintă limbajul W3C Web Ontology. OWL este un limbaj bazat pe logică, conceput pentru a fi interpretat de mașini, astfel încât acestea să poată procesa și raționa automat prin informațiile pe care le descrie.

Datele descrise de o ontologie în familia OWL sunt interpretate ca un set de elemente ce au o serie de „afirmații despre proprietăți”, care relaționează aceste elemente între ele. O ontologie reprezintă un set de axiome care instalează legături între elemente (numite clase) și specifica tipurile de relații dintre ele. Aceste axiome oferă înțeles semantic, permițând sistemelor să formuleze informații adiționale pe baza datelor prezentate explicit.

Un exemplu de cod ce folosește OWL:

```
01. <rdf:RDF
02.   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
03.   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
04.   xmlns:owl="http://www.w3.org/2002/07/owl#"
05.   xmlns:dc="http://purl.org/dc/elements/1.1/"
06.
07.   <!-- OWL Header Example -->
08.   <owl:Ontology rdf:about="http://www.linkeddatatools.com/plants">
09.     <dc:title>The LinkedDataTools.com Example Plant Ontology</dc:title>
10.     <dc:description>An example ontology written for the LinkedDataTools.com RDFS &
11.       OWL introduction tutorial</dc:description>
12.   </owl:Ontology>
13.
14.   <!-- OWL Class Definition Example -->
15.   <owl:Class rdf:about="http://www.linkeddatatools.com/plants#planttype">
16.     <rdfs:label>The plant type</rdfs:label>
17.     <rdfs:comment>The class of plant types.</rdfs:comment>
18.   </owl:Class>
19. </rdf:RDF>
```

Web Ontology Language are trei sub-limbaje distincte OWL Lite, OWL DL și OWL Full ce diferă prin ceea ce pot exprima.

OWL Lite suportă restricții de cardinalitate (0 sau 1) precum și restricții de tip „AllValuesFrom” și „SomeValuesFrom”. Restricțiile de cardinalitate specifică numărul de proprietăți de un anumit tip pe care le poate avea o clasă. Restricțiile de tip „SomeValuesFrom” și „AllValuesFrom” specifică faptul că unele sau toate valorile proprietății pe care se aplică restricția au tipul specificat. De asemenea, se poate specifica faptul că unele clase și proprietăți sunt echivalente cu alte clase și proprietăți, ceea ce poate fi foarte util pentru a arăta că unele concepte cu nume diferite din ontologii diferite reprezintă de fapt același lucru. Tot în OWL Lite se poate specifica faptul că o proprietate poate fi funcțională (ceea ce înseamnă că un element din domeniu poate fi asociat printr-o proprietate funcțională unui singur element din co-domeniu), simetrică, tranzitivă sau inversă.

OWL DL (Description Logic) este un limbaj mai avansat, bazat pe un subset decidabil al logicii cu predicate. OWL DL permite definirea unor relații de disjuncție între clase. El permite și definirea de clase prin reuniune, diferență și intersecție de clase. Permite și definirea unor restricții de cardinalitate mai avansate, astfel că orice număr nenegativ poate fi specificat în cadrul restricției. Pe o ontologie OWL DL, motoare de raționament ca de exemplu Racer (Racer 2007), Fact++ (Fact 2007), Kaon2 (Kaon 2007) sau Pellet (Pellet 2007) pot verifica consistența modelului sau poate să realizeze clasificare automată.

OWL Full permite definirea mai multor tipuri de constrângeri decât OWL DL dar fără a oferi garanții computaționale. În momentul actual, cele mai multe aplicații și instrumente de dezvoltare sunt construite pentru OWL DL.

OWL este un limbaj conceput pentru a defini ontologii. Este un limbaj care extinde RDF, permițând folosirea unor instrumente de inferență pe datele din ontologie. O ontologie OWL este compusă din clase, instanțe de clase și relațiile dintre aceste instanțe. Succesul unei

ontologii depinde în mare măsură de abilitatea de a raționa despre indivizii unei clase. OWL oferă posibilitatea descrierii, via construcții XML, a claselor din care fac parte indivizi specifici și a proprietăților care-i caracterizează. În *OWL Full* o clasă reprezintă un obiect din lumea reprezentată de ontologia modelată. Filosofia limbajului este diferită de aceea a unui sistem de baze de date, neimpunându-se *a-priori* o cunoaștere completă a domeniului de interes. Clasele și proprietățile pot fi specificate în forme diferite, iar declarațiile privitoare la indivizi nu trebuie obligatoriu să fie incluse în același document în care s-a specificat o anumită ontologie.

## 9. Interogări folosind SPARQL

Dacă veniți dintr-un mediu IT tradițional, este posibil să fiți deja familiarizați cu SQL (Structured Query Language) care este folosit pentru a prelua date dintr-o bază de date relațională cum ar fi MS SQL sau MySQL.

În mod similar, bazele de date RDF pot fi, de asemenea, interogate utilizând propriul limbaj de interogare - SPARQL (SPARQL Protocol și RDF Query Language, pronunțat "sparkle"). SPARQL este un standard W3C și este în prezent la versiunea 1.1.

Aceste baze de date necesită o limbă de interogare mai avansată decât SQL, pentru a face posibilă interogarea semantică a datelor pentru a apropia lumea de conceptul de web semantic. SPARQL nu este numai eficient în interogări semantice, ci și în interfața cu datele.

SPARQL se bazează pe crearea de șabloane ce vor fi căutate în interiorul unui graf de cunoștințe; el permite definirea de subgrafuri prin specificarea unei succesiuni de triplete și caută să potrivească subgraful în graful de cunoștințe. Sintaxa SPARQL este foarte asemănătoare cu sintaxa comenzii SQL Select.

Un exemplu de interogare folosind sintaxa SPARQL:

```
1. PREFIX sch-ont: <http://education.data.gov.uk/def/school/>
2. SELECT ?name WHERE {
3.   ?school a sch-ont:School.
4.   ?school sch-ont:establishmentName ?name.
5.   ?school sch-ont:districtAdministrative <http://statistics.data.gov.uk/id/local-authority-
6.     district/00AA>.
7. }
8. ORDER BY ?name
```

Ca și SQL, SPARQL selectează datele din setul de date de interogare utilizând o instrucțiune SELECT pentru a determina ce subset de date selectate este returnat. De asemenea, SPARQL utilizează o clauză WHERE pentru a defini modele de grafuri pentru a găsi o potrivire în setul de date de interogare.

Un model de grafic într-o clauză SPARQL WHERE constă în subiectul, predicatul și obiectul triplu pentru a găsi o potrivire pentru date.

Instrucțiunea SELECT cere ca numele variabilei să fie returnat.

Prin urmare, în interogarea SPARQL de mai sus, "nume" returnează toate numele școlilor care corespund celor trei modele de căutare date în interogare. Dacă vrem, am putea face interogarea mai specifică adăugând criterii de potrivire suplimentare. Sau am putea să o facem mai largă, de exemplu, prin eliminarea ultimului model de căutare care impune școlii să se potrivească cu valoarea administrativă a districtului "00AA".



## 10. Concluzii

Ideea de bază din spatele Web-ului Semantic este să se folosească un nou set de standarde pentru a adnota paginile web cu fragmente mici de XML. Acești mici biți ai XML nu ar avea niciun efect asupra prezentării paginii web, dar ar putea fi citiți de programele software.

În concluzie, cand vorbim despre Webul Semantic ne referim la urmatorul set de caracteristici:

**Date accesibile calculatoarelor** – webul semantic presupune ideea de a avea datele pe web definite și legate în modalități care să permită calculatoarelor să le utilizeze nu numai în scopul afișării lor, dar mai ales pentru automatizarea, integrarea și utilizarea lor în diferite aplicații;

**Agenți inteligenți** – scopul webului semantic este să facă webul actual mai ușor de citit de calculatoare, pentru a permite agenților inteligenți artificiali să restabilească și să manipuleze informații;

**Baze de date distribuite:** webul semantic să facă pentru date ceea ce HTML-ul a făcut pentru informațiile textuale. Să poată să reprezinte toate bazele de date și regulile logice ale acestora ca un tot. Este o încercare de a face pentru datele prelucrabile de către calculatoare ceea ce WWWul a făcut pentru documentele citibile de către oameni.

## 11. Bibliografie

Buraga S.C. – Semantic Web, fundamente și aplicații, Ed. Matrix Rom, 2004

Gruber, T. R. (1993) - A translation approach to portable ontology specifications. Knowledge Acquisition, 5:199–220.

Staab, S. and Studer, R. (2004) - Handbook on ontologies. International handbooks on information systems. Springer-Verlag.

Varlan, Simona-Elena, (2011) - Query Evaluation of Ontology Data Model in E-commerce Organizations, 17th IBIMA Conference, Milan, Italy

<https://www.techopedia.com/definition/27961/semantic-web>

<https://ro.esdifferent.com/difference-between-rdf-and-owl>

[https://en.wikipedia.org/wiki/Semantic\\_Web](https://en.wikipedia.org/wiki/Semantic_Web)

<http://www.linkeddatatools.com/querying-semantic-data>

<http://www.w3.org/2003/08/owl-systems/test-results-out>

[http://en.wikipedia.org/wiki/Web\\_Ontology\\_Language\\_4](http://en.wikipedia.org/wiki/Web_Ontology_Language_4)

[http://en.wikipedia.org/wiki/Resource\\_Description\\_Framework](http://en.wikipedia.org/wiki/Resource_Description_Framework)

<https://www.todaysoftmag.com/article/82/semantic-web-scurta-introducere>

## **Anexa 1**

### **Tutorial – Introducere în RDF**

Acest set de lecții este o introducere despre RDF, modelul de bază al datelor Web-ului Semantic și fundamentul tuturor celorlalte tehnologii Web de tip Semantic. Lecțiile introduc RDF, prezintă detalii suplimentare și apoi plasează RDF în contextul altor tehnologii precum XML și JSON. Acoperirea este destinată începătorilor noi pentru RDF, însă un fundal tehnic vă va ajuta foarte mult să urmați materialul.

În particular, RDF este modelul de date folosit de Webului Semantic. Aceasta înseamnă că toate datele din tehnologiile Web semantice sunt reprezentate ca RDF. Dacă stocați date Semantic Web, acestea folosesc modelul RDF. Dacă interogați date Web Semantic (de obicei, folosind SPARQL), acestea sunt date RDF.

#### **Obiective**

- ✓ Ce RDF este și care sunt diferențele fundamentale dintre XML și bazele de date relaționale
- ✓ Ce se înțelege printr-un "model de date grafic"
- ✓ Cum este RDF reprezentat vizual
- ✓ Importanța URI și semnificația (sau lipsa acesteia) a identității "universalității"

#### **Ce este RDF?**

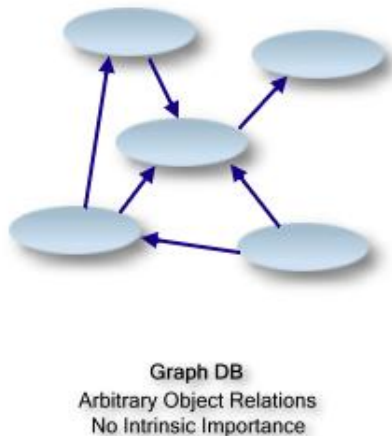
- RDF este reprezentat de cadrul de descriere a resurselor
- RDF este un cadru pentru descrierea resurselor pe web
- RDF este conceput pentru a fi citit și înțeles de computere
- RDF nu este proiectat pentru a fi afișat oamenilor
- RDF este scris în XML
- RDF face parte din Activitatea Web Semantică a W3C
- RDF este o recomandare a W3C din data de 10 februarie 2004

RDF este proiectat pentru a fi citit de computere, descrierile RDF nu sunt concepute pentru a fi afișate pe web.

Documentele RDF sunt scrise în XML. Limba XML utilizată de RDF se numește RDF / XML. Folosind XML, informațiile RDF pot fi schimbate cu ușurință între diferite tipuri de calculatoare, utilizând diferite tipuri de sisteme de operare și limbaje de aplicație.

#### **Model de date sub formă de graf**

În special, este un grafic etichetat. Prin urmare, vă puteți gândi la RDF ca la o mulțime de noduri (punctele) conectate între ele de muchii (liniile), dar unde nodurile și muchiile grafului au etichete.



Nodurile graficului sunt de formă ovală și dreptunghice. Marginile sunt etichetate cu săgeți care leagă nodurile unul de altul. Etichetele sunt URI.

Există trei tipuri de noduri într-un grafic orientat RDF:

**Resurse noduri.** O resursă este tot ce poate spune lucrurile despre ea. Este ușor să te gândești la o resursă ca pe un lucru în raport cu o valoare. Într-o reprezentare vizuală, resursele sunt reprezentate de ovale.

**Noduri literale.** Termenul literal este un termen folosit pentru valoare. În exemplul de mai sus, resursa este <http://www.cambridgesemantics.com/people/about/rob> (din nou un URI) și proprietatea nume este șirul "Rob Gonzalez". Într-o reprezentare vizuală, literalii sunt reprezentați prin dreptunghiuri.

**Noduri goale.** Un nod gol este o resursă fără un URI.

Într-un graf pentru date, nu există niciun concept de rădăcini (sau o ierarhie). Un graf constă din resurse legate de alte resurse, fără ca o anumită resursă să aibă o importanță intrinsecă deosebită față de alta.

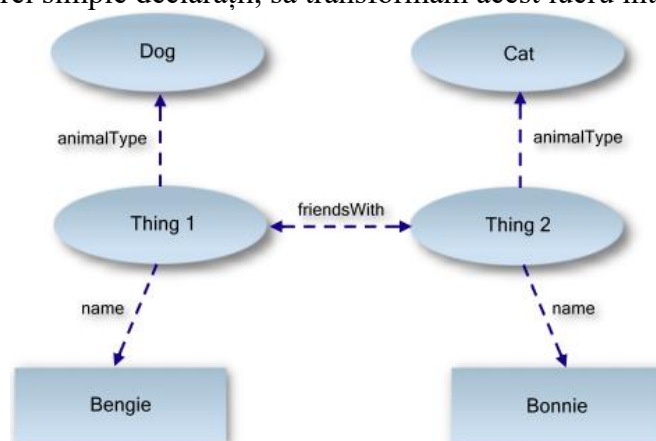
Uitați-vă la următoarele afirmații care descriu relația dintre un câine (numit Bengie) și o pisică (numită Bonnie):

*Bengie is a dog.*

*Bonnie is a cat.*

*Bengie and Bonnie are friends.*

Folosind aceste trei simple declarații, să transformăm acest lucru într-un grafic de date:



Relațiile implicate în acest grafic sunt destul de intuitive, dar trebuie să le analizăm cu atenție. Putem vedea că cele două lucruri - identificate de "Thing 1" și "Thing 2" - au proprietățile name, animalType și friendsWith. Din aceasta vedem că numele "Thing 1" este

Bengie, iar numele "Thing 2" este Bonnie. "Thing 1" este un câine, iar "Thing 2" este o pisică. Și, în sfârșit, ambii sunt prieteni unul cu celălalt.

Săgețile din diagrama de mai sus sunt numite proprietăți, uneori în terminologia RDF - predicate.

RDF identifică lucrurile utilizând identificatori web (URI) și descrie resursele cu proprietățile și valorile proprietăților.

Explicarea valorii resurselor, a proprietății și a proprietății:

-O resursă este orice poate avea un URI, cum ar fi "https://www.w3schools.com/rdf"

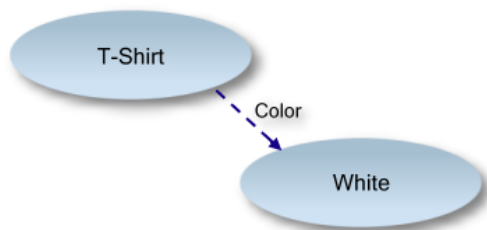
-O proprietate este o resursă care are un nume, cum ar fi "autor" sau "pagină de pornire"

-O valoare a proprietății este valoarea unei proprietăți, cum ar fi "Jan Egil Refsnes" sau "https://www.w3schools.com" (rețineți că o valoare a proprietății poate fi o altă resursă)

RDF / XML de mai sus (între etichetele <rdf: Description>) se numește o instrucțiune RDF sau, uneori, se numește un triplă RDF. Dintre cele două, triplă este termenul cel mai folosit, deoarece descrie ruperea declarației în cele trei părți componente ale ei: subiectul, predicatul și obiectul declarației.

Combinarea unei resurse, a unei proprietăți și a unei valori de proprietate formează o declarație (cunoscută ca subiect, predicat și obiect al unei declarații).

Este mai simplu să înțelegem mai întâi acești termeni sub forma unui grafic simplu. Uită-te la următorul grafic de date care descrie culoarea unui tricou:



- Subiectul (Resursa) este the *T-shirt*
- Predicatul (proprietatea) este *color*
- Obiectul (valoarea proprietății) is *white*

Exemplu de implementare:

```
01. <?xml version="1.0" encoding="UTF-8"?>
02.
03. <rdf:RDF
04.   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
05.   xmlns:feature="http://www.linkeddatatools.com/clothing-features#"
06.
07.   <rdf:Description rdf:about="http://www.linkeddatatools.com/clothes#t-shirt">
08.
09.     <feature:color rdf:resource="http://www.linkeddatatools.com/colors#white"/>
10.
11.   </rdf:Description>
12.
13. </rdf:RDF>
```

## Construirea unui document folosind RDF

Se adaugă nodul rădăcină

```
1. <rdf:RDF
2.   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.
4.   <!-- Body Code Omitted -->
5.
6. </rdf:RDF>
```

La linia 2, este trecut spațiul de nume standard W3.org <http://www.w3.org/1999/02/22-rdf-syntax-ns#>. Acest spațiu de nume spune fiecărui cititor de mașini că documentul care însoțește este un document RDF și că tag-ul rdf: RDF se află în acest spațiu de nume.

Acest spațiu de nume și nodul RDF formează rădăcina tuturor documentelor RDF.

### *Adăugarea unei declarație*

Un document RDF poate conține mai multe instrucțiuni. Pentru simplificare, vom adăuga doar una. Începem prin adăugarea unei etichete rdf: Description, care în RDF / XML poate conține una sau mai multe instrucțiuni despre același subiect:

```
01. <rdf:RDF
02.   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
03.
04.   <rdf:Description rdf:about="http://www.linkeddatatools.com/clothes#t-shirt">
05.
06.     <!-- Statement Code Omitted -->
07.
08.   </rdf:Description>
09.
10. </rdf:RDF>
```

### *Adăugarea unei proprietăți ca atribut:*

```
01. <rdf:RDF
02.   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
03.   xmlns:feature="http://www.linkeddatatools.com/clothing-features#"
04.
05.   <rdf:Description rdf:about="http://www.linkeddatatools.com/clothes#t-shirt">
06.
07.     <feature:size>12</feature:size>
08.
09.   </rdf:Description>
10.
11. </rdf:RDF>
```

La linia 07. Acest lucru spune pur și simplu "Subiectul are o proprietate cu caracteristică de nume: dimensiune care are valoarea literală 12". În terminologia RDF, aceasta este o declarație.

Să adăugăm încă un predicat: culoarea tricoului.

```
01. <rdf:RDF
02.   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
03.   xmlns:feature="http://www.linkeddatatools.com/clothing-features#"
04.
05.   <rdf:Description rdf:about="http://www.linkeddatatools.com/clothes#t-shirt">
06.
07.     <feature:size>12</feature:size>
08.     <feature:color rdf:resource="http://www.linkeddatatools.com/colors#white"/>
09.
10.   </rdf:Description>
11.
12. </rdf:RDF>
```

În timp ce ultima avea valoarea literală 12, aceasta se referă la subiectul (ID) al unei alte afirmații. Predicatul a fost definit ca resursă.

Acum am analizat un exemplu simplu de document RDF, să formalizăm ceea ce am învățat și să rupem declarația în părțile sale componente:

```
1. <rdf:Description rdf:about="subject">
2.   <predicate rdf:resource="object" />
3.   <predicate>literal value</predicate>
4. </rdf:Description>
```

Aici vedeți subiectul instrucțiunii (despre ce este vorba) și cele două forme de predicate (valori și resurse literale care se referă la alte instrucțiuni RDF).

```

01. <?xml version="1.0" encoding="UTF-8"?>
02.
03. <rdf:RDF
04.   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
05.   xmlns:dc="http://purl.org/dc/elements/1.1/"
06.   xmlns:region="http://www.country-regions.fake/">
07.
08.   <rdf:Description rdf:about="http://en.wikipedia.org/wiki/Oxford">
09.     <dc:title>Oxford</dc:title>
10.     <dc:coverage>Oxfordshire</dc:coverage>
11.     <dc:publisher>Wikipedia</dc:publisher>
12.     <region:population>10000</region:population>
13.     <region:principaltown rdf:resource="http://www.country-regions.fake/oxford"/>
14.   </rdf:Description>
15.
16. </rdf:RDF>

```

Pentru a testa înțelegerea dvs. și a indica toate zonele în care trebuie să recapitulați, vedeți dacă puteți identifica în documentul RDF anterior următoarele:

- ✓ Subiectul declarației
- ✓ Predicatul - inclusiv dacă sunt resurse sau literale
- ✓ Obiectele menționate de predicatele resurselor

Elementul `<rdf: Bag>` este folosit pentru a descrie o listă de valori care nu trebuie să fie într-o anumită ordine. Elementul `<rdf: Bag>` poate conține valori duplicate.

```

<?xml version="1.0"?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cd="http://www.recshop.fake/cd#">

  <rdf:Description
    rdf:about="http://www.recshop.fake/cd/Beatles">
    <cd:artist>
      <rdf:Bag>
        <rdf:li>John</rdf:li>
        <rdf:li>Paul</rdf:li>
        <rdf:li>George</rdf:li>
        <rdf:li>Ringo</rdf:li>
      </rdf:Bag>
    </cd:artist>
  </rdf:Description>

</rdf:RDF>

```

#### Bibliografie:

<https://www.cambridgesemantics.com/blog/semantic-university/lear-rdf/>

<https://www.linkeddatatools.com/introducing-rdf-part-2>

[https://www.w3schools.com/XML/xml\\_rdf.asp](https://www.w3schools.com/XML/xml_rdf.asp)