

Web ngữ nghĩa

Soạn bởi: Nguyễn Bá Ngọc

Chương 3

Hà Nội-2021

Chương 3.

RDF & RDFS

Nội dung

- 3.1. Mô hình dữ liệu RDF
- 3.2. Cú pháp RDF
- 3.3. Ontology đơn giản với RDFS
- 3.4. Biểu diễn các cấu trúc đặc biệt

Nội dung

3.1. Mô hình dữ liệu RDF

3.2. Cú pháp RDF

3.3. Ontology đơn giản với RDFS

3.4. Biểu diễn các cấu trúc đặc biệt

Giới thiệu

- XML chỉ thiết lập cấu trúc dữ liệu
 - Không hàm chứa ngữ nghĩa để máy tính xử lý
- RDF giống 1 mô hình dữ liệu hơn 1 ngôn ngữ
 - Dữ liệu RDF có thể được biểu diễn theo nhiều định dạng khác nhau
- RDF quy ước các thành phần ngữ nghĩa hạn chế
 - RDFS và OWL bổ xung các từ khóa giàu ngữ nghĩa hơn

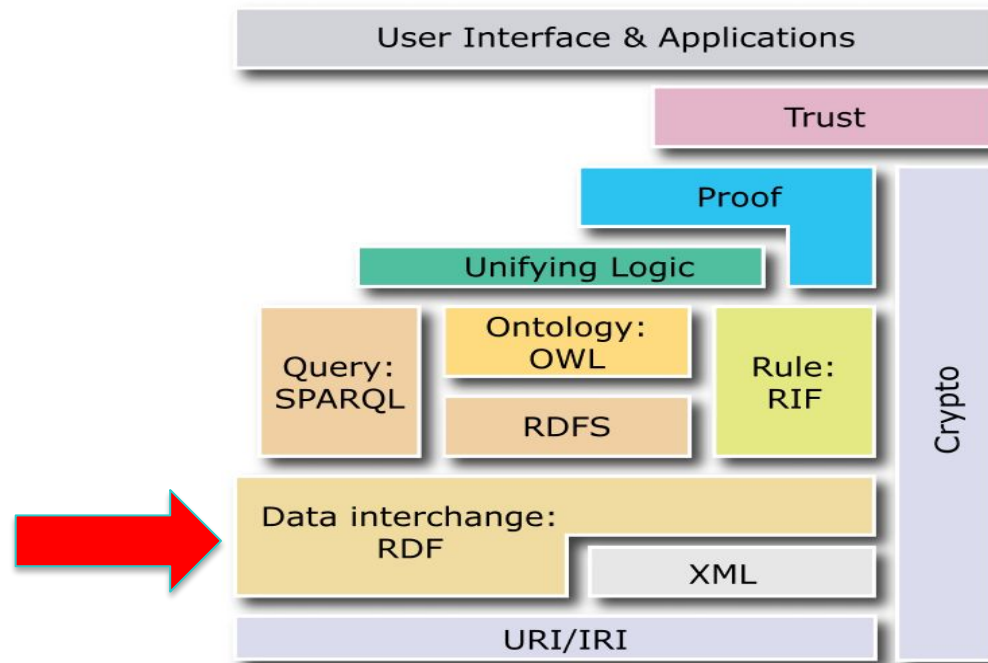
XML, RDF(S) và biểu diễn ngữ nghĩa

```
<CanBo> Nguyen Van A </CanBo>  
<GiangVien> Nguyen Van B </GiangVien>  
<HocPhan Ten="Toan roi rac">  
  <DuocDayBoi> Nguyen Van C </DuocDayBoi>  
</HocPhan>
```

- Các vấn đề ngữ nghĩa:
 - Nếu ai đó dạy 1 học phần, thì người đó có phải là giảng viên hay không?
 - Giảng viên có phải là cán bộ không?
 - *(không biểu diễn được bằng XML, nhưng có thể biểu diễn được bằng RDFS).*
- XML không quy ước ngữ nghĩa cho các nút lồng nhau
 - Người đọc có thể tự suy luận ngữ nghĩa theo cách riêng

RDF là gì?

- RDF - Resource Description Framework



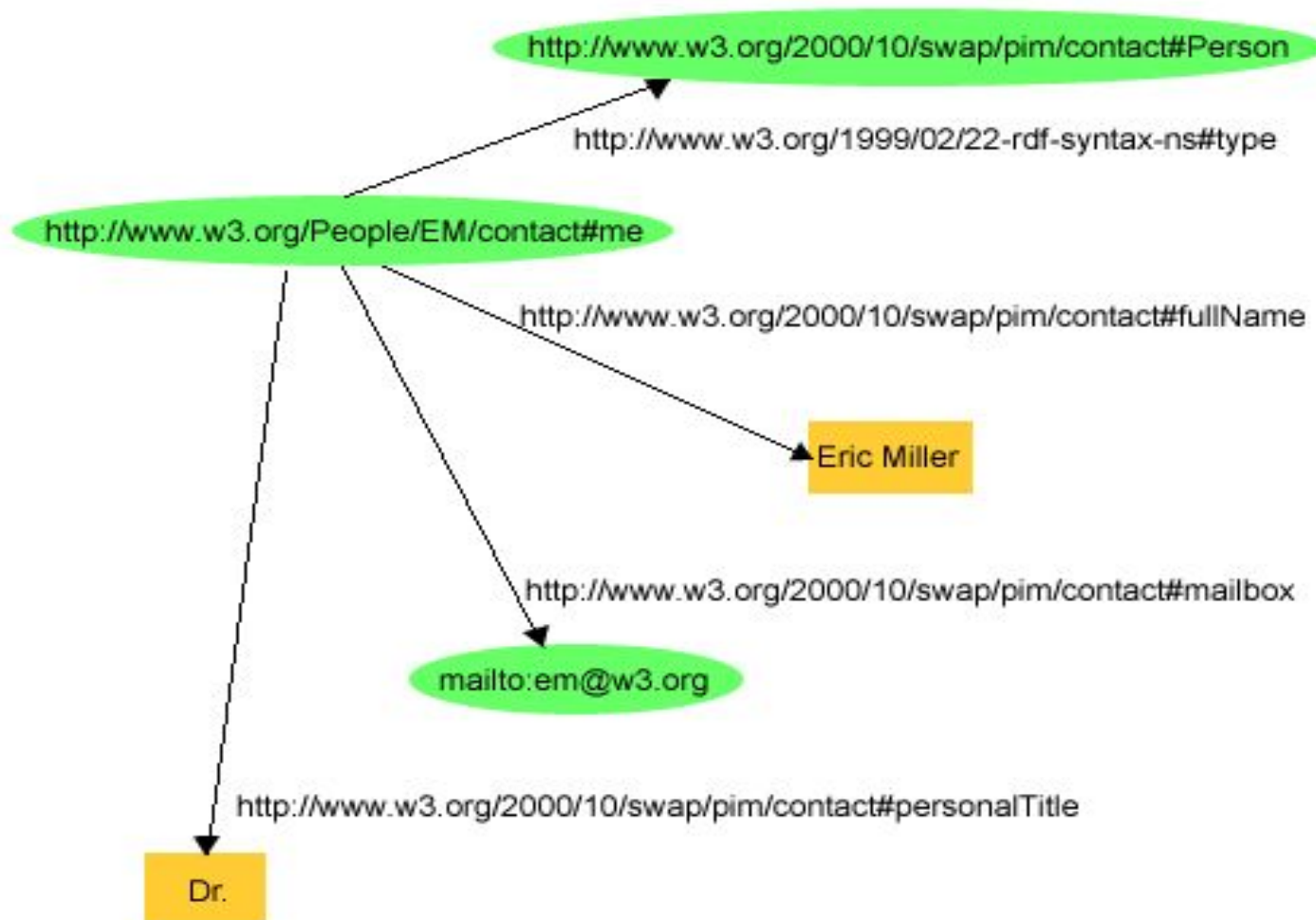
- Được sử dụng để biểu diễn các tài nguyên, có thể xử lý được bằng máy tính.

Khái quát về RDF

- **Thiết kế cơ bản:** Xác định các tài nguyên bằng các định danh Web và mô tả các tài nguyên bằng các cặp thuộc tính cùng với giá trị.
- Mô hình dữ liệu RDF là mô hình đồ thị thuần khiết
 - *(Khái niệm đồ thị thuần khiết sẽ được giải thích ngay sau)*

Ví dụ 3.1. Dữ liệu RDF

<http://www.w3.org/People/EM/contact#me> là định danh của 1 người; **Có tên là:** Eric Miller; **Email:** em@w3.org; **Học vị:** Dr.



Bộ-3

Các tài nguyên được mô tả bằng các câu theo cấu trúc đơn giản gồm 3 thành phần:

- **Tài nguyên - subject**, thành phần thông tin đang được mô tả.
- **Thuộc tính - predicate**, thuộc tính của tài nguyên đang được mô tả;
- **Giá trị - object**, xác định giá trị của thuộc tính của tài nguyên đang được mô tả.

Câu RDF = Bộ-3, triple

Ví dụ 3.2a. Câu RDF

```
<http://www.example.org/index.html>  
  <http://semweb.edu.vn/vocab#hasCreator>  
  "John Smith" .
```

Trong ví dụ này:

- **Tài nguyên/Chủ thể - Subject** được xác định bởi URI
http://www.example.org/index.html
- **Thuộc tính - Predicate**,
http://semweb.edu.vn/vocab#hasCreator
(*hasCreator - Được tạo bởi*).
- **Giá trị - object**, là chuỗi "John Smith"

Ví dụ 3.2b. Câu RDF

<http://www.example.org/index.html>

<http://semweb.edu.vn/vocab#hasCreator>

<http://www.example.org/people/John-Smith> .

Trong ví dụ này:

- **Tài nguyên/Chủ thể - Subject:**

http://www.example.org/index.html

- **Thuộc tính - Predicate:**

<http://semweb.edu.vn/vocab#hasCreator>

- **Giá trị - object:**

<http://www.example.org/people/John-Smith>

Định dạng ntriples

- Định dạng lưu trữ đơn giản nhất
- ntriples - tập hợp các bộ-3
 - Mỗi bộ-3 được kết thúc bởi 1 dấu chấm;
 - Các URIs nằm trong các cặp dấu < >.

<http://www.example.org/index.html>

<http://purl.org/dc/elements/1.1/creator>

<http://www.example.org/staffid/85740> .

<http://www.example.org/index.html>

<http://www.example.org/terms/creation-date> "August 16, 1999" .

<http://www.example.org/index.html>

<http://purl.org/dc/elements/1.1/language> "en" .

Đơn giản nhưng không tối ưu, các URI dài có thể chiếm nhiều dung lượng

Các loại đồ thị:
Đồ thị thuần khiết &
Đồ thị thuộc tính

Mô hình đồ thị thuần khiết

- Đồ thị thuần khiết chỉ bao gồm các cạnh giữa các cặp đỉnh
 - Cạnh có thể có hướng hoặc vô hướng, có thể có nhãn hoặc không.
- Đồ thị thuần khiết có thể được biểu diễn như 1 tập không quy ước thứ tự của các bộ-3 (chủ thể-thuộc tính-giá trị).
 - Nếu có hướng thì thuộc tính đi từ chủ thể tới giá trị.
- Không được có các nút không phải chủ thể hoặc giá trị của bất kỳ bộ-3 nào.

Mô hình đồ thị RDF

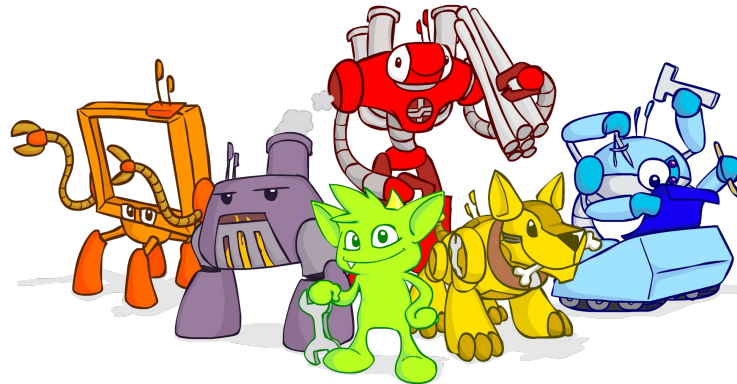
- Đồ thị RDF là đồ thị thuần khiết cùng với các ràng buộc
 - Chủ thể và thuộc tính được xác định bởi các URIs
 - Giá trị có thể là URI hoặc các hằng giá trị.
- RDF bao gồm các URIs tiêu chuẩn và có quy ước ngữ nghĩa, ví dụ từ type được xác định bởi
 - <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
 - Được sử dụng để định kiểu cho tài nguyên.
- RDF có nhiều cách để biểu diễn các giá trị phức tạp (chứa nhiều thành phần) và các nhóm tài nguyên bằng các cấu trúc tổng hợp đơn giản.

Đồ thị thuộc tính

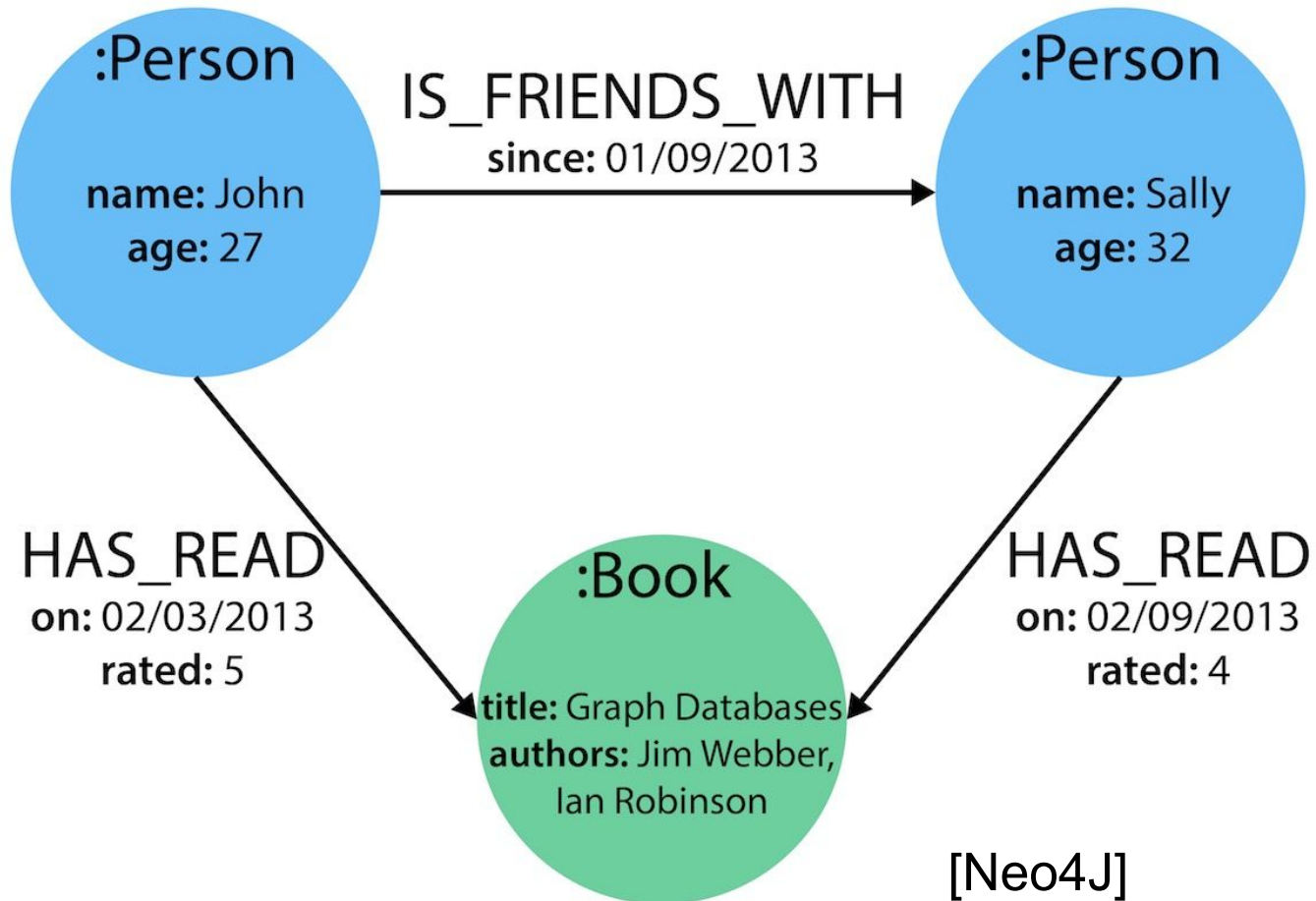
- Các CSDL đồ thị đã được sử dụng ngày càng phổ biến trong thời gian gần đây.
- Mở rộng đơn giản đối với mô hình đồ thị thuần khiết cho phép cả nút và cạnh có thể có các thuộc tính.
- Ví dụ các thuộc tính đơn giản:
 - Age: 30
 - Date: 1990-09-21

Các công nghệ đồ thị thuộc tính

- **Neo4J** - CSDL đồ thị thuộc tính được sử dụng phổ biến.
- **OrientDB** - Hỗ trợ đồng thời cả mô hình đồ thị thuộc tính và mô hình quan hệ.
- **Apache TinkerPop** là 1 dự án mã nguồn mở.
- **Neptune** của **Amazon** được xây dựng cho nền tảng đám mây.



Ví dụ 3.3. Đồ thị thuộc tính



URIs và URIsrefs

URIs

- Uniform Resource Identifiers (URIs)
- Mở rộng hơn so với URL, URI có thể được sử dụng để xác định các thứ không có trong không gian mạng
- Không có cơ quan điều phối việc tạo URIs hoặc sử dụng như thế nào
 - Có nhiều loại URI (ví dụ, URL - http) phụ thuộc vào các hệ thống tập trung như các hệ thống DNS.
 - Tuy nhiên URI có thể hoàn toàn phi tập trung.

Tham chiếu URI và URIref

- URIref: URI với mã phân đoạn ở cuối, ví dụ <http://example.org/index.html#section2>
- Trường hợp sử dụng phân đoạn
 - Các phân đoạn HTML tham chiếu các vị trí trong 1 trang
 - Các phân đoạn RDF tham chiếu tới các tài nguyên trong 1 bộ từ vựng RDF được xác định bởi URI, ví dụ:
 - <http://www.w3.org/2004/02/skos/core#>
 - Bộ từ vựng để mô tả các chủ đề
 - <http://www.w3.org/2004/02/skos/core#broader>
 - Thuộc tính: Khái niệm rộng hơn trong bộ từ vựng SKOS Core
- Tương tự các URLs, các URIrefs có thể là tuyệt đối hoặc tương đối





URIrefs trong RDF

- Trình duyệt và các ứng dụng Web khác sử dụng URIrefs để xác định các tài nguyên:
 - Trình duyệt sử dụng các URIrefs để tải về các tài nguyên.
 - Trong mô hình dữ liệu RDF, URIrefs được sử dụng để xác định các tài nguyên thông tin: Bao gồm cả các tài nguyên có trong Web và cả các tài nguyên không có trong Web.
- Nên sử dụng các URIs với giao thức HTTP được gắn với các tài nguyên có trong môi trường Web (có thể tải về được).

Tải về nội dung

- Giao thức HTTP cho phép thiết lập định dạng dữ liệu trả về, ví dụ, HTML, XML, RDF được mã hóa theo nhiều cách.
 - Sử dụng cùng 1 URL, ví dụ
 - `http://dbpedia.org/resource/Alan_Turing`
 - Có thể cung cấp nội dung theo định dạng thuận tiện nhất (có thể khác nhau) cho người và máy:
 - `curl -LH "Accept:text/html"`
`http://dbpedia.org/resource/Alan_Turing`
 - `curl -LH "Accept: application/rdf+xml"`
`http://dbpedia.org/resource/Alan_Turing`

Ví dụ 3.4a. Nội dung cho người

 Browse using  Formats  Faceted Browser  Sparql Endpoint

About: Alan Turing

An Entity of Type : [scientist](#), from Named Graph : <http://dbpedia.org>, within Data Space : <dbpedia.org>

Alan Mathison Turing OBE FRS (/ˈtʃʊərɪŋ/; 23 June 1912 – 7 June 1954) was an English computer scientist, mathematician, logician, cryptanalyst and theoretical biologist. He was highly influential in the development of theoretical computer science, providing a formalisation of the concepts of algorithm and computation with the Turing machine, which can be considered a model of a general purpose computer. Turing is widely considered to be the father of theoretical computer science and artificial intelligence.

Property	Value
dbo:abstract	<ul style="list-style-type: none">Alan Mathison Turing OBE FRS (/ˈtʃʊərɪŋ/; 23 June 1912 – 7 June 1954) was an English computer scientist, mathematician, logician, cryptanalyst and theoretical biologist. He was highly influential in the development of theoretical computer science, providing a formalisation of the concepts of algorithm and computation with the Turing machine, which can be considered a model of a general purpose computer. Turing is widely considered to be the father of theoretical computer science and artificial intelligence. During the Second World War, Turing worked for the Government Code and Cypher School (GC&CS) at Bletchley Park, Britain's codebreaking centre. For a time he led Hut 8, the section responsible for German naval cryptanalysis. He devised a number of techniques for speeding the breaking of German ciphers, including improvements to the pre-war Polish bombe method, an electromechanical machine that could find settings for the Enigma machine. Turing played a pivotal role in cracking intercepted coded messages that enabled the Allies to defeat the Nazis in many crucial engagements, including the Battle of the Atlantic; it has been estimated that this work shortened the war in Europe by more than two years and saved over fourteen million lives. After the war, he worked at the National Physical Laboratory, where he designed the ACE, among the first designs for a stored-program computer. In 1948 Turing joined Max Newman's Computing Machine Laboratory at the Victoria University of Manchester, where he helped develop the Manchester computers and became interested in mathematical biology. He wrote a paper on the chemical basis of morphogenesis, and predicted oscillating chemical reactions such as the Belousov–Zhabotinsky reaction, first observed in the 1960s. Turing was prosecuted in 1952 for homosexual acts, when such behaviour was still a criminal act in the UK. He accepted treatment with DES (chemical castration) as an alternative to prison. Turing died in 1954, 16 days before his 42nd birthday, from cyanide poisoning. An inquest determined his death as suicide, but it has been noted that the known evidence is equally consistent with accidental poisoning. In 2009, following an Internet campaign, British Prime Minister Gordon Brown made an on behalf of the British government for "the appalling way he was treated." Queen Elizabeth II granted him a posthumous pardon in 2013. ^(en)
dbo:almaMater	<ul style="list-style-type: none">dbr:Princeton_Universitydbr:King's_College,_Cambridge
dbo:award	<ul style="list-style-type: none">dbr:Smith's_Prize
dbo:birthDate	<ul style="list-style-type: none">1912-06-23 ^(xsd:date)1912-6-23

Ví dụ 3.4b. Dữ liệu cho máy

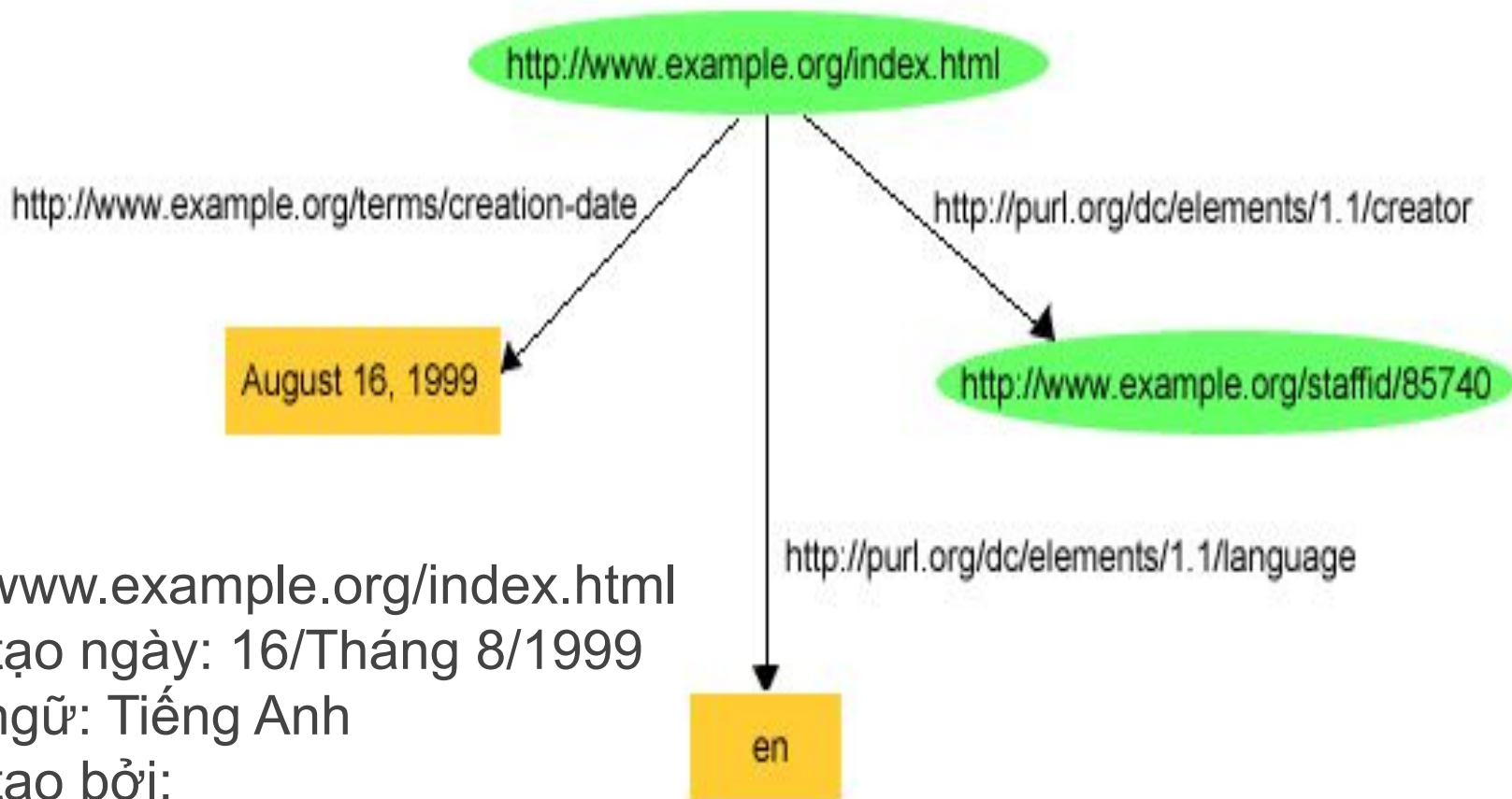
```
bangoc:~$ curl -LH "Accept: application/rdf+xml" http://dbpedia.org/resource/Alan_Turing
<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:dbo="http://dbpedia.org/ontology/"
  xmlns:dct="http://purl.org/dc/terms/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:prov="http://www.w3.org/ns/prov#"
  xmlns:dbp="http://dbpedia.org/property/"
  xmlns:ns8="http://purl.org/linguistics/gold/" >
  <rdf:Description rdf:about="http://dbpedia.org/resource/Alonzo_Church">
    <dbo:doctoralStudent rdf:resource="http://dbpedia.org/resource/Alan_Turing" />
  </rdf:Description>
  <rdf:Description rdf:about="http://dbpedia.org/resource/Allan_Turing">
    <dbo:wikiPageRedirects rdf:resource="http://dbpedia.org/resource/Alan_Turing" />
  </rdf:Description>
  <rdf:Description rdf:about="http://dbpedia.org/resource/A._Turing">
    <dbo:wikiPageRedirects rdf:resource="http://dbpedia.org/resource/Alan_Turing" />
  </rdf:Description>
  <rdf:Description rdf:about="http://dbpedia.org/resource/Metamagical_Themas">
    <dbo:nonFictionSubject rdf:resource="http://dbpedia.org/resource/Alan_Turing" />
  </rdf:Description>
  <rdf:Description rdf:about="http://dbpedia.org/resource/A._M._Turing">
    <dbo:wikiPageRedirects rdf:resource="http://dbpedia.org/resource/Alan_Turing" />
  </rdf:Description>
  <rdf:Description rdf:about="http://dbpedia.org/resource/Alan_Turning">
    <dbo:wikiPageRedirects rdf:resource="http://dbpedia.org/resource/Alan_Turing" />
  </rdf:Description>
  <rdf:Description rdf:about="http://dbpedia.org/resource/Alan_Turing">
    <rdf:type rdf:resource="http://dbpedia.org/class/yago/WikicatBritishCryptographers" />
    <rdf:type rdf:resource="http://dbpedia.org/class/yago/WikicatEnglishInventors" />
    <rdf:type rdf:resource="http://dbpedia.org/class/yago/Theorist110706812" />
    <rdf:type rdf:resource="http://dbpedia.org/class/yago/Decoder109995398" />
    <rdf:type rdf:resource="http://umbel.org/umbel/rc/PersonWithOccupation" />
    <rdf:type rdf:resource="http://dbpedia.org/class/yago/NonreligiousPerson109625789" />
    <rdf:type rdf:resource="http://dbpedia.org/class/yago/Engineer109615807" />
  </rdf:Description>
</rdf:RDF>
```

Đồ thị RDF

Đồ thị RDF

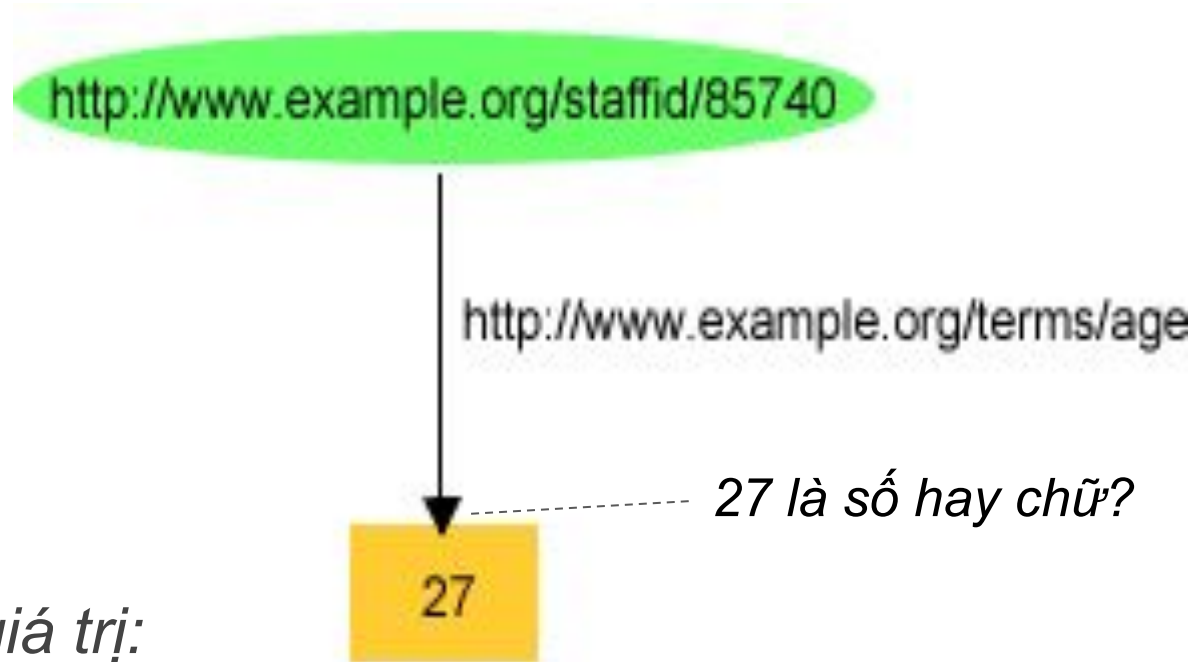
- Mô hình hóa các câu RDF bằng nút và cạnh trong đồ thị
- Mỗi câu được biểu diễn bởi 1 nút cho tài nguyên, 1 nút cho giá trị và 1 cạnh cho thuộc tính.
- Mỗi nút có thể được xác định bởi 1 URIref hoặc có thể là hằng giá trị hoặc nút rỗng.
- Mỗi cạnh được xác định bởi 1 URIref.
- Đồ thị RDF là 1 đồ thị có hướng.
- Các thuộc tính cũng có thể là các nút trong đồ thị RDF - có thể là chủ thể hoặc đối tượng của các bộ-3, ví dụ:
 - :has_parent owl:inverseOf :has_child

Ví dụ 3.5. Đồ thị RDF



`http://www.example.org/index.html`
được tạo ngày: 16/Tháng 8/1999
ngôn ngữ: Tiếng Anh
được tạo bởi:
`http://example.org/staffed/85740`

Hằng giá trị



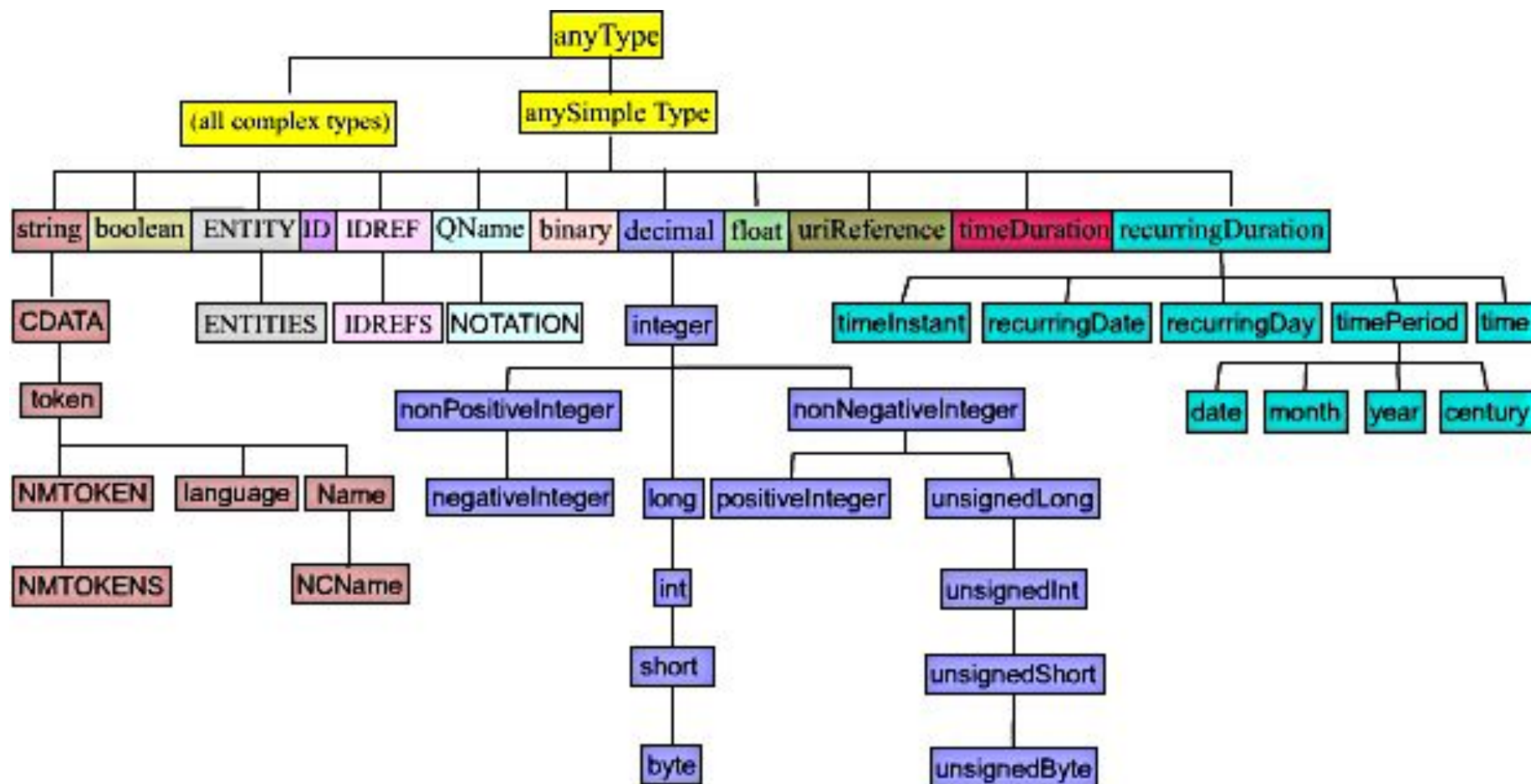
RDF có 2 định dạng giá trị:

- Giá trị thô được viết theo quy cách và có thể có thẻ ngôn ngữ, ví dụ: "27", "Xin chào"@vi, "Hello world"@en
- Giá trị có định kiểu là 1 chuỗi được gắn với 1 URIref cho 1 kiểu dữ liệu XMLS cụ thể, ví dụ:
- "27"^^http://www.w3.org/2001/XMLSchema#integer
- "27"^^xsd:int

Các kiểu dữ liệu cho giá trị

- Tài liệu RDF có thể sử dụng bất kỳ hệ kiểu dữ liệu nào
 - Tuy nhiên hệ kiểu trong lược đồ XML được sử dụng phổ biến nhất trong thực tế.
- Lược đồ XML định nghĩa nhiều kiểu dữ liệu cho các giá trị chân lý, số nguyên, số thực dấu chấm động, thời gian, ngày, v.v...
- Không gian tên: <http://www.w3.org/2001/XMLSchema#>

Các kiểu dữ liệu trong lược đồ XML



Qnames cho URIs

- Các URIs thường rất dài
- Chúng ta có thể sử dụng các định danh XML - QName, XML qualified name để thay thế cho các tham chiếu URI đầy đủ
 - http://dbpedia.org/page/Alan_Turing
 - `dbp:Alan_Turing`
- Qnames có tiền tố được gắn với 1 không gian tên URI, dấu: và tên địa phương
- Khái niệm tên và không gian tên được sử dụng trong RDF bắt nguồn từ XML

Nội dung

3.1. Mô hình dữ liệu RDF

3.2. Cú pháp RDF

3.3. Ontology đơn giản với RDFS

3.4. Biểu diễn các cấu trúc đặc biệt

Định kiểu tài nguyên

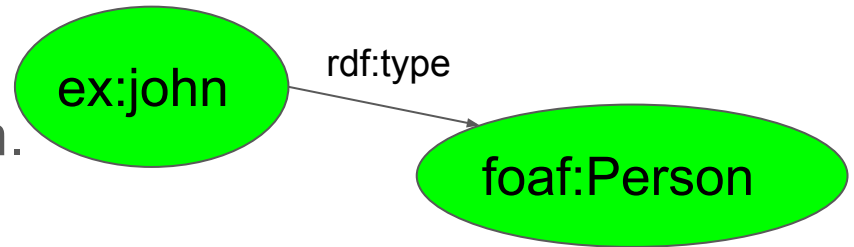
- Thuộc tính `rdf:type` được sử dụng để thiết lập kiểu của tài nguyên

- `ex:john` **`rdf:type`** `foaf:Person`.

- `<http://example.org/john >`

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

`<http://xmlns.com/foaf/0.1/Person> .`

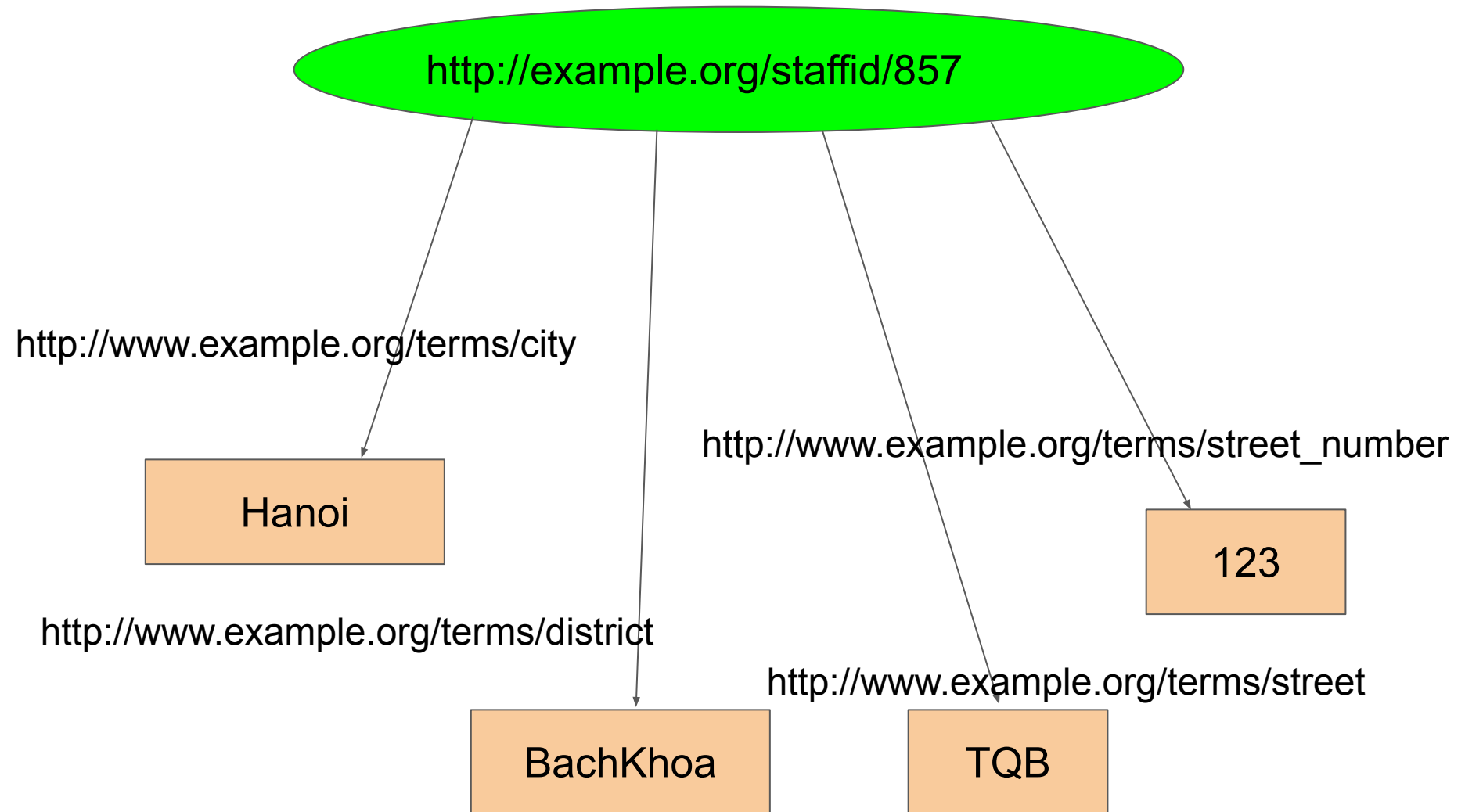


- RDFS thêm vào khái niệm lớp con & các ràng buộc giữa các thuộc tính & kiểu của tham số của nó
- OWL bổ xung thêm nhiều ràng buộc khác

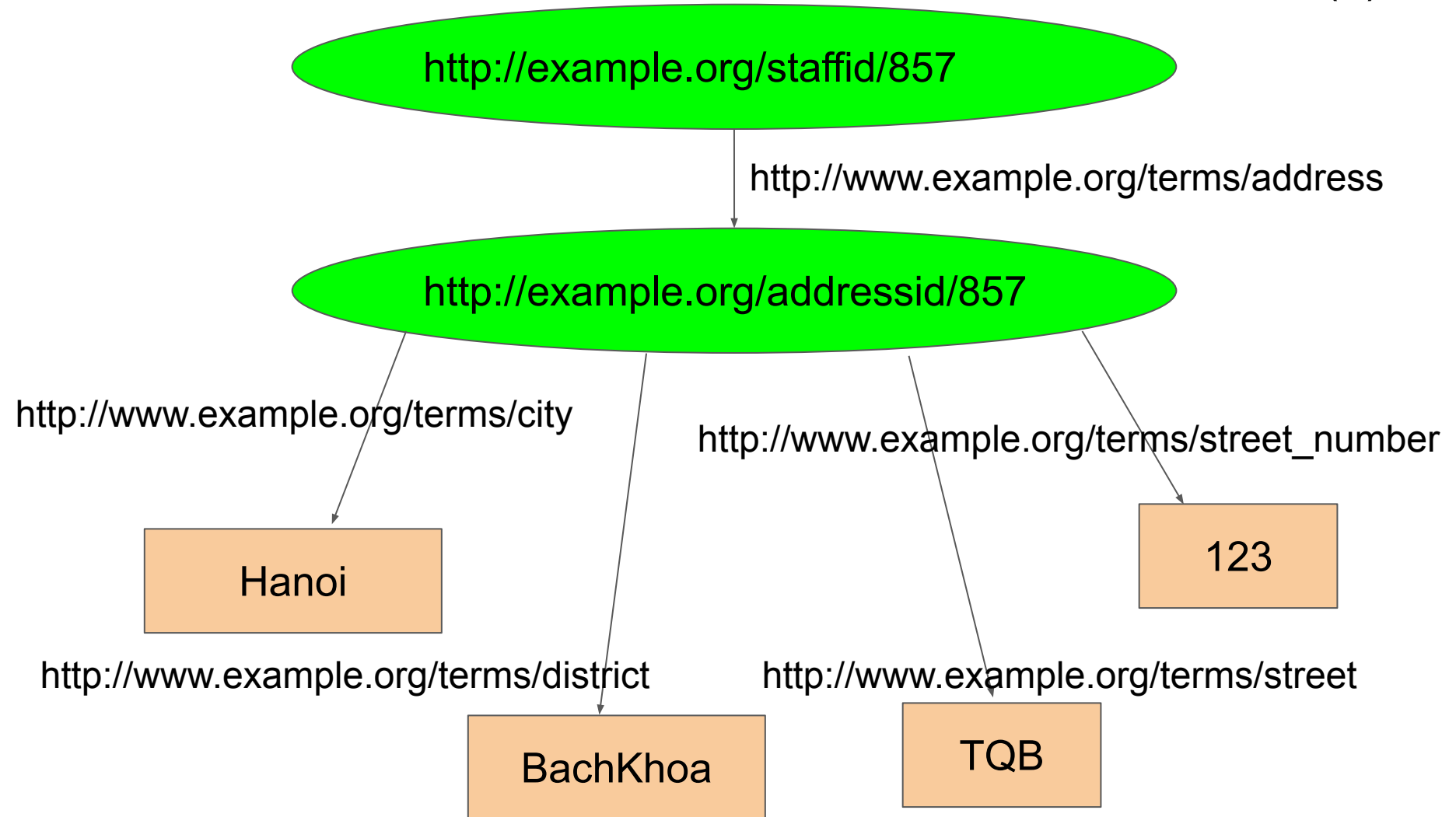
Dữ liệu có cấu trúc trong RDF

- Ví dụ bộ-3
ex:857 ex:address "123 TQB, BachKhoa, HaNoi".
- Cách biểu diễn nào trong 2 cách sau tốt hơn
 - Thiết lập 4 thuộc tính (ex:address_number, ex:address_street, ex:address_district, ex:address_city) cho tài nguyên ex:857.
 - Tạo 1 tài nguyên biểu diễn địa chỉ và thiết lập giá trị của thuộc tính ex:address cho tài nguyên ex:857.

Ví dụ 3.6. Dữ liệu có cấu trúc trong RDF



Ví dụ 3.6. Dữ liệu có cấu trúc trong RDF₍₂₎

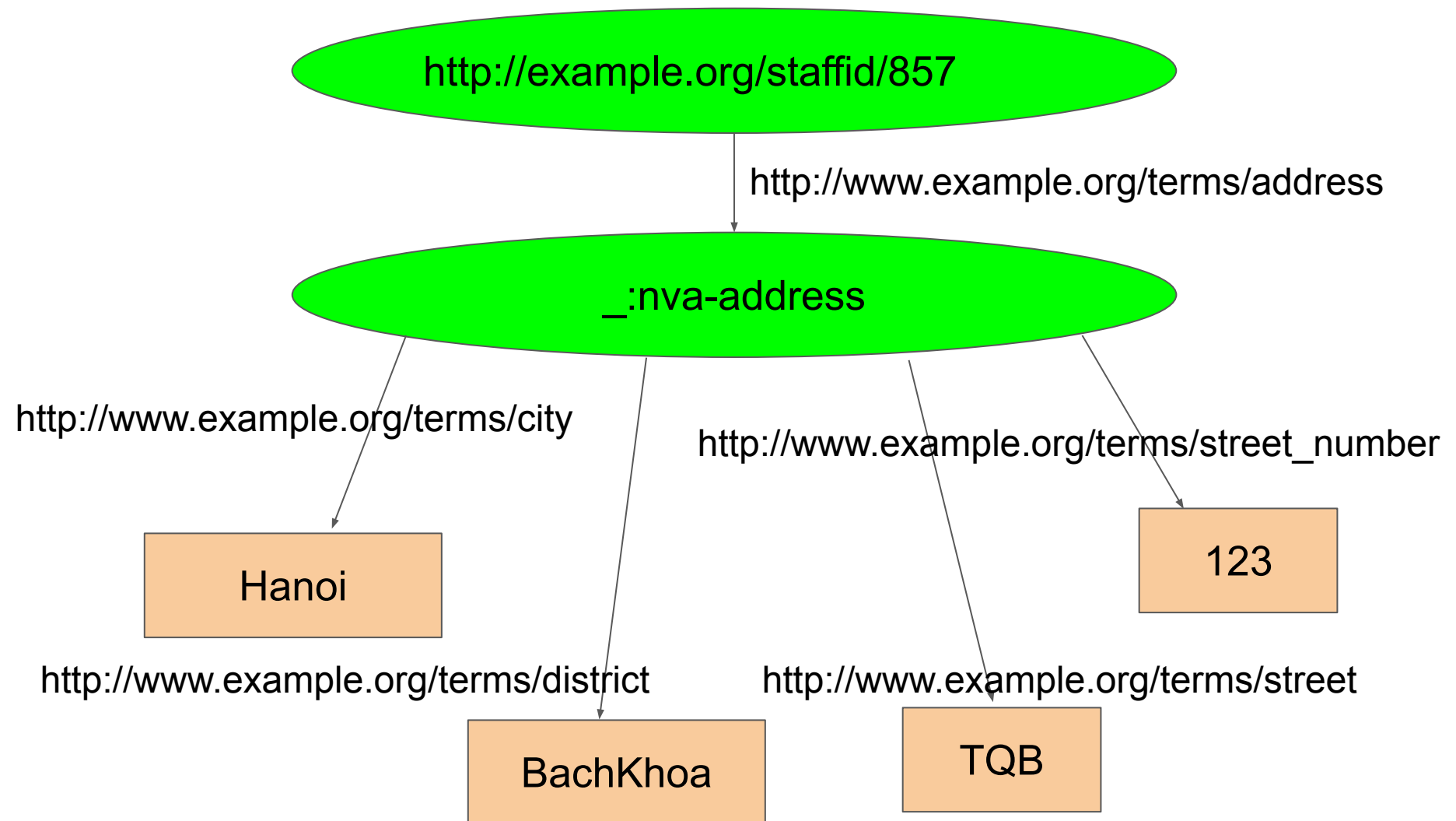


Cách triển khai này có thể dẫn đến nhiều URIsrefs trung gian

Dữ liệu có cấu trúc trong RDF₍₂₎

- Sử dụng các tài nguyên hỗ trợ để biểu diễn các cấu trúc có thể dẫn đến 1 lượng lớn các URI
 - Có những cấu trúc không được tham chiếu từ bên ngoài, vì vậy sử dụng các định danh toàn cục sẽ gây lãng phí
- RDF cho phép sử dụng nút rỗng để giải quyết vấn đề này
 - Các nút có id trong không gian tên _ là các nút rỗng
 - Ví dụ, _:

Ví dụ 3.7. Dữ liệu có cấu trúc, bnode



Id của nút rỗng

ex:857 ex:address _:NVAaddress .

_:NVAaddress ex:street "TQB" .

_:NVAaddress ex:city "Hanoi" .

- Các bnode khác nhau phải có id khác nhau
- Id của bnode chỉ có phạm vi trong 1 đồ thị
 - Các bnode có cùng id nhưng thuộc các đồ thị khác nhau là các bnode khác nhau
 - Hợp nhất các đồ thị có thể kéo theo yêu cầu đổi tên các bnode để tránh xung đột.
- bnode chỉ có thể xuất hiện ở vị trí chủ thể hoặc giá trị, không được sử dụng như thuộc tính trong bộ-3.

Các trường hợp sử dụng bnode

- Biểu diễn mối quan hệ nhiều ngôi trong RDF. Ví dụ, quan hệ giữa NVA với đường phố, thành phố, quận, số nhà.
- Viết các câu mô tả những tài nguyên chưa có URI nhưng có liên hệ với các tài nguyên khác
 - Ví dụ, nhà xuất bản của một cuốn sách

Mô tả đối tượng khuyết định danh

- Giả sử chúng ta muốn tạo các mô tả về Jane nhưng không có URI của Jane
 - Tạo 1 nút rỗng để kết nối các thông tin về Jane

`_:jane exterms:mailbox <mailto:jane@example.org> .`

`_:jane rdf:type exterms:Person .`

`_:jane exterms:name "Jane Smith" .`

`_:jane exterms:empID "23748" .`

`_:jane exterms:age "26" .`

Biểu diễn phép đo

Theo kết quả của 1 phép đo sông Nile có độ dài là 6853 Km

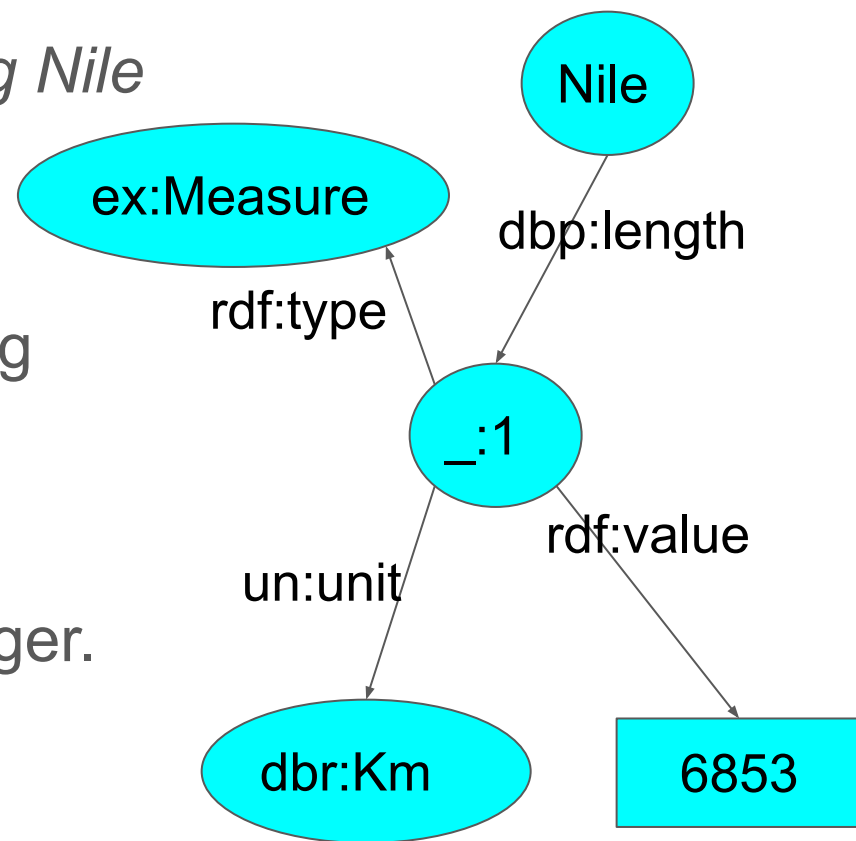
- Kết quả của các phép đo thường bao gồm giá trị số cùng với đơn vị đo

`_:1 rdf:type ex:Measure.`

`_:1 rdf:value "6853"^^xsd:integer.`

`_:1 un:units dbr:Kilometre.`

- Thuộc tính value trong không gian tên rdf không được thiết lập ngữ nghĩa cụ thể.



Định dạng lưu trữ

Định dạng lưu trữ RDF

- Đồ thị là một mô hình trực quan
 - *Không phải 1 định dạng lưu trữ (định dạng văn bản để trao đổi, lưu trữ, và xử lý bằng máy tính).*
- Các định dạng phổ biến:
 - **Turtle** - gọn nhẹ, dễ đọc và dễ viết hơn với con người
 - **XML/RDF** - Thiết kế ban đầu, có nhiều nền tảng phần mềm hỗ trợ
 - **JSON-LD** - Các câu RDF như những đối tượng JSON
 - **Ntriples** - đơn giản, dễ xử lý nhưng có phần cồng kềnh.
- Một số định dạng đặc biệt
 - **TriG** - Định dạng cho đồ thị có tên
 - **RDFa** - RDF nhúng trong các thuộc tính HTML

Định dạng XML cho 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:bib="http://daml.umbc.edu/ontologies/bib/">
<rdf:Description rdf:about="http://umbc.edu/~finin/talks/idm02/">
  <dc:title>Intelligent Information Systems on the Web </dc:title>
  <dc:creator>
    <rdf:Description >
      <bib:name>Tim Finin</bib:name>
      <bib:email>finin@umbc.edu</bib:email>
      <bib:aff rdf:resource="http://umbc.edu/" />
    </rdf:Description>
  </dc:creator>
</rdf:Description>
</rdf:RDF>
```

RDF/XML là quy chuẩn W3C
được sử dụng rộng rãi cho lưu
trữ và trao đổi dữ liệu RDF.

*Tuy nhiên đọc/viết XML
tương đối khó cho người*

Định dạng XML cho RDF₍₂₎

```
<rdf:RDF
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:bib="http://daml.umbc.edu/ontologies/bib/">
```

```
<rdf:Description
```

```
  rdf:about="http://umbc.edu/~finin/talks/idm02/">
```

```
    <dc:title>Intelligent Information Systems on the Web
```

```
</dc:title>
```

```
    <dc:creator>
```

```
      <rdf:Description >
```

```
        <bib:name>Tim Finin</bib:name>
```

```
        <bib:email>finin@umbc.edu</bib:email>
```

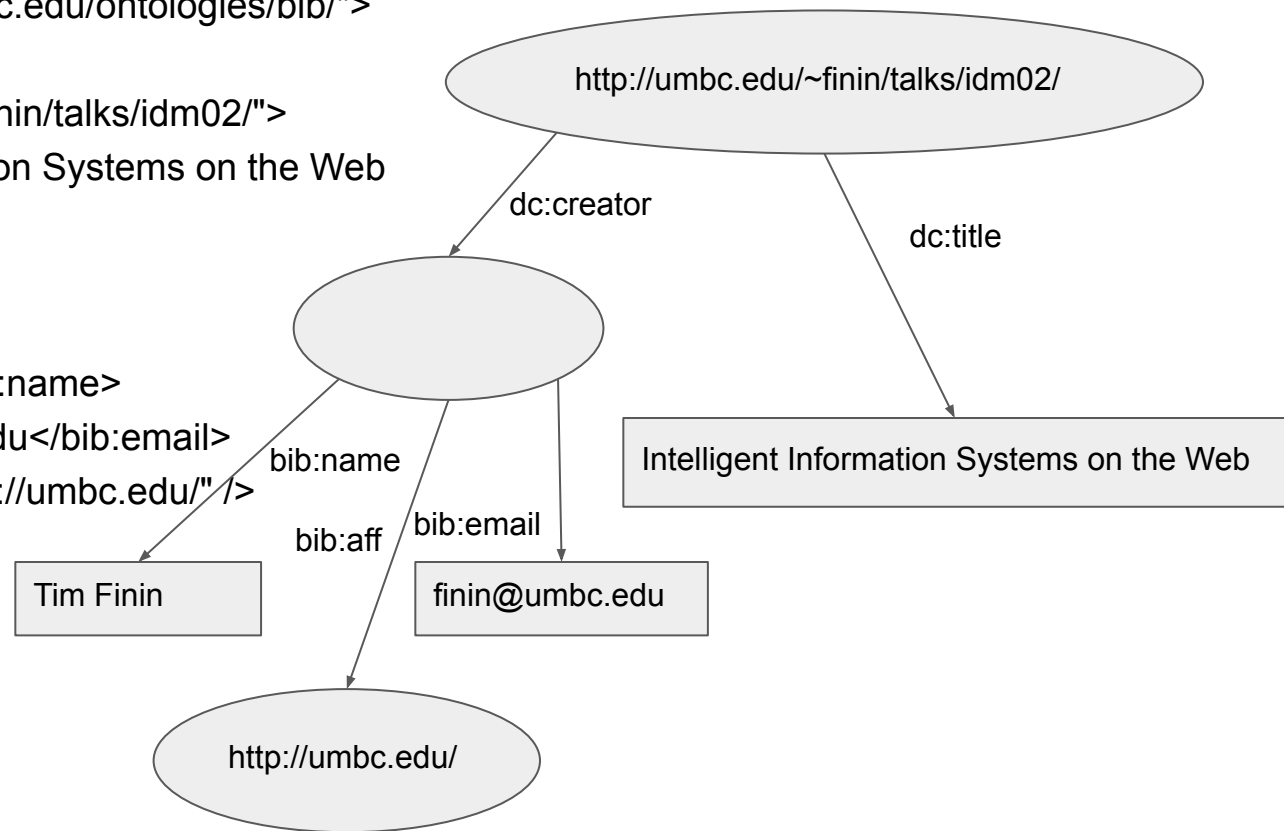
```
        <bib:aff rdf:resource="http://umbc.edu/" />
```

```
      </rdf:Description>
```

```
    </dc:creator>
```

```
</rdf:Description>
```

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



Định dạng Turtle cho RDF

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

@prefix dc: <http://purl.org/dc/elements/1.1/> .

@prefix bib: <http://daml.umbc.edu/ontologies/bib/> .

<http://umbc.edu/~finin/talks/idm02/>

dc:title "Intelligent Information Systems on the Web " ;

dc:creator _:b0 .

Turtle cũng là quy chuẩn W3C.

_:b0 bib:name "Tim Finin" ;

bib:email "finin@umbc.edu" ;

bib:aff <http://umbc.edu/> .

Thuận tiện cho người, có thể biên soạn bằng trình soạn thảo văn bản.

Ntriples

- Tập hợp các bộ-3, mỗi bộ-3 được kết thúc bởi dấu chấm/"."
- URIs được đặt trong cặp dấu ngoặc < >; không sử dụng QNames; hằng giá trị được đặt trong nháy kép.
- Dễ đọc và xử lý; là định dạng tải về phổ biến cho các bộ dữ liệu RDF (ví dụ, DBpedia)
- Sử dụng nhiều ký tự do các URLs lặp lại, tuy nhiên có tỉ lệ nén cao
 - Tốt cho lưu trữ

```
<http://example.org/Turing><http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .  
<http://example.org/Turing> <http://xmlns.com/foaf/0.1/name> "Alan Turing" .  
<http://www.w3.org/2001/sw/RDFCore/ntriples/> <http://xmlns.com/foaf/0.1/mbox> <mailto:alan@turing.org> .  
<http://example.org/Turing> <http://xmlns.com/foaf/0.1/mbox> <mailto:alan@turing.org> .
```

Turtle

- Ntriples \subset Turtle \subset N3
- Gọn nhẹ, dễ đọc, dễ viết và xử lý
- Qnames, bnode (biểu diễn bằng []), các dấu ; và ,

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

@prefix dc: <<http://purl.org/dc/elements/1.1/>> .

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

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

dc:title "RDF/XML Syntax Specification (Revised)" ;

dc:creator [foaf:name "Dave Beckett";

foaf:mbox <<mailto:dave@beckett.org>> ,

<<mailto:dbeck@gmail.com>>

] .

Khái quát cú pháp Turtle

- Dòng @PREFIX khai báo viết tắt không gian tên
- Mẫu bộ-3 cơ bản:

Subject predicate1 object1; predicate2 object2;
predicate3 object3, object4.

Tương đương với:

Subject predicate1 object1 .

Subject predicate2 object2 .

Subject predicate3 object3 .

Subject predicate3 object4.

- Ký hiệu đặc biệt, dành riêng cho rdf:type
 - :john **a foaf:Person** .
 - Tương đương với :john rdf:type foaf:Person .
- Ký hiệu nút rỗng ([])
 - :john foaf:knows [a foaf:Person; foaf:nick "Bob"] .

N3 (Notation3)

- N3 là một định dạng tương tự turtle, ban đầu được phát triển bởi Tim Berners Lee.
- Hỗ trợ luật suy diễn
- Không phải là quy chuẩn W3C
 - Một số tính năng không tương thích với OWL
 - Được thay thế bởi Turtle
 - Có lẽ sẽ không bao giờ trở thành quy chuẩn W3C

Chuyển đổi giữa các định dạng

- Hầu hết các phần mềm WebNN hiện đại (Protégé, Jena, Sesame v.v...) đều có thể đọc và ghi rdf theo các định dạng chính

Một tài liệu Turtle đơn giản

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

@prefix : <http://semweb.edu.vn/example#> .

:john a foaf:Person; foaf:gender "Male";

foaf:name "John Smith", "Johnny Smith";

foaf:knows :mary,

[a foaf:Person;

foaf:mbox <mailto:mary.smith@gmail.com>] .

:mary a foaf:Person; foaf:name "Mary Smith" .

Thử chuyển đổi giữa các định dạng

Nội dung

3.1. Mô hình dữ liệu RDF

3.2. Cú pháp RDF

3.3. Ontology đơn giản với RDFS

3.4. Biểu diễn các cấu trúc đặc biệt



Giới thiệu RDFS

- RDF là mô hình dữ liệu đơn giản
- Lược đồ RDF - RDFS làm giàu thêm ngữ nghĩa bằng cách bổ xung các từ vựng để mô tả
 - Các lớp và các lớp con
 - Các thuộc tính và thuộc tính con
 - Kiểu thuộc tính
- RDFS cho phép mô tả các ontologies đơn giản
- Bổ xung thêm các yếu tố hướng đối tượng
 - Nhưng với cách tiếp cận theo hướng Lô-gic, sử dụng ngữ nghĩa với giả thuyết không gian mở.

Bộ từ vựng RDFS

RDFS đưa ra các từ khóa sau, đồng thời quy ước ý nghĩa tương thích với mô hình dữ liệu RDF

- Từ khóa cho lớp
 - [rdfs:Class](#)
 - [rdfs:subClassOf](#)
- Từ khóa cho thuộc tính
 - [rdfs:domain](#)
 - [rdfs:range](#)
 - [rdfs:subPropertyOf](#)
- Các lớp đặc biệt
 - [rdfs:Resource](#)
 - [rdfs:Literal](#)
 - [rdfs:Datatype](#)
- Từ khóa cho cấu trúc
 - [rdfs:member](#)
 - [rdfs:Container](#)
 - [rdfs:ContainerMembershipProperty](#)
- Các thuộc tính đặc biệt
 - [rdfs:comment](#)
 - [rdfs:seeAlso](#)
 - [rdfs:isDefinedBy](#)
 - [rdfs:label](#)

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

Lớp và các phần tử

- Chúng ta phân biệt giữa
 - Các phần tử (còn được gọi là các đối tượng, các thực thể, các tài nguyên) trong lĩnh vực
 - Tập hợp các phần tử có những thuộc tính giống nhau được gọi là lớp: Giảng viên, sinh viên, khóa học, v.v..
- Các tài nguyên thuộc 1 lớp được coi như phần tử của lớp đó (mô hình tập hợp).
- Mỗi quan hệ giữa phần tử và lớp của nó trong RDF được thiết lập bằng thuộc tính `rdf:type`
- Lưu ý: Có phần giống Lớp và đối tượng trong ngôn ngữ HDT nhưng lớp RDF giữ vai trò như tập hợp.

Lớp như công cụ quản lý thông tin

- Lớp cho phép chúng ta thiết lập giới hạn cho các câu trong RDF
 - Như trong ngôn ngữ lập trình
 - Ví dụ, $A \% B$ có ý nghĩa đúng đắn với A, B là các số nguyên, nhưng vô nghĩa với số thực.
 - Ngăn chặn các câu vô nghĩa dựa trên phát hiện mâu thuẫn.
 - *Lô-gic tính toán* được dạy bởi *cây bàng*: Đây là một câu vô nghĩa, chúng ta muốn học phần phải được dạy bởi Giảng viên \Rightarrow Giới hạn phạm vi giá trị của "*Được dạy bởi*" (giới hạn khoảng - range).
 - Phòng D9-101 được dạy bởi NVA: Chỉ có học phần mới là chủ thể của *được dạy bởi* - giới hạn các tài nguyên có thể được áp dụng thuộc tính (giới hạn miền, domain).
- Có thể suy diễn kiểu của 1 tài nguyên dựa vào cách nó được sử dụng - tương tự suy diễn kiểu trong ngôn ngữ lập trình.

Phân cấp lớp

- Các lớp được liên kết bởi quan hệ subClassOf có thể tạo thành 1 cấu trúc phân cấp
 - A là lớp con của B khi và chỉ khi tất cả các phần tử của A đồng thời là phần tử của B
 - Chúng ta còn nói rằng B là lớp cha của lớp A
- Quan hệ subClassOf giữa các lớp không tạo thành 1 cấu trúc cây
 - Một lớp có thể là lớp con của nhiều lớp khác nhau.
 - Có thể hình thành chu trình
 - A rdfs:subClassOf B. B rdfs:subClassOf A.
- Trong lô-gic:
 - $\text{subClassOf}(p, q) \Leftrightarrow p(x) \Rightarrow q(x)$
 - $\text{subClassOf}(p, q) \wedge p(x) \Rightarrow q(x)$

Miền & phạm vi

- Các ràng buộc miền & phạm vi cho phép giới hạn các tài nguyên có thể được sử dụng như chủ thể và giá trị của thuộc tính
 - Để hạn chế các câu vô nghĩa \Leftrightarrow Đảm bảo tính đúng đắn của dữ liệu
- Miền / chủ thể: Chỉ có học phần mới có thể *được dạy bởi*
 - $\text{domain}(\text{isTaughtBy}, \text{course})$
- Phạm vi / giá trị: Giá trị của thuộc tính *được dạy bởi* phải là giảng viên
 - $\text{range}(\text{isTaughtBy}, \text{Teacher})$
- Ngữ nghĩa trong lô-gic:
 - $\text{domain}(\text{pred}, \text{domclass}) \wedge \text{pred}(\text{subj}, \text{obj}) \Rightarrow \text{domclass}(\text{subj})$
 - $\text{range}(\text{pred}, \text{ranclass}) \wedge \text{pred}(\text{subj}, \text{obj}) \Rightarrow \text{ranclass}(\text{obj})$

Phân cấp thuộc tính

- Quan hệ phân cấp cho các thuộc tính được mô tả bằng từ khóa `rdfs:subPropertyOf`
 - Ví dụ, được dạy bởi là thuộc tính con của thuộc tính được phụ trách bởi
 - Nếu khóa C được dạy bởi A, thì C được phụ trách bởi A
- Chiều ngược lại không luôn luôn đúng
 - Nghĩa là: Nếu C được phụ trách bởi A, thì A có thể là giảng viên của khóa C, nhưng A cũng có thể là trợ giảng và không giảng dạy khóa C.
- Ngữ nghĩa lô-gic
 - $\text{subPropertyOf}(p, q) \wedge p(\text{subj}, \text{obj}) \Rightarrow q(\text{sub}, \text{obj})$
 - $\text{subPropertyOf}(\text{isTaughtBy}, \text{isTakenBy}) \wedge \text{isTaughtBy}(p, q) \Rightarrow \text{isTakenBy}(p, q)$
 - $\text{subPropertyOf}(\text{mother}, \text{parent}) \wedge \text{mother}(p, q) \Rightarrow \text{parent}(p, q)$

Biểu diễn lược đồ RDF bằng RDF

- Mô hình RDFS cũng có thể được biểu diễn bằng các bộ-3: tài nguyên-thuộc tính-giá trị.
- Để khai báo Giảng Viên là lớp con của lớp cán bộ
 - Chúng ta khai báo các lớp Giảng Viên, lớp Cán bộ, và thiết lập quan hệ subClassOf
 - `:lecturer rdf:type rdfs:Class.`
`:staffMember rdf:type rdfs:Class.`
`:lecturer rdfs:subClassOf :staffMember.`

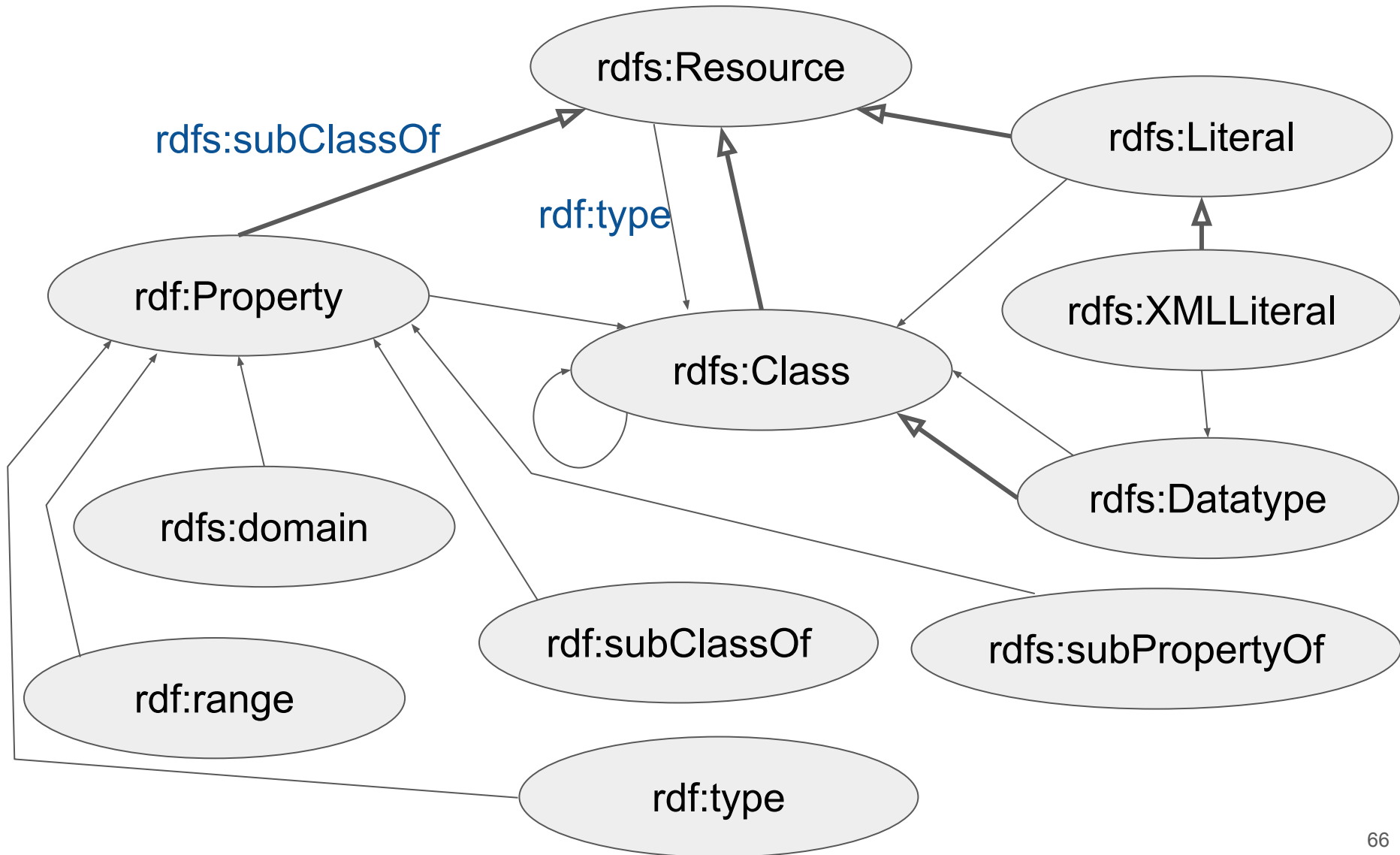
Các thành phần nền tảng

- Các lớp nền tảng
 - **rdfs:Resource** - Lớp của tất cả các tài nguyên
 - **rdfs:Class** - Lớp của tất cả các lớp
 - **rdfs:Literal** - Lớp của tất cả các hằng giá trị (chuỗi)
 - **rdf:Property** - Lớp của tất cả các thuộc tính
 - **rdf:Statement** - Lớp của tất cả các câu thành phần
- Các thuộc tính nền tảng
 - **rdf:type** - Xác định lớp của tài nguyên.
 - **rdfs:subClassOf** - Quan hệ lớp con của.
 - **rdfs:subPropertyOf** - Quan hệ thuộc tính con của.
 - **rdfs:domain** - Giới hạn miền của thuộc tính
 - **rdfs:range** - Giới hạn khoảng giá trị của thuộc tính

Các thành phần nền tảng: Các thuộc tính

- **rdfs:subClassOf** và **rdfs:subPropertyOf** được quy ước có tính chất bắc cầu
- **rdfs:Class** là lớp con của **rdfs:Resource**
 - Bởi vì tất cả các lớp đều là các tài nguyên
- **rdfs:Resource** là 1 phần tử của **rdfs:Class**
 - **rdfs:Resource** là lớp của tất cả các tài nguyên
- Tất cả các lớp đều là phần tử của **rdfs:Class**

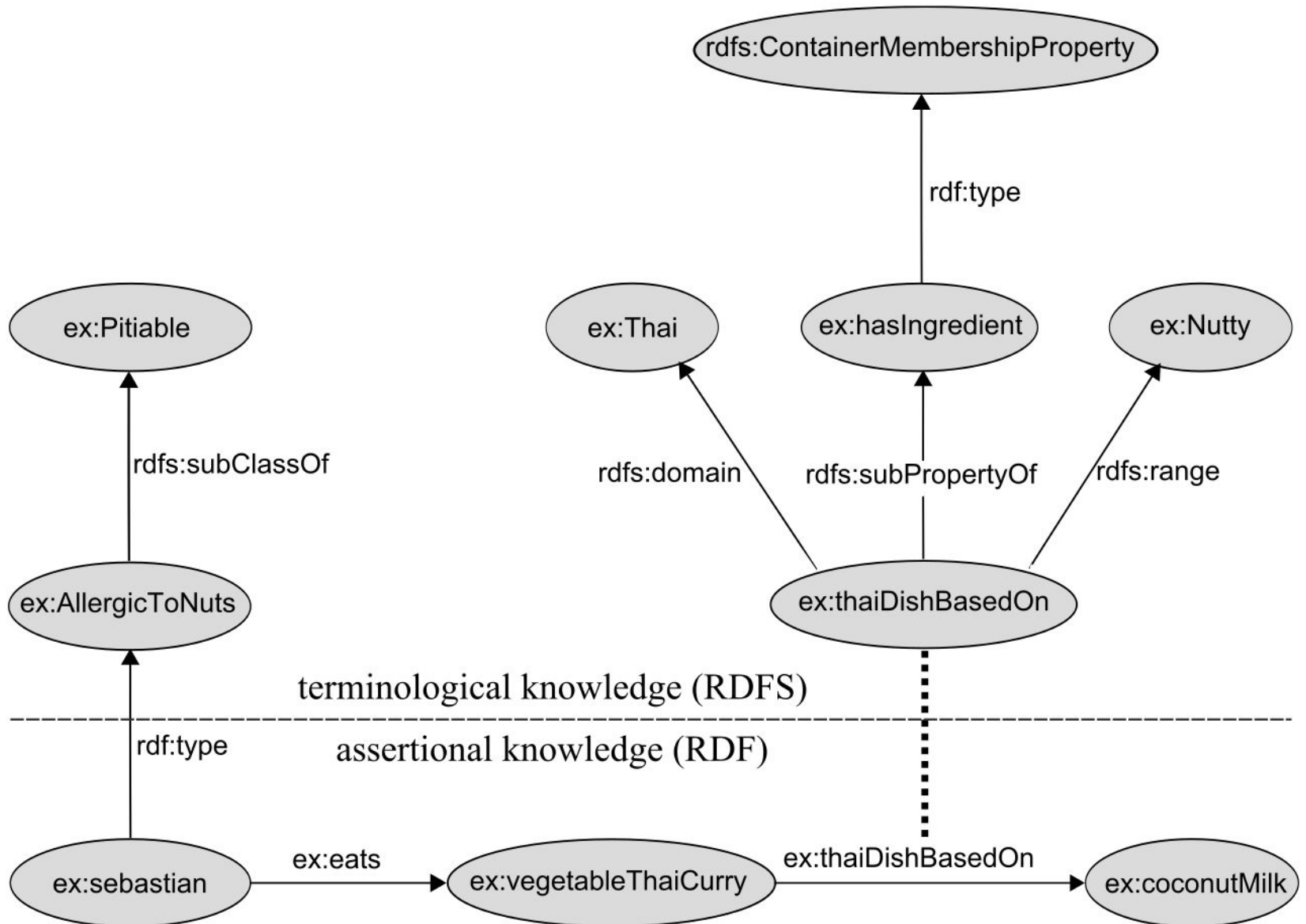
Các thành phần nền tảng: Biểu diễn đồ thị



Các thuộc tính hỗ trợ

- **rdfs:seeAlso** - Xem thêm, gợi ý tài nguyên liên quan.
- **rdfs:isDefinedBy** - Được định nghĩa bởi, thuộc tính con của **rdfs:seeAlso**.
- **rdfs:comment** - Ghi chú, thường nội dung văn bản mô tả, giới thiệu về 1 tài nguyên.
- **rdfs:label** - Tên thân thiện với người, được đặt cho tài nguyên.

Ví dụ 3.8. Dữ liệu và lược đồ



Không gian tên RDF và RDFS

- Các không gian tên RDF và RDFS
 - rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
 - rdfs: <http://www.w3.org/2000/01/rdf-schema#>
- Các từ khóa từ các không gian tên này được sử dụng để định nghĩa từ khóa riêng cho từng lĩnh vực và từng trường hợp sử dụng

Ví dụ 3.9. không gian tên

@prefix dc: <http://purl.org/dc/elements/1.1/> .

@prefix owl: <http://www.w3.org/2002/07/owl#>

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

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

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

a owl:Ontology ;

dc:title "The RDF Vocabulary (RDF)" ;

dc:description "This is the RDF Schema for the RDF vocabulary defined in the
RDF namespace." .

rdf:type a rdf:Property ;

rdfs:isDefinedBy <http://www.w3.org/1999/02/22-rdf-syntax-ns#> ;

rdfs:label "type" ;

rdfs:comment "The subject is an instance of a class." ;

rdfs:range rdfs:Class ;

rdfs:domain rdfs:Resource .

Ví dụ 3.9. không gian tên₍₂₎

```
rdf:Statement a rdfs:Class ;  
    rdfs:subClassOf rdfs:Resource ;  
    rdfs:comment "The class of RDF statements." .
```

```
rdf:subject a rdf:Property ;  
    rdfs:domain rdf:Statement ;  
    rdfs:range rdfs:Resource .
```

```
rdf:predicate a rdf:Property ;  
    rdfs:domain rdf:Statement ;  
    rdfs:range rdfs:Resource .
```

RDF(S) vs. mô hình HĐT

- Trong các mô hình LT HĐT, 1 lớp xác định các thuộc tính có thể áp dụng lên các đối tượng thuộc lớp
 - Bổ xung thêm thuộc tính có nghĩa là thay đổi lớp
- Trong RDF(S), các thuộc tính được định nghĩa toàn cục và không được đóng gói lớp như trong LT HĐT
 - Chúng ta có thể định nghĩa các thuộc tính mới, có thể thay đổi hoặc không thay đổi lớp
 - Các thuộc tính có thể có thuộc tính
 - Nhưng không thể thu hẹp miền & khoảng của thuộc tính trong phạm vi các lớp con

Ví dụ 3.10. Miền & Khoảng và các lớp con

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

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

@prefix bio: <http://example.com/biology#> .

bio:Animal a rdfs:Class.

bio:offspring a rdf:Property;

 rdfs:domain bio:Animal;

 rdfs:range bio:Animal.

bio:Cat rdfs:subClassOf bio:Animal.

bio:Dog rdfs:subClassOf bio:Animal.

bio:fido a bio:Dog.

bio:milo a bio:Cat;

 bio:offspring bio:fido.

Lược đồ và dữ liệu

Nên lưu lược đồ tách biệt với dữ liệu trong các tệp khác nhau

```
# A simple Biology ontology - animal.ttl
```

```
@prefix rdf:
<http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:
<http://www.w3.org/2000/01/rdf-schema#> .
@prefix bio: <http://example.com/biology#> .
```

```
bio:Animal a rdfs:Class.
bio:offspring a rdf:Property;
  rdfs:domain bio:Animal;
  rdfs:range bio:Animal.
bio:Cat rdfs:subClassOf bio:Animal.
bio:Dog rdfs:subClassOf bio:Animal.
```

```
# Some biological data - data.ttl
```

```
@prefix rdf:
<http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:
<http://www.w3.org/2000/01/rdf-schema#> .
@prefix bio: <http://example.com/biology#> .
```

```
# fido's a dog!
bio:fido a bio:Dog.
```

```
# milo's cat & has offspring fido
bio:milo a bio:Cat;
  bio:offspring bio:fido.
```

Apache Jena

- Apache Jena là 1 dự án mã nguồn mở, được phát triển bằng Java, là 1 công nghệ WebNN
- Jena có 1 tập công cụ dòng lệnh, ví dụ:
 - riot có thể chuyển đổi giữa các định dạng và thực hiện các suy diễn rdfs đơn giản.

```
riot --rdfs animal.ttl --formatted=ttl data.ttl
```

```
@prefix :      <http://finin.org/example/#> .
```

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

```
@prefix bio:    <http://example.com/biology#> .
```

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

```
:milo  a          bio:Animal , bio:Cat ;
```

```
      bio:offspring :fido .
```

```
:fido  a          bio:Animal , bio:Dog .
```

Nội dung

3.1. Mô hình dữ liệu RDF

3.2. Cú pháp RDF

3.3. Ontology đơn giản với RDFS

3.4. Biểu diễn các cấu trúc đặc biệt



Câu thành phần/Reification

Câu thành phần

- Câu thành phần - Reification
- Có những trường hợp, chúng ta muốn viết các câu về những câu khác
 - Ví dụ, đưa ra giả thuyết, mô tả nguồn của thông tin
 - John tin rằng Mary yêu John .
 - Theo dự báo của trung tâm khí tượng thủy văn, ngày mai trời nắng...
- Cần biểu diễn câu như 1 tài nguyên
 - RDF cho phép mô tả như vậy qua cơ chế Reification

Ví dụ 3.11. Câu thành phần

:123 :name "Nguyen Van A" .

Được tái cấu trúc lại như một tài nguyên như sau:

```
[a rdf:Statement;  
  rdf:subject :123;  
  rdf:predicate :name;  
  rdf:object "Nguyen Van A"] .
```

Ví dụ 3.11. Câu thành phần₍₂₎

Alice tin rằng Bob yêu Carol với độ chắc chắn = 0.5

@prefix ep: <<http://example.com/epistemology>>

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

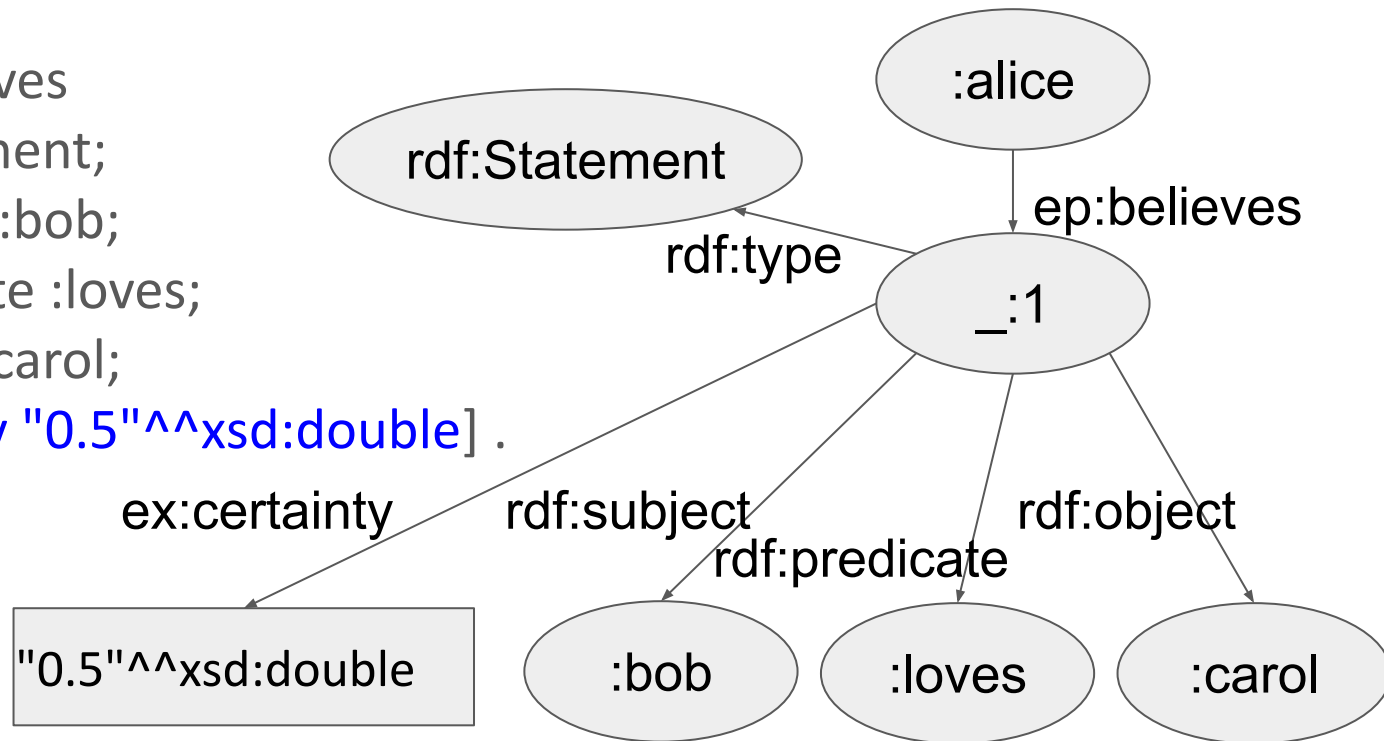
@prefix xsd: <http://www.w3.org/2001/XMLSchema>

@prefix : <#>.

:alice ep:believes

[a rdf:Statement;
rdf:subject :bob;
rdf:predicate :loves;
rdf:object :carol;

ex:certainty "0.5"^^xsd:double] .



Biểu diễn câu Mary tin rằng Bob yêu Carol với độ chắc chắn = 0.8

Cấu trúc nhóm/ Nhóm tài nguyên

Các thành phần của cấu trúc nhóm

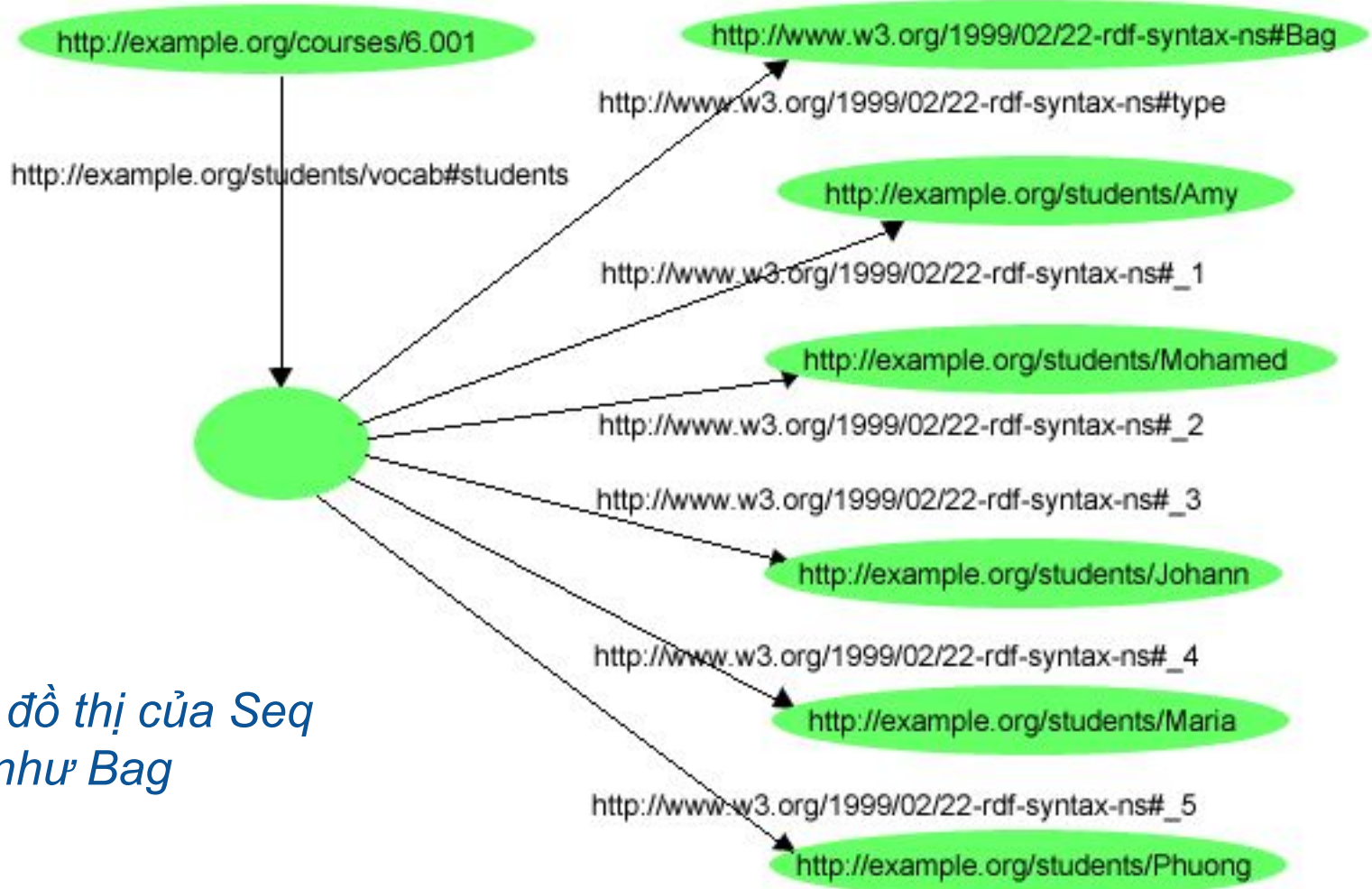
- RDF có các từ khóa để mô tả các nhóm và viết các câu về nhóm
- Ví dụ, chúng ta muốn diễn đạt thông tin 1 giảng viên dạy nhiều khóa học
- Các thành phần của nhóm được ký hiệu `rdf:_1`, `rdf:_2`, v.v..
 - Tương đương với `rdf:li`
- `:john :teaches [a rdf:Bag; rdf:li :cmssc201, :cmssc202, :cmssc345] .`

Ba loại cấu trúc nhóm

- **rdf:Bag** - một cấu trúc nhóm không thứ tự, cho phép phần tử xuất hiện nhiều lần
 - Ví dụ, các sinh viên trong lớp, các tệp trong một thư mục
- **rdf:Seq** - một cấu trúc nhóm có thứ tự, một phần tử có thể xuất hiện nhiều lần
 - Ví dụ, các mô-đun của một khóa đào tạo, danh sách sinh viên đã sắp xếp theo tên
- **rdf:Alt** - một tập các phần tử tương đương
 - ví dụ, một trang web và các ảnh của nó, các bản in của cùng một cuốn sách

Ví dụ 3.12. Bag

Mô tả danh sách sinh viên của 1 khóa học



*Biểu diễn đồ thị của Seq
tương tự như Bag*

Ví dụ 3.12. Bag - định dạng turtle

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

@prefix s: <http://example.org/students/vocab#>.

<http://example.org/courses/6.001>

s:students [

a rdf:Bag;

rdf:_1 <http://example.org/students/Amy>;

rdf:_2 <http://example.org/students/Mohamed>;

rdf:_3 <http://example.org/students/Johann>;

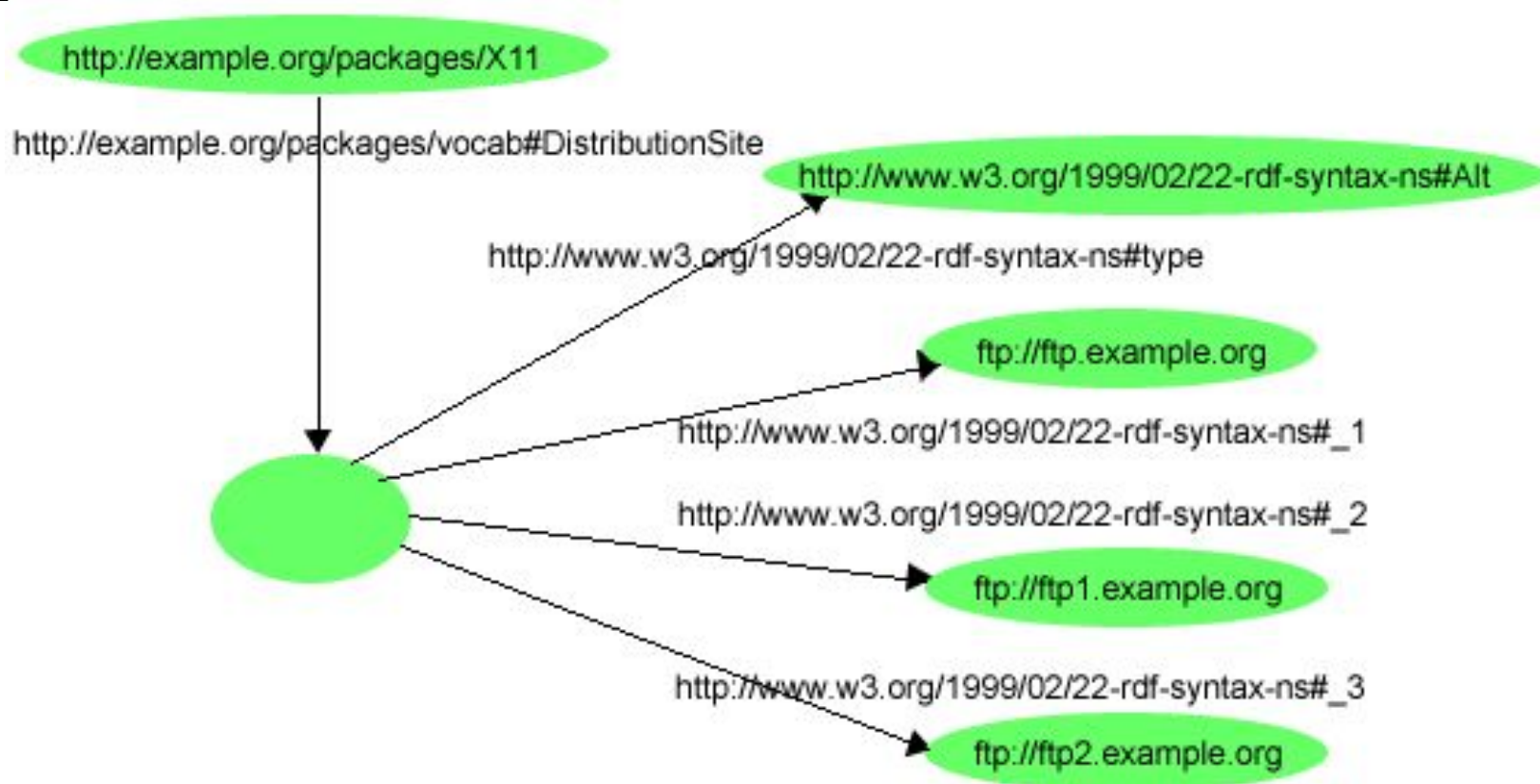
rdf:_4 <http://example.org/students/Maria>;

rdf:_5 <http://example.org/students/Phuong>.

].

Cú pháp của Seq và Alt tương tự như Bag, chỉ thay Bag bằng Seq / Alt

Ví dụ 3.13. Alt



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

@prefix s: <http://example.org/packages/vocab#>.

<http://example.org/packages/X11>

s:DistributionSite [a rdf:Alt;

 rdf:_1 <ftp://ftp.example.org>;

 rdf:_2 <ftp://ftp1.example.org>;

 rdf:_3 <ftp://ftp2.example.org>.]

Bag và Seq không bao giờ đầy

- Các cấu trúc Bag và Seq luôn mở:
 - Không thể "đóng" các nhóm và tuyên bố: "Đây là tất cả các phần tử, ngoài ra không có phần tử nào khác nữa"
 - RDF là mô hình dữ liệu đồ thị, không có cách nào để loại trừ khả năng một đồ thị khác có thể bổ xung thêm các phần tử mới.
- Danh sách - List, là các cấu trúc nhóm đóng, chỉ có các phần tử được liệt kê
- Được mô tả theo khuôn mẫu danh sách liên kết:
 - `rdf:List`, `rdf:first`, `rdf:rest`, `rdf:nil`

Giả thuyết không gian mở vs. đóng

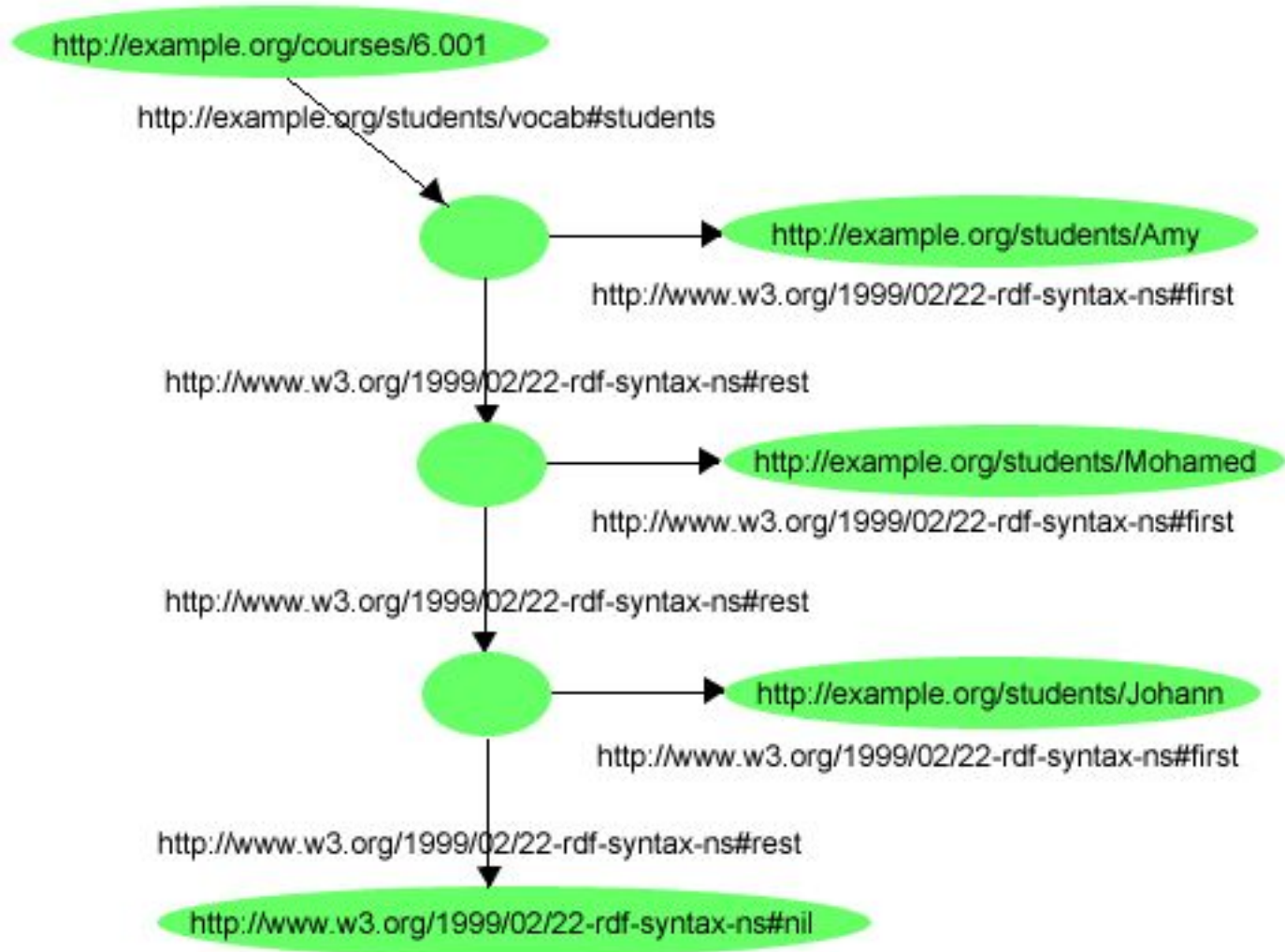
- Các hệ thống suy diễn phân biệt giữa các giả thuyết không gian mở và không gian đóng
 - Không gian mở - OWA, Open World Assumption, coi cơ sở tri thức luôn ở trạng thái chưa đầy đủ.
 - Nếu không thể chứng minh một khẳng định là đúng hoặc sai, thì coi như không biết giá trị chân lý của khẳng định đó.
 - Không gian đóng - CWS, Close World Assumption, coi cơ sở tri thức đã ở trạng thái hoàn thiện.
 - Tất cả những gì không thể chứng minh là đúng được coi là sai.
- Mô hình ngữ nghĩa mặc định cho Web ngữ nghĩa được dựa trên OWA

Ngữ nghĩa không gian mở vs đóng

- Lô-gic cổ điển sử dụng ngữ nghĩa không gian mở
 - Nếu không thể chứng minh P hoặc NP thì không phủ nhận
- HQT CSDL thường áp dụng CWS
 - Nếu không tìm thấy đơn hàng nào nghĩa là chưa mua hàng
- Toán tử không thể chứng minh (not hoặc \+) trong Prolog hỗ trợ CWS
 - `fly(x) :- bird(x), \+flightless(x) .`
 - `flightless(x) :- penguin(x); ostrich(x); emu(x) .`
- Một số hệ thống cho phép chúng ta chỉ ra khẳng định nào chúng ta đã có đủ thông tin, khẳng định nào chưa có đủ thông tin.

Ví dụ 3.14. Danh sách RDF

Một danh sách của 3 sinh viên trong lớp



Ví dụ 3.14. Danh sách RDF

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

@prefix s: <http://example.org/students/vocab#>.

<http://example.org/courses/6.001>

s:students

[a rdf:List;

 rdf:first <http://example.org/students/Amy>;

 rdf:rest [a rdf:list

 rdf:first <http://example.org/students/Mohamed>;

 rdf:rest [a rdf:List;

 rdf:first <http://example.org/students/Johann>;

 rdf:rest rdf:nil]]] .

Ví dụ 3.14. Danh sách RDF₍₂₎

Turtle có một cấu trúc riêng để biểu diễn danh sách:

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

@prefix s: <http://example.org/students/vocab#>.

<http://example.org/courses/6.001>

s:students (

<http://example.org/students/Amy>

<http://example.org/students/Mohamed>

<http://example.org/students/Johann>

).

Thử nghiệm

Lược đồ schema.ttl

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

@prefix : <http://semweb.edu.vn/demo#> .

:A rdfs:subClassOf :B.

:p rdfs:subPropertyOf :q.

Dữ liệu data.ttl

@prefix : <http://semweb.edu.vn/demo#> .

:a a :A.

:b a :B.

:a :p :c.

Thử: *Tìm những khẳng định mới có thể suy ra được trên cơ sở lược đồ.*

RDFS: riot --rdfs schema.ttl data.ttl

Bài tập 3.1

Xây dựng mô hình rdf cho nội dung sau:

Nhóm 1 gồm các thành viên An, Nam và Linh đã chọn đề tài “xây dựng mô hình rdf cho lĩnh vực Điện Ảnh”.

Sử dụng: a) Turtle b) Đồ thị RDF c) RDF/XML

Bài tập 3.2

Xây dựng mô hình rdf cho các nội dung sau:

Sông Mê-Kông có chiều dài 4350 Km, bắt nguồn từ Thanh Hải/Trung Quốc và đổ ra biển ở Đồng bằng sông cửu Long/Việt Nam.

Sử dụng: a) Turtle b) Đồ thị RDF c) RDF/XML

<https://github.com/bangoc/semweb>

