

RDF - cadru de descriere a resurselor Internet bazat pe XML

[Sabin Corneliu Buraga](mailto:busaco@infoiasi.ro) (busaco@infoiasi.ro)

Articol aparut in PC Report, PC Report, vol.8, 10 (85), octombrie 1999

Prezentare generala

Spatiul cibernetic a fost initial conceput pentru a usura regasirea de catre calculator a oricarei date indiferent de localizarea ei, fara a se pune problema intelegerii semnificatiei acesteia de catre masina. Din cauza volumului tot mai mare de informatii prezent pe Web este dificil de a automatiza regasirea lor inteligenta, cu atit mai putin de catre operatorul uman.

Una din solutii, pe care o vom prezenta in continuare ca studiu de caz, este de a utiliza **metadata** pentru descrierea datelor continute de Internet. Metadata reprezinta un set de date referitoare la date (resurse Web).

RDF (Resource Description Framework) este un cadru menit sa proceseze metadatale, oferind interoperabilitatea intre diverse aplicatii care fac schimb inteligent de informatii, in sensul intelegerii de catre masina a semanticii acestora. RDF isi gaseste loc in utilizari ca:

- *inspectarea resurselor*, oferind noi capabilitati motoarelor de cautare;
- *catalogarea datelor* pentru descrierea si/sau evaluarea continutului si relatiilor intre diverse informatii stocate intr-o biblioteca electronica, site Web etc.
- *agenti inteligenti*, facilitind schimbul si partajarea cunostintelor;
- *descrierea drepturilor de proprietate intelectuala* a paginilor Web;
- *securitate personala sau generala* a datelor (oferind suport pentru *semnaturi digitale* utile in comertul electronic, tranzactii economice si juridice etc.)

RDF foloseste limbajul XML pentru reprezentarea sintactica a metadatelor. Unul din scopurile cadrului este de a face posibila specificarea semantica a datelor, bazata pe XML, printr-o metoda standardizata, independenta de masina, extensibila. RDF si XML sint complementare in acest sens.

In primul rind, RDF trebuie sa defineasca un mecanism de descriere a resurselor independent de domeniul de folosire a datelor, fara a specifica a priori vreo semantica. Definirea acestui mecanism trebuie sa ramina neutra, dar generala, dupa cum vom vedea mai jos.

Pentru a facilita definirea datelor RDF, va fi necesar un sistem de clase similar celui din programarea orientata-obiect. O colectie de clase (dezvoltata pentru un anumit scop specific) se numeste **schema**. Clasele sint organizate ierarhic oferind extensibilitatea prin rafinarea subclaselor. Astfel, pentru crearea unei noi scheme putem sa ne bazam pe o schema de baza (un fel de clasa abstracta in termenii programarii orientate obiect). Se asigura in acest mod si reutilizarea definitiilor de metadata. Datorita caracterului extensibil, agentii care proceseaza metadatale vor fi capabili de versatilitate in manipularea schemelor. Mostenirea multipla permite exploatarea in mai multe metode a aceleasi informatii. E posibil sa cream instante de date RDF bazate pe multiple scheme din diverse surse.

Influienta RDF se poate intrevedea in *structurarea inteligenta a documentelor* (realizata in SGML ori XML), in *reprezentarea cunostintelor* (KR - Knowledge Representation), in *standardizarea Web-ului*. Alte arii de interes ar fi limbajele orientate-obiect si de modelare a cunostintelor sau bazele de date distribuite.

Modelul de baza al RDF

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

- **resurse**

Datele descrise de expresiile RDF sint denumite **resurse**. O resursa poate fi o pagina Web completa (de exemplu un document HTML desemnat printr-un URL:

`http://www.infoiasi.ro/circles/index.html`), o parte a unei pagini Web (un element specific HTML sau XML prezent in sursa documentului, de pilda o imagine) sau un obiect care nu-i direct accesibil via Web (e.g. o carte tiparita). Resursele sint specificate de URI-uri plus un identificator de legatura, optional.

- **proprietati**

O **proprietate** reprezinta un aspect specific, o caracteristica, un atribut sau o relatie pentru a descrie o resursa. Fiecare proprietate posedă o semantica, un set de valori permise, o multime de tipuri de resurse pe care le descrie si un set de relatii (interdependente) cu alte proprietati.

- **declaratii**

O anumita resursa impreuna cu o proprietate a sa avind asignata o valoare formeaza o **declaratie**. Putem privi declaratia ca un 3-uplu: {**subiect, predicat, obiect**}. Obiectul declaratiei (valoarea proprietatii) poate desemna o alta resursa (specificata de un URI) sau un literal (tip primitiv de data sau sir de caractere, conform specificatiilor XML). In modelul RDF, un *literal* poate contine marcaje XML care inasa nu vor fi evaluate (analizate) de procesorul RDF.

Modul de reprezentare

Declaratiile se pot reprezenta astfel:

- **graf orientat**: nodurile sint fie subiecte fie obiecte, iar arcele semnifica un predicat;
- **marcaje**: <subject> HAS <predicate> <object>
- **RDF/XML** (vezi mai jos)

O proprietate poate avea drept valoare o *entitate structurata*:

```
Individul Sabin Corneliu Buraga, avind adresa
busaco@infoiasi.ro, este creatorul resursei
http://www.infoiasi.ro/~busaco/odix.
```

In acest caz, obiectul nu va fi un sir de caractere, ci o colectie formata din trei literali: "individ", "Sabin Corneliu Buraga", "busaco@infoiasi.ro". Vom vedea ulterior cum specificam acest lucru in RDF.

Sintaxa de baza RDF

Vom da descrierile sintactice in notatia EBNF. Toate facilitatile sintactice din XML (regulile spatiilor albe, diferentele dintre apostrof si ghilimele, senzitivitatea caracterelor, moduri de adnotare) sint suportate.

Mai multe definitii pentru o anumita resursa pot fi grupate in cadrul elementului *Description* (suport pentru incapsulare).

Regulile sintactice sint urmatoarele:

```
[1] RDF      ::= ['<rdf:RDF>'] descript* ['</rdf:RDF>']
[2] descript ::= '<rdf:Description' idAboutAttr? '>' propElt*
                '</rdf:Description>'
[3] idAboutAttr ::= idAttr | aboutAttr
[4] idAttr      ::= 'ID="' Idsymbol '"]
```

```

[5]  aboutAttr ::= 'about="' URI-ref '"'
[6]  propElt   ::= '<' propName '>' value '</' propName '>' |
                '<' propName resAttr '/>'
[7]  propName  ::= QName
[8]  value     ::= descript | string
[9]  resAttr   ::= 'resource="' URI-ref '"'
[10] QName    ::= [ NSprefix ':' ] name
[11] URI-ref  ::= string
[12] IDsymbol ::= (orice simbol legal XML)
[13] name     ::= (orice simbol legal XML)
[14] NSprefix ::= (orice prefix al spatiului de nume din XML)
[15] string   ::= (orice text XML)

```

rdf reprezentat cu italice este utilizat pentru a reprezenta un prefix al spatiului de nume, util pentru imbricarea exacta a tag-urilor de inceput si de sfirsit.

Pentru exemplul de mai sus, avem:

```

<rdf:RDF>
  <rdf:Description about="http://www.infoiasi.ro/~busaco/odix">
    <s:Creator>Sabin Corneliu Buraga</s:Creator>
  </rdf:Description>
</rdf:RDF>

```

Aici prefixul 's' se refera la un prefix specific ales de autorul expresiei RDF si definit intr-o declaratie a spatiului de nume XML, conform unei scheme: `xmlns:s="http://description.org/schema/"`.

Scheme si spatii de nume

Atunci cind scriem o afirmatie in limbaj natural, utilizam cuvinte care au un anumit inteles pentru noi si pentru cel careia ii este adresata. Intelegerea semanticii propozitiei este cruciala in stabilirea cu exactitate a procesarii ce trebuie urmata. Este extrem de important ca atat scriitorul cit si cititorul enuntului sa recepteze *acelasi* inteles al termenilor utilizati, altfel s-ar crea confuzii. In mediul global reprezentat de WWW nu-i suficient a ne ghida dupa intelegerea culturala comuna a conceptelor.

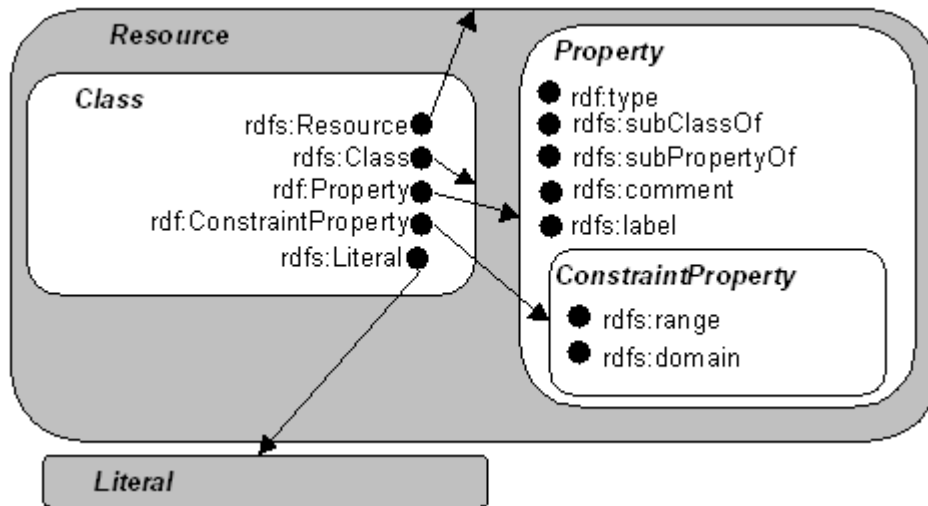
Intelesul unui termen (lingvistic sau nu) in RDF este exprimat printr-o referinta la o **schema**. Putem privi schema ca un fel de dictionar, definind termenii pe care ii vom utiliza in declaratiile RDF si asociindu-le o semantica precisa. Se pot folosi o varietate de scheme, specificate sau nu ca documente separate.

O schema contine definitii si restrictii de utilizare a proprietatilor. Pentru evitarea confuziilor dintre definitiile independente a unui acelasi lucru, RDF se bazeaza pe facilitatea spatiilor de nume din XML. Spatiile de nume ofera o modalitate simpla de a folosi la un moment dat o unica definitie a unui termen. Fiecare predicat al unei declaratii RDF trebuie identificat de o unica schema. Un element `Description` poate insa contine declaratii avind predicate din mai multe scheme.

Schemele in detaliu

Declararea proprietatilor (atributelor) unor resurse si semantica asociata lor se realizeaza prin intermediul **schemelor**. RDF poate fi vazut astfel si ca *limbaj de specificare a schemelor*, fiind mai facil de implementat decit limbajele mai complexe *CycL* (The CYC Representation Language) sau *KIF* (Knowledge Interchange Format). Schemele RDF au la baza idei preluate din reprezentarea cunostintelor (retele semantice, logica predicatelor) ori din limbajele de specificare a bazelor de date.

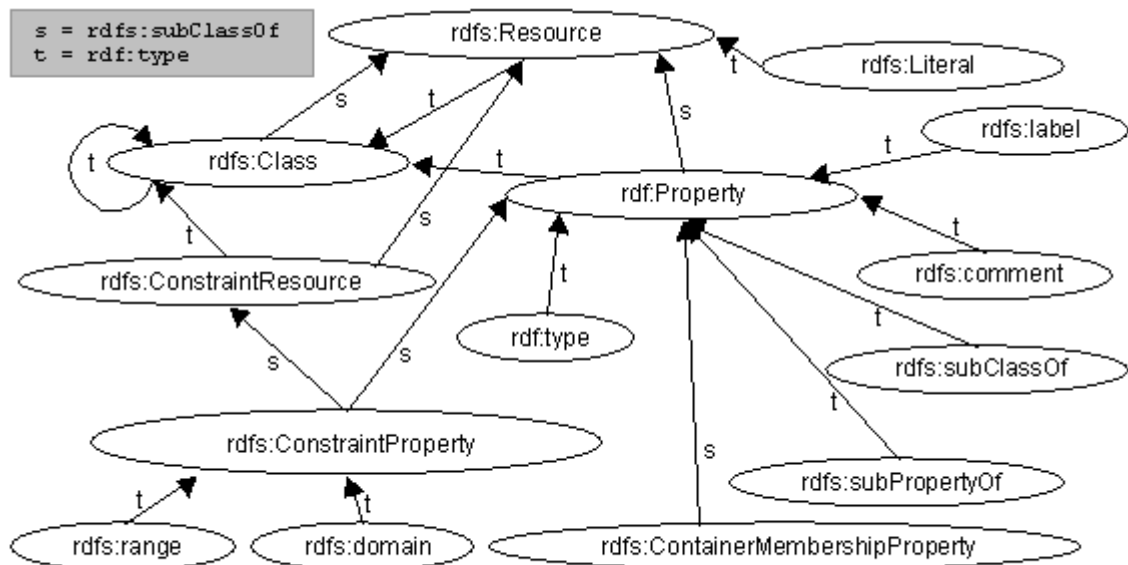
O schema consta dintr-un set de clase si proprietati. Sint definite o serie de clase si de proprietati fundamentale. De asemeni, pentru schemele RDF se defineste un spatiu de nume XML denumit `rdfs`.



Seturile de clase si proprietati

Clase fundamentale

- `rdfs:Resource` defineste clasa resurselor, corespunzind conceptului de *obiect* din limbajele de programare orientate-obiect.
- `rdf:Property` reprezinta clasa proprietatilor resurselor.
- `rdfs:Class` corespunde conceptului general de tip sau categorie. Cind o schema defineste o noua clasa, resursa reprezentind acea clasa trebuie sa aiba o proprietate `rdfs:type` a carei valoare e resursa `rdfs:Class`. Clasele RDF pot specifica, de exemplu, pagini Web, tipuri de documente, baze de date, persoane etc.



Ierarhiile de clase RDF

Proprietati fundamentale

Fiecare model RDF care utilizeaza un mecanism de scheme include, in mod implicit, proprietatile de mai jos, instante ale clasei `rdf:Property`, oferind o modalitate de a exprima relatiile dintre clase si instantele lor sau supraclase.

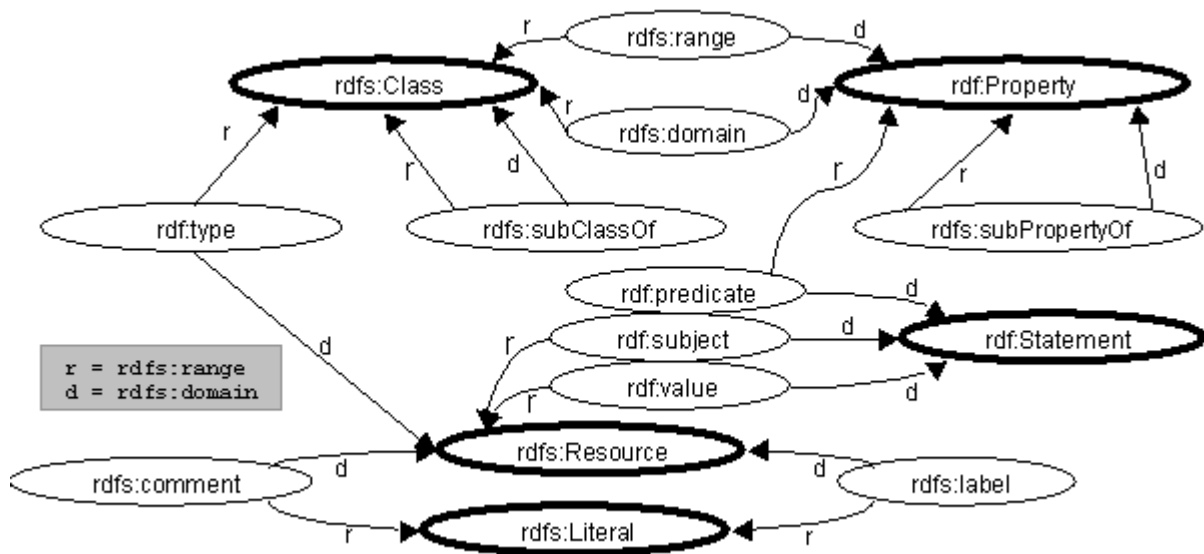
- `rdf:type` indica faptul ca o resursa este membra a unei clase. Atunci cind o resursa are o proprietate `rdf:type` a carei valoare reprezinta o anumita clasa, vom spune ca resursa este o

instanta a acelei clase. Valoarea lui `rdf:type` pentru o resursa este o alta resursa, instanta a lui `rdfs:Class`. Clasele individuale intotdeauna vor avea `rdf:type` asignata cu valoarea `rdfs:Class` (ori o sub-clasa a lui `rdfs:Class`). O resursa poate fi instanta a mai multor clase, desigur.

- `rdfs:subClassOf` indica relatia de mostenire a claselor. Este o relatie tranzitiva. Doar instancele lui `rdfs:Class` pot avea proprietatea `rdfs:subClassOf` si valoarea ei este intotdeauna `rdf:type rdfs:Class`. O clasa poate fi sub-clasa a mai multor clase. O clasa niciodata nu poate fi declarata ca sub-clasa a ei insasi sau drept sub-clasa a sub-claselor sale.
- `rdfs:subPropertyOf` - O proprietate poate avea zero, una sau mai multe proprietati, specializari ale ei. Daca o anumita proprietate P_1 este o subproprietate a unei proprietati mai generale P_2 si daca o resursa A are proprietatea P_2 avind asignata valoarea B , atunci aceasta implica: resursa A are de asemeni proprietatea P_1 cu valoarea B .

Restrictii

O schema poate declara anumite restrictii asociate claselor si proprietatilor. In jargonul RDF, vor fi folosite conceptele de **domeniu (domain)** si **interval (range)**.



Restrictiile in RDF

Un model care violeaza o restrictie este un *model inconsistent*. Diverse aplicatii pot avea comportamente eronate in cadrul unui model inconsistent.

Exemple de restrictii:

- valoarea unei proprietati trebuie sa fie o resursa ori o clasa definita de proiectantul unei scheme. Aceasta restrictie este exprimata de proprietatea `range` (de exemplu, restrictia aplicata proprietatii `"autor"` poate avea restrictia ca valoarea ei sa fie o resursa, instanta a clasei `"Persoana"`).
- o proprietate poate fi utilizata doar de resursele unei anumite clase (de exemplu, proprietatea `"autor"` poate fi folosita numai daca valoarea ei este o resursa a carei instanta este clasa `"Tratat"`). Acest lucru se exprima prin proprietatea `domain`.

Restrictii fundamentale

- `rdfs:ConstraintResource` defineste o sub-clasa a lui `rdfs:Resource` ale carei instance sint constructii de scheme implicate in exprimarea restrictiilor (mecanism de verificare de catre procesoarele RDF a consistentei unui model).
- `rdfs:Range` este folosita pentru a restrictiona valorile unei proprietati. Valoarea lui `range` este intotdeauna o clasa. Valoarea unei proprietati a carei interval este A este constrinsa sa fie

instanta a clasei A. Putem avea cel mult o proprietate `range`.

- `rdfs:domain` e utilizata sa specifice o clasa ce poate fi asignata ca valoare a unei proprietati. O proprietate poate avea valori din zero, una sau mai multe clase. Daca nu exista vreun domeniu, poate fi folosita oricare resursa.

Colectii de resurse

Este necesar deseori sa utilizam colectii de resurse, pentru aceasta in RDF definindu-se trei tipuri de obiecte:

- **bag** - lista neordonata de resurse sau literali, valorile duplicate fiind permise;
- **secventa** - lista ordonata de resurse sau literali. Ca mai sus, valorile pot fi duplicate;
- **alternativa** - o lista de resurse/literali care reprezinta alternative pentru o singura valoare a unei proprietati.

Tipul multime nu este inca definit de specificatiile RDF.

Pentru a crea o colectie de resurse, RDF utilizeaza o resursa suplimentara ce reprezinta o colectie specifica (o *instanta* a colectiei), aceasta resursa declarandu-se ca instanta a unui tip din cele anterioare, prin proprietatea `type`. Relatia dintre colectia de resurse si resursele ce apartin acesteia este data de un set de proprietati denumite `"_1"`, `"_2"`, `"_3"` etc. Colectiile de resurse pot avea si alte proprietati, desigur.

Sintaxa formala este urmatoarea:

```
[16] contain ::= seq | bag | alt
[17] seq     ::= '<rdf:Seq' idAttr? '>' member* '</rdf:Seq>'
[18] bag     ::= '<rdf:Bag' idAttr? '>' member* '</rdf:Bag>'
[19] alt     ::= '<rdf:Alt' idAttr? '>' member* '</rdf:Alt>'
[20] member  ::= referItem | inlineItem
[21] referItem ::= '<rdf:li resourceAttr '>'
[22] inlineItem ::= '<rdf:li>' value '</rdf:li>'
```

Colectiile pot apare oriunde este permis un element `Description`, deci regulile sintactice se modifica astfel:

```
[1a] RDF     ::= '<rdf:RDF>' obj* '</rdf:RDF>'
[8a] value  ::= obj | string
[23] obj    ::= descript | contain
```

Exemple:

a. Modelul pentru enuntul:

Studentii cursului de Limbaj Natural sint Cristina, Mihaela si Cosmin.

este scris in RDF in modul urmator:

```
<rdf:RDF>
  <rdf:Description about="http://www.infoiasi.ro/courses/nlp">
    <s:Students>
      <rdf:Bag>
        <rdf:li resource="http://www3.infoiasi.ro/~Cristina" />
        <rdf:li resource="http://www3.infoiasi.ro/~Mihaela" />
        <rdf:li resource="http://www3.infoiasi.ro/~Cosmin" />
      </rdf:Bag>
    </s:Students>
  </rdf:Description>
</rdf:RDF>
```

b. Modelul pentru propozitia:

Codul sursa pentru GAEN poate fi gasit la [ftp.infoiasi.ro](ftp://www.infoiasi.ro), [ftp.uaic.ro](ftp://ftp.uaic.ro) sau hal.cs.tuiasi.ro.

in RDF/XML se scrie astfel:

```
<rdf:RDF>
  <rdf:Description about="http://www.infoiasi.ro/~busaco/gaen">
    <s:DistributionSite>
      <rdf:Alt>
        <rdf:li resource="ftp://ftp.infoiasi.ro" />
        <rdf:li resource="ftp://ftp.uaic.ro/pub/misc/gaen" />
        <rdf:li resource="ftp://hal.cs.tuiasi.ro/pub/sources/Unix" />
      </rdf:Alt>
    </s:DistributionSite>
  </rdf:Description>
</rdf:RDF>
```

Referenti distributivi

Obiectul descris de o declaratie RDF (indicat de atributul `about`) este numit **referent**.

In urmatorul exemplu:

```
<rdf:Bag ID="pages">
  <rdf:li resource="http://www.infoiasi.ro/circles/1/index.html">
  <rdf:li resource="http://www.infoiasi.ro/circles/2/index.html">
  <rdf:li resource="http://www.infoiasi.ro/circles/3/index.html">
</rdf:Bag>

<rdf:Description about="#pages">
  <s:Creator>Sabin Corneliu Buraga</s:Creator>
</rdf:Description>
```

exprimam faptul ca 'Sabin Corneliu Buraga' este creatorul colectiei de pagini 'pages'. Referentul elementului `Description` este o colectie (de tip *bag*), nu membrii ei. Pentru a specifica faptul ca 'Sabin Corneliu Buraga' este creatorul fiecarei pagini, se foloseste un alt tip de referire, specificat de atributul `aboutEach`, acest tip de referent fiind numit **referent distributiv**.

```
<rdf:Description aboutEach="#pages">
  <s:Creator>Sabin Corneliu Buraga</s:Creator>
</rdf:Description>
```

Colectii referite de un URI

Una din utilizarile metadatelor este de a face declaratii despre "toate paginile Web disponibile pe un anumit server" sau "toate paginile Web descriind un anumit aspect, aflate la o adresa specifica". In multe cazuri este dificil ori neimportant sa incercam sa enumeram fiecare resursa in mod explicit si s-o identificam ca membru al unei colectii. Folosind o a doua forma de referenti distributivi putem crea o colectie de tip *Bag* pentru a defini toate resursele ce ne intereseaza in aplicatiile noastre:

```
[3a] idAboutAttr ::= idAttr | aboutAttr | aboutEachAttr
[24] aboutEachAttr ::= 'aboutEach="' URI-ref '"' |
                       'aboutEachPrefix="' string '"'
```

Atributul `aboutEachPrefix` declara o colectie ale carei membri sint toate resursele corespunzatoare identificatorilor ce incep cu sirul de caractere al valorii atributului.

Un exemplu:

Pentru toate paginile Web prezente pe serverul [Facultatii de Informatica](#) putem seta o proprietate de copyright scriind:

```
<rdf:Description aboutEachPrefix="http://www.infoiasi.ro">
  <s:Copyright>©1998-1999,
```

Sabin Corneliu Buraga</s:Copyright>
</rdf:Description>

Nu ne intereseaza cite documente exista pe disc si interdependenta lor.

Declaratii despre declaratii

In cadrul RDF se pot crea declaratii privitoare la alte declaratii. Vom numi acest tip de declaratii: **declaratii de nivel inalt**.

Modelarea declaratiilor

Considerind afirmatia:

Sabin Corneliu Buraga este creatorul resursei <http://www.infoiasi.ro>.

o putem vedea ca un *fapt* (asemanator clauzelor Prolog).

Daca insa avem enuntul:

Dumitru Todoroi spune ca Sabin Corneliu Buraga este creatorul resursei <http://www.infoiasi.ro>.

exprimam un fapt despre o afirmatie facuta de altcineva.

Putem modela, dupa cum vom vedea mai jos, declaratia originala ca o resursa avind patru proprietati. Acest proces este denumit, in termenii reprezentarii cunostintelor, **reificare** (reification).

Se definesc urmatoarele proprietati:

- *subiect*
identifica resursa descrisa de declaratia modelata; valoarea subiectului este resursa specificata in declaratia initiala (in exemplul nostru <http://www.infoiasi.ro>);
- *predicat*
identifica proprietatea originala (reprezentind in cazul nostru creatorul);
- *obiect*
specifica valoarea proprietatii declaratiei modelate (aici Sabin Corneliu Buraga);
- *tip*
descrie tipul noii resurse. Toate declaratiile reificate sint instante ale lui `RDF:Statement`. Astfel, ele au o proprietate `type` al carei obiect este `RDF:Statement`. Proprietatea `type` se poate folosi pentru declararea oricarui tip de resursa, cistigind in acest mod flexibilitate.

Aplicatii

Exista deja definite, avind ca suport cadrul de descriere a resurselor Internet prezentat aici, diverse metadata utilizate in aplicatiile Web. In continuare ne vom ocupa numai de o modalitate de a descoperi resursele electronice intr-o maniera similara celei de consultare a unui catalog de biblioteca: **Dublin Core Metadata**, folosind vocabulare definite de **Dublin Core Initiative**.

Se specifica spatiile de nume: `dc` (Dublin Core) disponibil la adresa http://purl.org/metadata/dublin_core si `dcq` (Dublin Core Qualifiers) la adresa http://purl.org/metadata/dublin_core_qualifiers. In fapt, Dublin Core Metadata defineste o schema avind 15 proprietati de baza utile pentru descrierea oricarei resurse Web, in special pentru activitati de cautare.

Prezentam un exemplu de descriere inteligenta a unei publicatii electronice disponibile pe Web, [Circles](#).

<rdf:RDF

```

xmlns:rdf="http://www.w3.org/TR/1999/PR-rdf-syntax#"
xmlns:dc="http://purl.org/metadata/dublin_core#"
xmlns:dcq="http://purl.org/metadata/dublin_core_qualifiers#"
<rdf:Description about="http://www.infoiasi.ro/circles">
  <dc:Title>Circles - an electronic magazine</dc:Title>
  <dc:Description>Circles este o revista electronica independenta care
    apare regulat incepind cu luna februarie 1997, imbinind stiri din
    lumea informaticii cu aspecte culturale si nu numai atit.
  </dc:Description>
  <dc:Contributor rdf:parseType="Resource">
    <dcq:AgentType
      rdf:resource=
        "http://purl.org/metadata/dublin_core_qualifiers#Editor" />
    <rdf:value>Sabin Corneliu Buraga</rdf:value>
  </dc:Contributor>
  <dc:Publisher>Faculty of Computer Science</dc:Publisher>
  <dc>Date>1997-17-02</dc>Date>
  <dc:Type>electronic magazine</dc:Type>
  <dc:Subject>      <!-- subiectul abordat (colectie) -->
    <rdf:Bag>
      <rdf:li>computer science</rdf:li>
      <rdf:li>literature and art</rdf:li>
      <rdf:li>other different topics</rdf:li>
    </rdf:Bag>
  </dc:Subject>
  <dc:Format>      <!-- formatul revistei: cimpuri MIME -->
    <rdf:Bag>
      <rdf:li>text/html</rdf:li>
      <rdf:li>image/jpeg</rdf:li>
      <rdf:li>image/gif</rdf:li>
    </rdf:Bag>
  </dc:Bag>
  <dc:Relation rdf:parseType="Resource">
    <dcq:RelationType      <!-- relatia cu serverul infoiasi.ro -->
      rdf:resource=
        "http://purl.org/metadata/dublin_core_qualifiers#IsPartOf" />
    <rdf:value resource="http://www.infoiasi.ro" />
    </dcf:RelationType>
  </dc:Relation>
</rdf:Description>
</rdf:RDF>

```

Resurse

- Resource Description Framwork (RDF) Model and Syntax: <http://www.w3.org/TR/1999/PR-rdf-syntax/>
- Resource Description Framwork (RDF) Schema Specification: <http://www.w3.org/TR/WD-rdf-schema/>
- Knowledge Interchange Format (KIF): <http://logic.stanford.edu/kif>
- The Dublin Core Initiative: http://purl.oclc.org/metadata/dublin_core
- Tim Berners-Lee & Dan Connolly - Web Architecture: Extensible Languages: <http://www.w3.org/TR/NOTE-webarch-extlang>
- Grady Booch - Object Oriented Analysis and Design, The Benjamin/Cummings Publishing Company, 1994
- G.M.Nijssen & Terry Halpin - Conceptual Schema and Relational Database Design, Prentice Hall, 1989