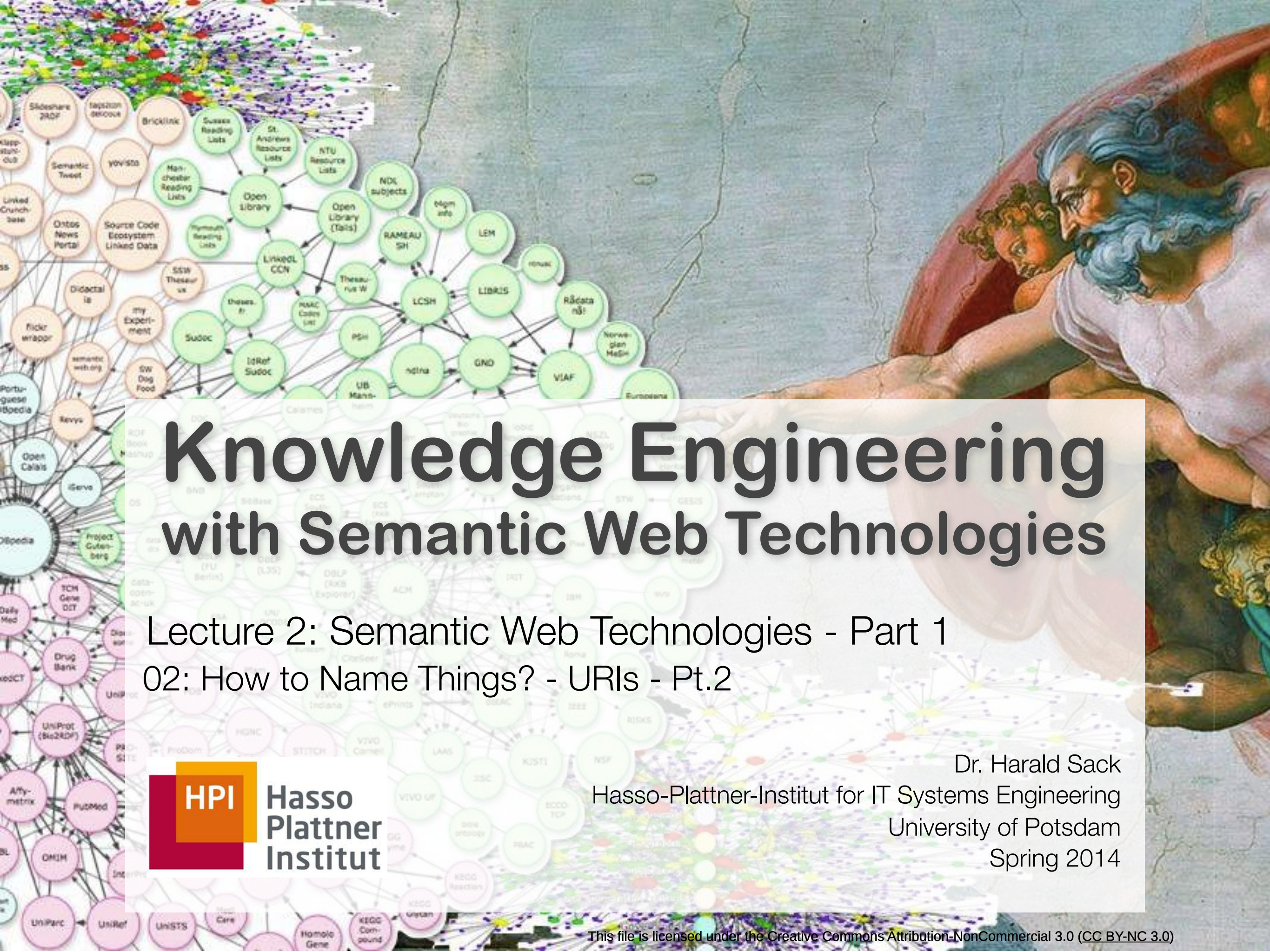




Knowledge Engineering with Semantic Web Technologies

Lecture 2: Semantic Web Technologies - Part 1
02: How to Name Things? - URIs - Pt.2



Dr. Harald Sack
Hasso-Plattner-Institut for IT Systems Engineering
University of Potsdam
Spring 2014



Lecture 2: Semantic Web Technologies - Part 1

Open HPI - Course: Knowledge Engineering with Semantic Web Technologies



02 How to Name Things? - URIs Pt.2

Open HPI - Course: Knowledge Engineering with Semantic Web Technologies
Lecture 2: Semantic Web Technologies Part I

Use URIs to name things on the Web

Resource: Eiffel Tower

URI

<http://www.tour-eiffel.fr/index.html>

identifies

(stands for)



5 ■ URI combines

□ **Address (Locator)**

- **Uniform Resource Locator** (URL, RFC 1738)
- denotes, **where** a resource can be found in the Web by stating its primary access mechanism
- might change during life cycle

5 ■ URI combines

□ **Address (Locator)**

- **Uniform Resource Locator** (URL, RFC 1738)
- denotes, **where** a resource can be found in the Web by stating its primary access mechanism
- might change during life cycle

□ **Identity (Name)**

- **Uniform Resource Name** (URN, RFC 2141)
- persistent identifier for a Web resource
- remains unchanged during life cycle

URI - Generic Syntax

- US-ASCII Encoding
- Percent Encoding for reserved characters, or characters that do not exist in US-ASCII encoding existieren

pct-encoded = "%" HEXDIG HEXDIG

- Reserved characters with special function

reserved = gen-delims / sub-delims

gen-delims = ":" / "/" / "?" / "#" / "[" / "]" / "@"

sub-delims = "!" / "\$" / "&" / "'" / "(" / ")" / "*" / "+" / "," / ";" / "="

- Permitted characters

unreserved = ALPHA / DIGIT / "-" / "." / "_" / "~"

- Extension to Universal Character Code (Unicode/ISO 10646)

- Internationalized Resource Identifier (IRI, RFC 3987)

e.g.:

کق#فغ=عظ;طض=صش؟سز/رذ/دخ/حج.ثت.با http://

URI - Generic Syntax

URI = schema"://"[userinfo"@"]host[:port][path]["?"query]["#"fragment]

URI - Generic Syntax

URI = schema"://[userinfo"@"]host[:port][path][?"query]["#"fragment]

- **schema:** e.g. http, ftp, mailto,...

URI - Generic Syntax

URI = schema"://"[userinfo"@"]host[:port][path]["?"query]["#"fragment]

- **schema**: e.g. http, ftp, mailto,...
- **userinfo**: e.g. username:password

URI - Generic Syntax

URI = schema:"://"[userinfo"@"]host[:port][path]["?"query]["#"fragment]

- **schema**: e.g. http, ftp, mailto,...
- **userinfo**: e.g. username:password
- **host**: Domain-Name, IPv4/IPv6 Address

URI - Generic Syntax

URI = schema"://"[userinfo"@"]host[:port][path]["?"query]["#"fragment]

- **schema**: e.g. http, ftp, mailto,...
- **userinfo**: e.g. username:password
- **host**: Domain-Name, IPv4/IPv6 Address
- **port**: e.g. :80 for standard http port

URI - Generic Syntax

URI = schema"://"[userinfo"@"]host[:port][path]["?"query]["#"fragment]

- **schema**: e.g. http, ftp, mailto,...
- **userinfo**: e.g. username:password
- **host**: Domain-Name, IPv4/IPv6 Address
- **port**: e.g. :80 for standard http port
- **path**: path in file system of WWW server

URI - Generic Syntax

URI = schema"://" [userinfo"@"] host[:port] [path] ["?" query] ["#" fragment]

- **schema**: e.g. http, ftp, mailto,...
- **userinfo**: e.g. username:password
- **host**: Domain-Name, IPv4/IPv6 Address
- **port**: e.g. :80 for standard http port
- **path**: path in file system of WWW server
- **query**: parameters to be passed over to applications

URI - Generic Syntax

URI = schema"://" [userinfo"@"] host[:port] [path] ["?" query] ["#" fragment]

- **schema**: e.g. http, ftp, mailto,...
- **userinfo**: e.g. username:password
- **host**: Domain-Name, IPv4/IPv6 Address
- **port**: e.g. :80 for standard http port
- **path**: path in file system of WWW server
- **query**: parameters to be passed over to applications
- **fragment**: determines a specific fragment of a document

What if a URI for a resource does not exist?

- Define a URI by yourself:
 - **avoid overlaps** → use your own website
 - enable **documentation at the same place** → Content Negotiation
- Use **separate URIs** for the resource (*Designatum*) and its documentation (*Designator*) via
 - Content Negotiation or
 - URI references (i.e. via "#" fragment identifier)

9

Let's try an example:

- I want to have **information** about the Eiffel Tower from DBpedia

http://dbpedia.org/resource/Eiffel_Tower

URI represents the *Designatum*

Content Negotiation

9

Let's try an example:

- I want to have **information** about the Eiffel Tower from DBpedia

http://dbpedia.org/resource/Eiffel_Tower

1

HTTP GET request

Accept Header: text/html

URI represents the *Designatum*

Content Negotiation

9

Let's try an example:

- I want to have **information** about the Eiffel Tower from DBpedia

http://dbpedia.org/resource/Eiffel_Tower

1

HTTP GET request

Accept Header: text/html

URI represents the *Designatum*

HTTP/1.1 303 See Other

2

http://dbpedia.org/page/Eiffel_Tower

Content Negotiation

9

Let's try an example:

- I want to have **information** about the Eiffel Tower from DBpedia

1

http://dbpedia.org/resource/Eiffel_Tower

HTTP GET request

Accept Header: text/html

URI represents the *Designatum*

HTTP/1.1 303 See Other

2

http://dbpedia.org/page/Eiffel_Tower

URI represents the *Designator*

Content Negotiation

9

Let's try an example:

- I want to have **information** about the Eiffel Tower from DBpedia

1

http://dbpedia.org/resource/Eiffel_Tower

HTTP GET request

Accept Header: text/html

URI represents the *Designatum*

HTTP/1.1 303 See Other

2

http://dbpedia.org/page/Eiffel_Tower

URI represents the *Designator*

3

HTTP GET request

Accept Header: text/html

Content Negotiation

9

Let's try an example:

- I want to have **information** about the Eiffel Tower from DBpedia

1

HTTP GET request
Accept Header: text/html

http://dbpedia.org/resource/Eiffel_Tower

URI represents the *Designatum*

HTTP/1.1 303 See Other

2

http://dbpedia.org/page/Eiffel_Tower

URI represents the *Designator*

3

HTTP GET request
Accept Header: text/html

4



HTML Document

Content Negotiation

Let's try an example:

- I want to have **machine readable information** about the Eiffel Tower from DBpedia

http://dbpedia.org/resource/Eiffel_Tower

URI represents the *Designatum*

Content Negotiation

Let's try an example:

- I want to have **machine readable information** about the Eiffel Tower from DBpedia

http://dbpedia.org/resource/Eiffel_Tower

1

HTTP GET request

Accept Header: `application/rdf+xml`

URI represents the *Designatum*

Content Negotiation

Let's try an example:

- I want to have **machine readable information** about the Eiffel Tower from DBpedia

http://dbpedia.org/resource/Eiffel_Tower

1

HTTP GET request

Accept Header: `application/rdf+xml`

URI represents the *Designatum*

HTTP/1.1 303 See Other 2

http://dbpedia.org/page/Eiffel_Tower

Content Negotiation

Let's try an example:

- I want to have **machine readable information** about the Eiffel Tower from DBpedia

http://dbpedia.org/resource/Eiffel_Tower

1

HTTP GET request

Accept Header: `application/rdf+xml`

URI represents the *Designatum*

HTTP/1.1 303 See Other 2

http://dbpedia.org/page/Eiffel_Tower

URI represents the *Designator*

Content Negotiation

Let's try an example:

- I want to have **machine readable information** about the Eiffel Tower from DBpedia

http://dbpedia.org/resource/Eiffel_Tower

1

HTTP GET request

Accept Header: `application/rdf+xml`

URI represents the *Designatum*

HTTP/1.1 303 See Other 2

http://dbpedia.org/page/Eiffel_Tower

URI represents the *Designator*

3

HTTP GET request

Accept Header: `application/rdf+xml`

Content Negotiation

Let's try an example:

- I want to have **machine readable information** about the Eiffel Tower from DBpedia

http://dbpedia.org/resource/Eiffel_Tower

1

HTTP GET request

Accept Header: `application/rdf+xml`

URI represents the *Designatum*

HTTP/1.1 303 See Other 2

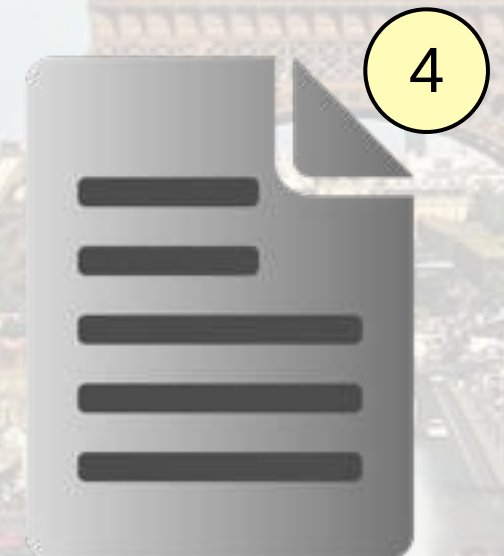
http://dbpedia.org/page/Eiffel_Tower

URI represents the *Designator*

3

HTTP GET request

Accept Header: `application/rdf+xml`



RDF Document

Fragment Identifier

11 Let's try another example:

- I want to have **machine readable information** myself

e.g.: http://hpi.uni-potsdam.de/harald_foaf.rdf

Designator



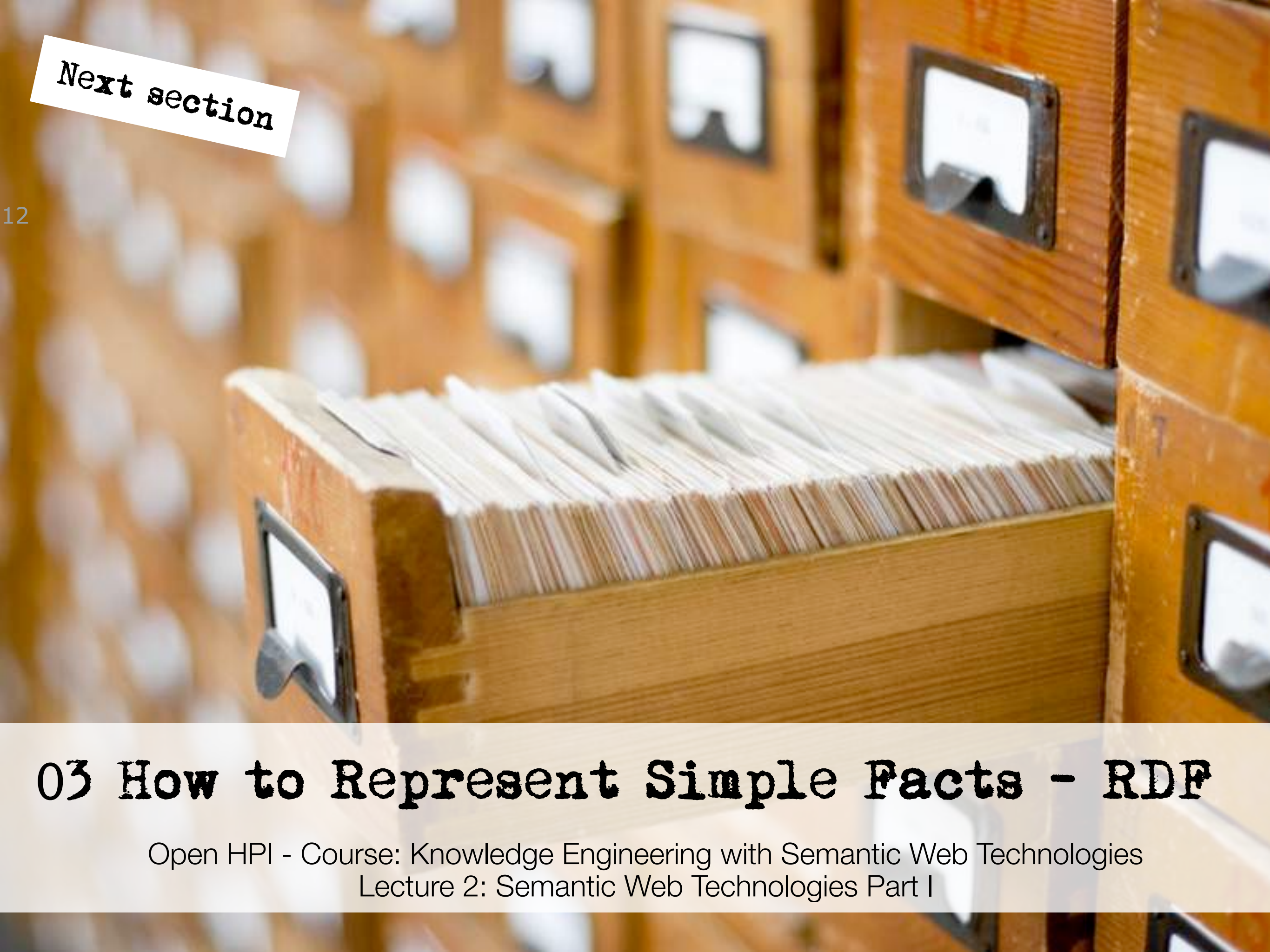
FOAF Document
with information
about me

http://hpi.uni-potsdam.de/harald_foaf.rdf#me

identifies me

Designatum





Next section

12

03 How to Represent Simple Facts - RDF

Open HPI - Course: Knowledge Engineering with Semantic Web Technologies
Lecture 2: Semantic Web Technologies Part I