

**Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie**



**Wydział Elektrotechniki, Automatyki, Informatyki i
Elektroniki**

Podstawy inżynierii wiedzy

Semantic Web

Temat: Introduction to the Semantic Web + XML

Nawojka Schmidt, Mateusz Ziarko

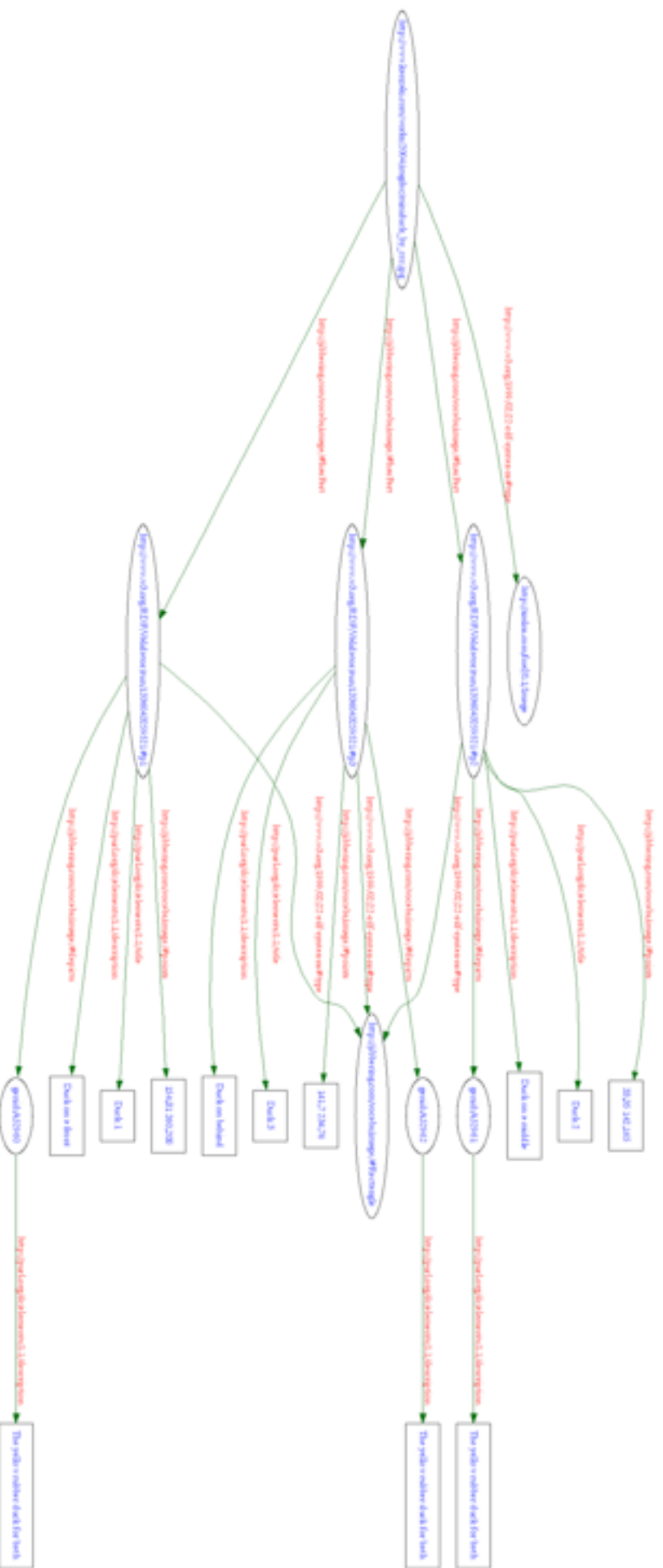
Wtorek 17:30

1. Extracting RDF metadata

Number	Subject	Predicate	Object
1	http://www.kanzaki.com/works/2004/imgdsc/miniduck_by_ccc.jpg	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://xmlns.com/foaf/0.1/Image
2	http://www.w3.org/RDF/Validator/run/1336043259521#p1	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://jibbering.com/vocabs/image/#Rectangle
3	http://www.kanzaki.com/works/2004/imgdsc/miniduck_by_ccc.jpg	http://jibbering.com/vocabs/image/#hasPart	http://www.w3.org/RDF/Validator/run/1336043259521#p1
4	http://www.w3.org/RDF/Validator/run/1336043259521#p1	http://jibbering.com/vocabs/image/#points	"154,81 260,208"
5	http://www.w3.org/RDF/Validator/run/1336043259521#p1	http://purl.org/dc/elements/1.1/title	"Duck 1"
6	http://www.w3.org/RDF/Validator/run/1336043259521#p1	http://purl.org/dc/elements/1.1/description	"Duck on a front"
7	http://www.w3.org/RDF/Validator/run/1336043259521#p1	http://jibbering.com/vocabs/image/#depicts	genid:A52960
8	genid:A52960	http://purl.org/dc/elements/1.1/description	"The yellow rubber duck for bath"
9	http://www.w3.org/RDF/Validator/run/1336043259521#p2	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://jibbering.com/vocabs/image/#Rectangle
10	http://www.kanzaki.com/works/2004/imgdsc/miniduck_by_ccc.jpg	http://jibbering.com/vocabs/image/#hasPart	http://www.w3.org/RDF/Validator/run/1336043259521#p2

Number	Subject	Predicate	Object
11	http://www.w3.org/RDF/Validator/run/1336043259521#p2	http://jibbering.com/vocabs/image/#points	"33,35 142,165"
12	http://www.w3.org/RDF/Validator/run/1336043259521#p2	http://purl.org/dc/elements/1.1/title	"Duck 2"
13	http://www.w3.org/RDF/Validator/run/1336043259521#p2	http://purl.org/dc/elements/1.1/description	"Duck on a middle"
14	http://www.w3.org/RDF/Validator/run/1336043259521#p2	http://jibbering.com/vocabs/image/#depicts	genid:A52961
15	genid:A52961	http://purl.org/dc/elements/1.1/description	"The yellow rubber duck for bath"
16	http://www.w3.org/RDF/Validator/run/1336043259521#p3	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://jibbering.com/vocabs/image/#Rectangle
17	http://www.kanzaki.com/works/2004/imgdsc/miniduck_by_ccc.jpg	http://jibbering.com/vocabs/image/#hasPart	http://www.w3.org/RDF/Validator/run/1336043259521#p3
18	http://www.w3.org/RDF/Validator/run/1336043259521#p3	http://jibbering.com/vocabs/image/#points	"141,7 234,76"
19	http://www.w3.org/RDF/Validator/run/1336043259521#p3	http://purl.org/dc/elements/1.1/title	"Duck 3"
20	http://www.w3.org/RDF/Validator/run/1336043259521#p3	http://purl.org/dc/elements/1.1/description	"Duck on behind"
21	http://www.w3.org/RDF/Validator/run/1336043259521#p3	http://jibbering.com/vocabs/image/#depicts	genid:A52962

Number	Subject	Predicate	Object
22	genid:A52962	http://purl.org/dc/elements/1.1/description	"The yellow rubber duck for bath"



2. Common Vocabularies

Definicje

SKOS - jest to rodzina słowników służąca do dokumentowania taksonomi, tezaurusów czy schematów klasyfikacyjnych, ich łączenia z innymi danymi w celu publikacji w sieci.

Dublin Core - standard metadanych, służący do opisu zasobów internetowych z wykorzystaniem 15 zdefiniowanych elementów.

FOAF - jest formatem służącym do opisu osób, dzięki czemu opisy mogą być przetwarzane przez maszyny, dzięki możliwościom odwołań do innych dokumentów tworzą sieci powiązań.

DOAP - format służący do opisu projektów informatycznych, w szczególności Open Source.

Przykłady użycia

SKOS - np. SWED Environmental Directory

Dublin Core - np. dLibra

FOAF - np. [Social Graph API](#), [SearchMonkey](#)

DOAP - np. generator plików DOAP (<http://crschmidt.net/semweb/doapamatic/>)

3. FOAF

Plik FOAF:

```
<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:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:admin="http://webns.net/mvcb/">
<foaf:PersonalProfileDocument rdf:about="">
  <foaf:maker rdf:resource="#me"/>
  <foaf:primaryTopic rdf:resource="#me"/>
  <admin:generatorAgent rdf:resource="http://www.ldodds.com/foaf/foaf-a-matic"/>
  <admin:errorReportsTo rdf:resource="mailto:leigh@ldodds.com"/>
</foaf:PersonalProfileDocument>
<foaf:Person rdf:ID="me">
<foaf:name>Mateusz Ziarko</foaf:name>
<foaf:title>Mr</foaf:title>
<foaf:givenname>Mateusz</foaf:givenname>
<foaf:family_name>Ziarko</foaf:family_name>
<foaf:nick>cyp3r</foaf:nick>
<foaf:mbox_sha1sum>d6ffc0014e3a780a06001d517b9136971180489</
foaf:mbox_sha1sum>
<foaf:homepage rdf:resource="http://www.webninja.eu"/>
<foaf:workplaceHomepage rdf:resource="http://www.silvermedia.pl"/>
<foaf:workInfoHomepage rdf:resource="http://www.silvermedia.pl/static/solutions"/>
<foaf:schoolHomepage rdf:resource="http://www.agh.edu.pl"/></foaf:Person>
```

</rdf:RDF>

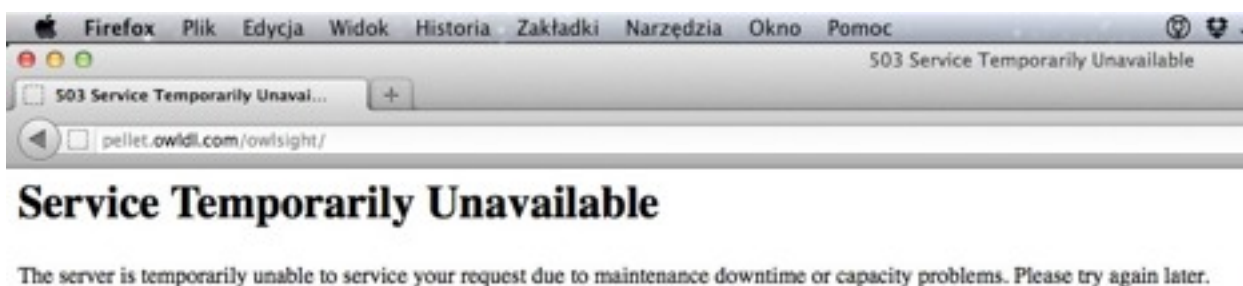
Wersja opublikowana: <http://webninja.eu/files/myFoaf.rdf>

4. Schemas

Ontologie pojawiały się w załączonym katalogu w latach 2000 - 2003. Ich wielkość (ilość klas i parametrów) jest zróżnicowana, przeważają posiadające po kilkanaście / kilkadziesiąt wpisów.

5. Ontology visualisation

Wykonanie tego punktu było niemożliwe z powodu błędu serwera podanej aplikacji:



6. Ontology reasoning

j.w.

7. Linked Open Data

- <http://lastfm.rdfize.com/> -> Artist by Artist Name
Zwracany jest dokument zawierający dane artysty / zespołu z wykorzystaniem np. FOAF, OWL czy vCard. W wygenerowanym dla przykładu pliku grupy **Linkin Park** znajduje się jej opis, opisy poszczególnych członków zespołu. Występują także powiązania z wyprodukowanymi albumami czy też innymi pokrewnymi artystami.
- <http://wiki.rkbexplorer.com/>
Wykonanie zapytania o nazwisko **Smith** zwraca dokument zawierający dane pracownika / członka zespołu projektowego. Nie są one zapisane z wykorzystaniem FOAF, a AKT mimo że w kilku miejscach się pokrywają. Występuje również dowiązanie do konkretnego projektu, a także dane tworzącego plik.
- <http://skipforward.opendfki.de/wiki/DBTropes> -> The Matrix
Bardzo rozległa struktura opisująca film / program telewizyjny poczynając od typowych danych tj. tytuł, kategoria poprzez powiązania z innymi produkcjami i komentarze.

- <http://productdb.org/> -> Coca-Cola

Struktury danych zawierają powiązania pomiędzy markami, producentami i produktami opisanymi w różnych systemach np. <http://www.productwiki.com/>

- <http://www.geonames.org/ontology/documentation.html>

Ontologie opisują i wiążą ze sobą regiony, nazwy geograficzne, a także wszystkie inne informacje które są możliwe do znalezienia na typowej mapie.

8. Semantic Web tools

- **RDFizers** - moim zdaniem najbardziej przydatne są konwertery, które możemy wykorzystać w codziennej pracy, aby w spójnym formacie udostępniać dane w sieci np. iCal -> RDF. Z mojego punktu widzenia właśnie z udostępnianiem kalendarzy w różnych formatach był zawsze problem. O ile jeszcze nie został stworzony, przydatny byłby konwerter Google Calendar -> RDF.
- **Semantic Web Browsers** - przegląd linków wykazał, że spośród podanych działa lub jest możliwych do instalacji 9 wyszukiwarek (nie załadowały się: razorbase, iLOD, FenFire, Objectviewer)
- **Semantic Web Search Engines** - działają jak każda inna wyszukiwarka typu Google czy Bing, wystarczy wpisać odpowiednią frazę i pojawiają się pasujące rekordy.
- **Applications** - dla mnie osobiście przydatne są aplikacje BBC z których zdarzało mi się korzystać. Ogólnie rzecz biorąc przydatna byłaby każda z aplikacji, która „łączyłaby” dane z wielu źródeł.

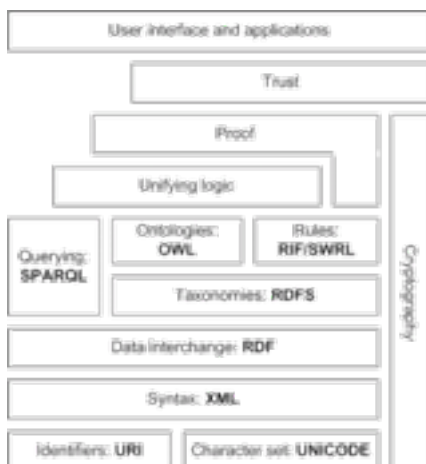
9. Control questions

- **How one can add semantic annotation to a web page?**

Nie różni się to niczym od pisania typowego kodu HTML, każdy „mikroformat” posiada ściśle określone klasy / parametry, które należy dodać do elementów strony, aby zostały one przetworzone odpowiednio przez parser danego formatu.

- **Explain the Semantic Web Stack of technologies.**

Jest to stos przedstawiający architekturę Semantic Web, poczynając od interfejsu użytkownika, poprzez warstwy ontologii i taksonomii, składnię aż po URI i zbiór znaków. Prezentuje to obrazek poniżej pochodzący z Wikipedii:



- **What is the main syntax for RDF? What are its advantages over other syntaxes?**

Jest to składnia określająca ogólne zasady pisania dokumentów RDF, które bazują na składni XML. Chodzi o wykorzystywane znaczniki i format ich zapisu. Przewagą takiej, a nie innej składni jest jej czytelność.

- **What is the role of the ontologies in the Semantic Web?**

Ontologie definiują zbiór pojęć dla pewnej dziedziny wiedzy, na której budowana jest Semantic Web. Mogą służyć jako podstawa do wnioskowania właściwości opisywanych przez ontologię pojęć.

- **What are the 4 principles of Linked Data?**

Oryginalnie brzmiące zasady:

- Use [URIs](#) to identify things.
- Use [HTTP](#) URIs so that these things can be referred to and looked up ("[dereferenced](#)") by people and [user agents](#).
- Provide useful information about the thing when its URI is dereferenced, using standard formats such as [RDF/XML](#).
- Include links to other, related URIs in the exposed data to improve discovery of other related information on the Web.

Polska interpretacja zasad:

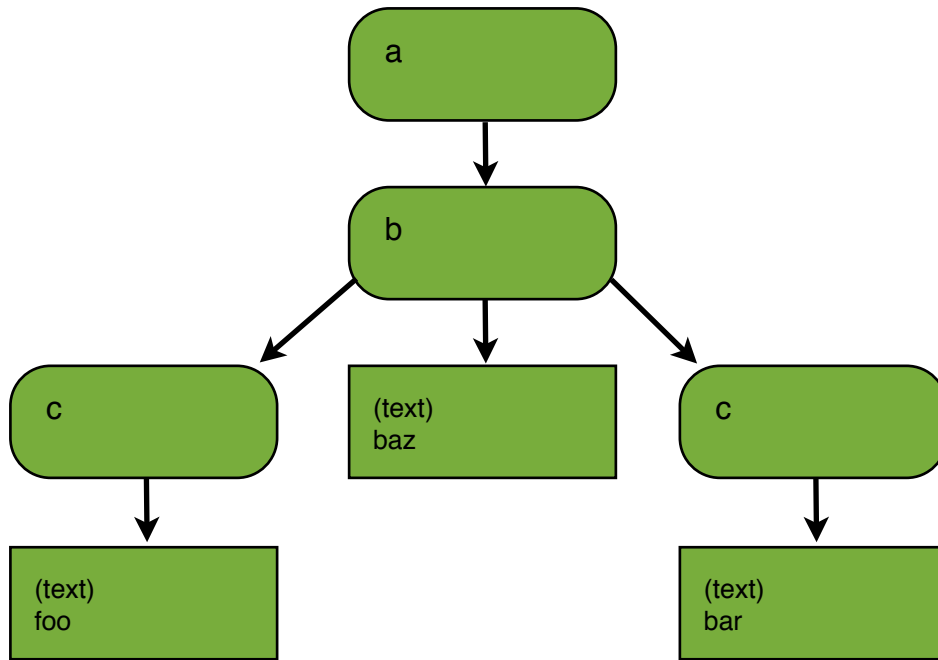
- Należy używać URI do identyfikacji rzeczy
- Należy używać HTTP URI aby mogły one zostać łatwo czytane i obsługane przez użytkownika i zdalnych agentów.
- Należy dostarczać przydatnych informacji kiedy URI jest odpowiednio obsługane, używając do tego celu standardowych formatów tj. RDF/XML.
- Należy załączać linki do innych powiązanych URI w odczytanych danych tak aby zapewnić odpowiednio szerokie powiązania do innych istniejących w sieci istotnych informacji.

Zasady te zostały już zdefiniowane do 3 bardziej ogólnych i prostych.

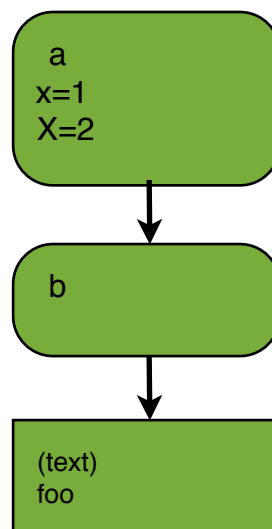
10. Warm-up: Well-formedness of the XML

- `<a><a>` - błąd, struktura musi kończyć się znacznikiem zamykającym tj. ``
- `<a>foo<a>bar` - błąd, znacznik po elemencie głównym musi być poprawnie sformułowany
- `<a>foobar` - błąd, element `` musi zostać zamknięty tagiem ``
- `<a><c>foobar</c>` - błąd, element `<c>` musi zostać zamknięty tagiem `</c>` przed ``
- `<a/>foobar` - błąd, znacznik po elemencie głównym musi być poprawnie sformułowany

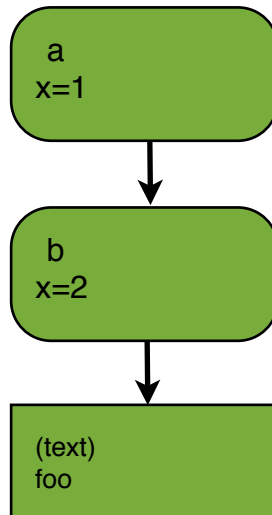
- `<a><c>foo</c>baz<c>bar</c>` - dobrze sformatowany



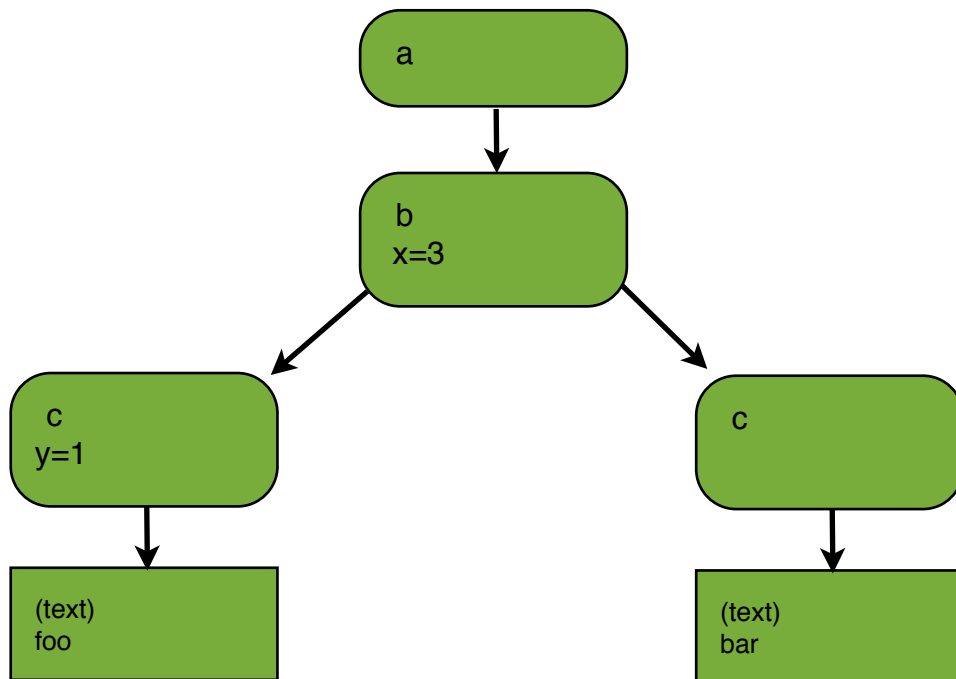
- `foo` - dobrze sformatowany



- `<b x="2">foo` - dobrze sformatowany



- `foo` - błąd, parametr „x” został już określony dla elementu `<a>`
- `<a><b x="3"><c y="1">foo</c><c>bar</c>` - dobrze sformatowany



11. Creating XML document

```
<book>
  <author>Dan Brown</author>
  <title>Cyfrowa Twierdza</title>
  <date>2009-11</date>
  <publisher>Albatros/Sonia Draga</publisher>
</book>
```

12. DTD

```
<!DOCTYPE book [

  <!ELEMENT book (author+, title, date, publisher)>
  <!ELEMENT author (#PCDATA)>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT date (#PCDATA)>
  <!ELEMENT publisher (#PCDATA)>

]>
```

13. XSD

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

  <xs:element name="book">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="author" maxOccurs="unbounded" type="xs:string"/>
        <xs:element name="title" type="xs:string"/>
        <xs:element name="date" type="xs:gYearMonth"/>
        <xs:element name="publisher" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```