

Web 2.0

Lecture 10: Annotations

doc. Ing. Tomáš Vitvar, Ph.D.

tomas@vitvar.com • @TomasVitvar • <http://vitvar.com>



Czech Technical University in Prague

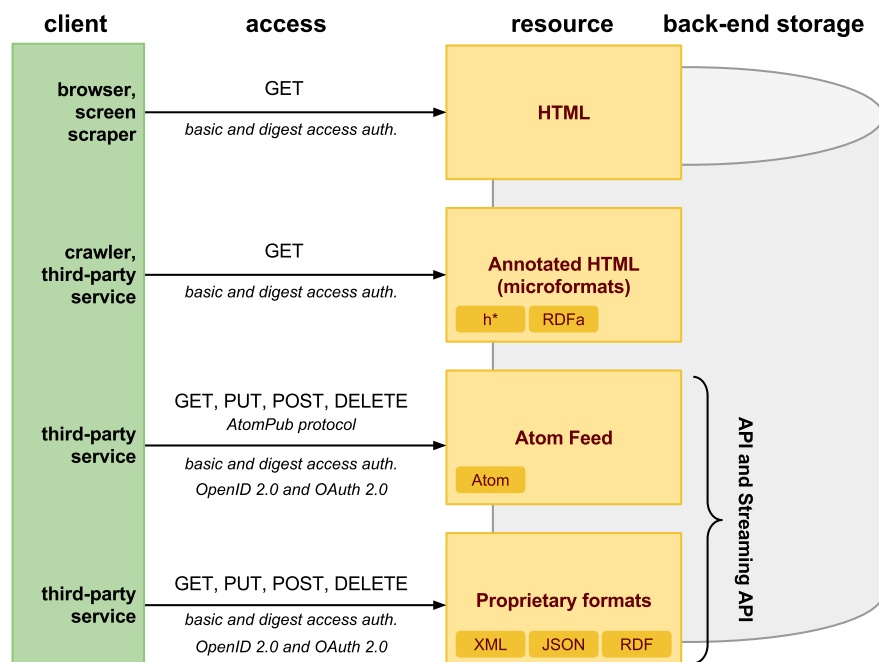
Faculty of Information Technologies • Software and Web Engineering • <http://vitvar.com/courses/w20>



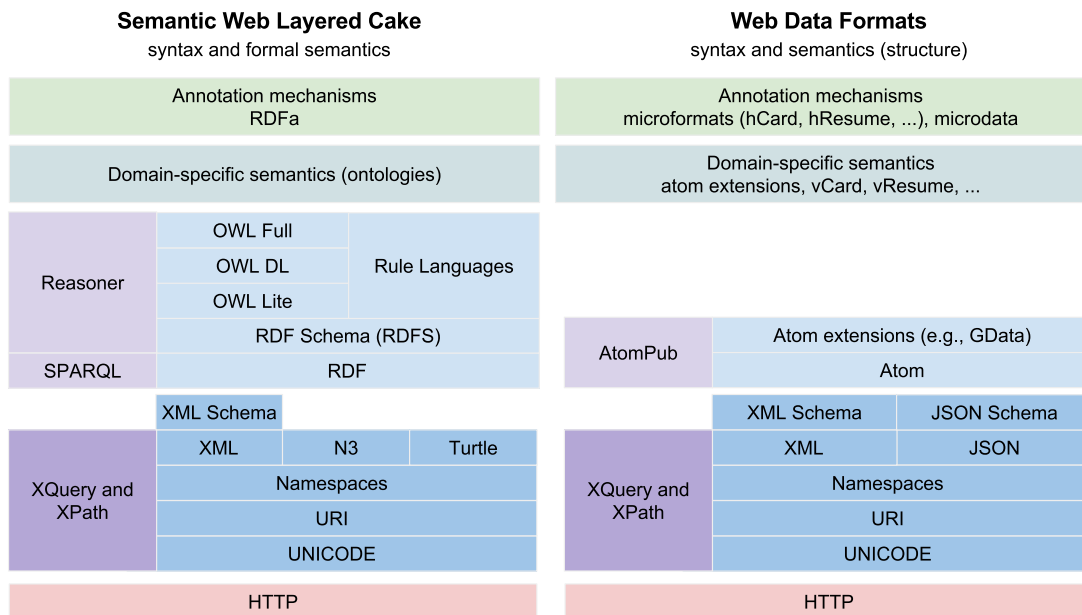
Evropský sociální fond
Praha & EU: Investujeme do vaší budoucnosti

Modified: Thu May 01 2014, 09:51:25
Humla v0.3

Data on the Web



Data Syntax, Structure and Semantics



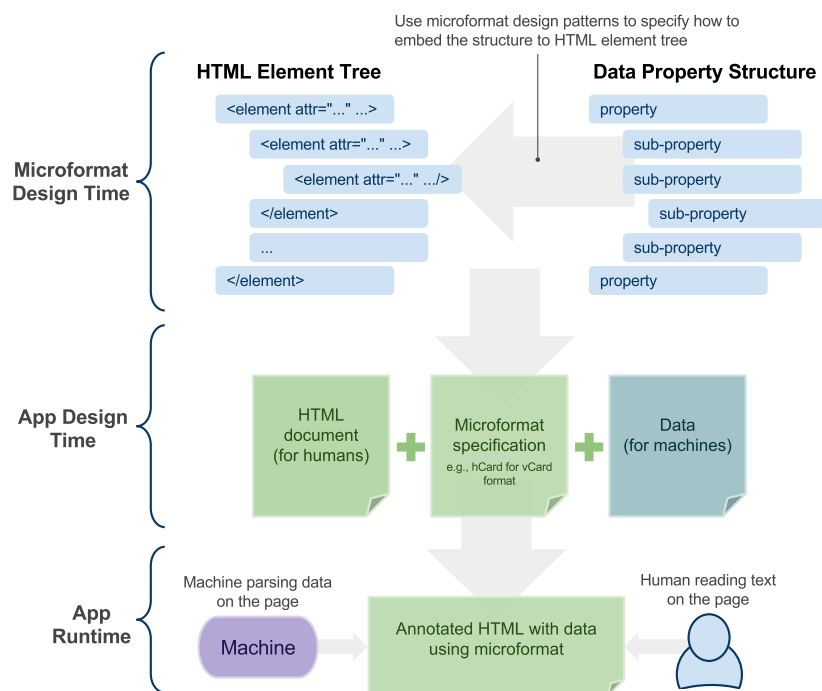
Overview

- **Microformats**
- Microdata
- RDF and RDFa
- OpenGraph Protocol

Microformats

- What is a microformat
 - How to embed data in HTML, XHTML, Atom, and XML
 - data: vCard, vResume, vRecipe
 - micorformat: hCard, hResume, hRecipe
 - Browsers display HTML, machines process data
 - Microformat vs. POSH format
 - POSH is same as microformat but data is not a standard format
- Difference to Atom feeds
 - Microformats require only a **single HTML document**
 - clients run GET to retrieve all data (human readable and machine readable)
 - No significant increase of the size of document
 - No requirements on data representation
 - can be in any representation
 - should be defined in a well-established format spec
 - a microformat spec needs to be defined for every data

Microformats Usage



Principles

- Design Patterns
 - How to embed data in HTML elements or elements' attributes
 - Applied for a particular microformat specification
- Follow semantics of (X)HTML elements
 - Use the most appropriate semantic HTML element [↗](#)
 - if not available, use `` or `<div>`
- XHTML Metadata Profiles (XMDP)
 - Definition of metadata of a microformat in (X)HTML page
 - Machine and human readable, not a Web standard
 - Uses **profile** attribute on `<head>` element
 - Is deprecated in HTML5
 - Is an analogy to a namespace but not really a namespace!
 - See XHTML Metadata Profiles [↗](#) specification

vCard Example

- Describes contact information
 - **N** – a structured representation of the name (person/organization)
 - **FN** – formatted name string
 - **ORG** – name of the organization and associated units
 - **TITLE** – job title, functional position
 - **LABEL** – Addressing label

Design Patterns Rules

- **class-design-pattern**
 - semantic meaning indicated on HTML content by **class** attribute
- **value-class-pattern**
 - embedding data structure when a property has subproperties (vCard fragment is **TEL;TYPE=WORK:+43 554 554 556**)
 - sometimes value needs to be split into multiple pieces as follows (note that dialing **+430554554556** is not valid)

Design Patterns Rules (cont.)

- **include-pattern**
 - to include a subset of data from one area of a page to the other area of the same page (same data to be reused by multiple microformats)
 - **cannot be used to include content from other URLs!**
 - Example, a verbose hCard on a page:
 - Reviews on the same page:
(parser replaces the whole **<a>** element including its content)

hCard Microformat Example

- hCard profile, options:
- Example specific rules
 - vCard properties that do not make sense for hCard
 - e.g., NAME, PROFILE, SOURCE, PRODID, VERSION
 - publishers should not use them, parses should ignore them
 - if `fn == org` (i.e, `class="fn org"`)
 - hCard is a contact for a company, organization or a place
 - **N** (person's name) property should not be used or be the empty string
 - if `fn != org` AND `fn` contains two words
 - `fn` is split into `given-name` and `last-name`
 - sub-properties of **N** property (by a whitespace or a comma)
 - see a complete specification in hCard Microformat Specification [🔗](#)

Known Issues

- Name conflicts and scalability
 - More microformats on a page may cause naming conflicts
 - no namespace support, **microformats do not scale**
 - functionality of tools may break when data formats change
- No formal semantics, no reasoning support
 - How important is it?
 - Semantics defined in XMDP profiles
 - no formal basis though machine processable
 - lack of compatibility with RDF/RDFa
 - See Microformats and RDF/RDFa compatibility [🔗](#) for details.

Uptake and some statistics

- Two billion pages annotated with hCard
 - Google Rich Snippets
 - Content indexing with microformats, microdata, RDFa
 - see *Google Rich Snippets* [↗](#)
 - 94% of the rich snippets data uses microformats
- [Pizza Pizzas Recipe : Alton Brown : Food Network](#)
[www.foodnetwork.com](#) > Recipes > Italian
★★★★★ 229 reviews - 24 hrs 45 mins
Food Network invites you to try this **Pizza Pizzas recipe** from Alton Brown.
- Firefox 3
 - Native API to parse and process microformats in JavaScript
 - see *Microformats support in Firefox 3* [↗](#)
 - Facebook
 - hCalendar and hCard for events
 - see *Microformats in Facebook* [↗](#)

Overview

- Microformats
- **Microdata**
- RDF and RDFa
- OpenGraph Protocol

Microdata

- Part of HTML5 specification
 - Google is the main driver (rich snippets support)
 - spec includes:
 - Microdata vocabularies
 - Microdata Global Attributes
 - see W3C working draft
- Idea similar to microformats, but
 - items (collection of properties) have ids (URIs)
 - Microdata vocabulary, a formal description of terms
 - <http://schema.org> is becoming a standard
 - e.g., Event, Organization, Person, Product, Review
 - Created and supported by Google, Microsoft, Yahoo!
 - have RDF representation too
 - data formats not directly based on formats such as vCard, vCalendar, they define its own "simple" vocabulary

Global Attributes

- Attributes on any HTML element
- **Itemscope**
 - identifies an element which descendants contain some properties
- **Itemtype**
 - pointer to a vocabulary that describes the item and its properties
<http://www.data-vocabulary.org/Person/>
- **Itemid**
 - global identifier of the item (URI)
 - such as a book's ISBN in urn schema, `urn:isbn:0-330-34032-8`
- **Itemprop**
 - a term from the vocabulary which value is in the element's content
- **Itemref**
 - a reference to other item within the same document

Example

- Non-annotated HTML text
- Annotated HTML text with microdata

Microformats vs. Microdata

- Scalability
 - *Microformats specs are complicated because of specific rules tailored for vCard, vResume, etc.*
 - *Microdata can be easilly extensible, when new property occur they can be added witout breaking conformance of tools*
- Standards-based
 - *Microdata is a standard part of HTML5 effort*
 - *Microformats is an "ad-hoc" group of ethusiastic people, though widely supported*
 - *Strength is in underyling well-established formats*
 - *Microdata have links to Semantic Web efforts and Linked Data (via RDF), microformats not*

Overview

- Microformats
- Microdata
- **RDF and RDFa**
 - *Structured Property Values*
 - *Encoding RDF in XML (RDF/XML)*
 - *RDF-in attributes (RDFa)*
- OpenGraph Protocol

RDF

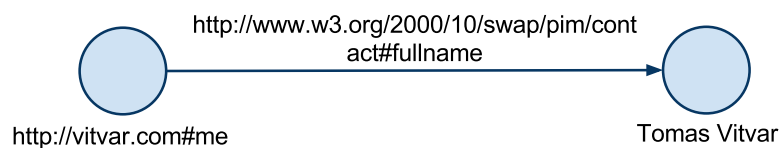
- Resource Description Framework (RDF)
 - *Resource – as defined in Web architecture*
 - *usually anything that can be conveyed electronically*
 - *plus abstract concepts that have no representation*
 - *RDF is at the bottom of Semantic Web stack of languages*
- References
 - *W3C Recommendations:*
 - *RDF Suite of W3C Recommendations* [🔗](#),
 - *RDF Primer* [🔗](#)

Meaning of Data in XML

- A resource with URI <http://www.vitvar.com/data-about-me>
- No explicit meaning of terms
 - `person`, `name`, `mailbox`, ... are terms defined in namespace <http://example.org/people> but there is no URI assigned to them
this does not work here: <http://example.org/people#name>
- No explicit meaning of relationships
 - a person has name with value *Tomas Vitvar* (\rightarrow *Tomas Vitvar is a person*),
this person has mailbox with value *tomas@vitvar.com* (\rightarrow *tomas@vitvar.com is a mailbox*), etc.
BUT this person lives?, works?, was born?, ... in a city *Innsbruck*
- Need for a language to describe statements
 \rightarrow Resource Description Framework

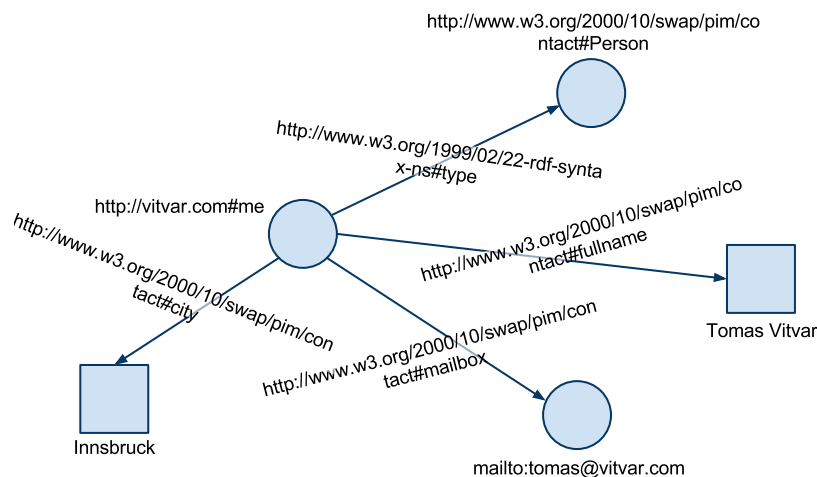
RDF Statement

- RDF Tripple: **subject** – **predicate** – **object**
 - a thing the statement describes (subject)
 - a specific property of the object (predicate)
 - a value of the property (subject)



- Representation of statements
 - using a graph notation
 - \rightarrow nodes are subject and objects (rectangles are literals)
 - \rightarrow arcs are predicates
 - identifiers to identify subject, predicate, object
 - \rightarrow URI references (URIs)
 - machine processable language
 - \rightarrow RDF serializations in triples, RDF/XML, N3, Turtle notations

Meaning of Data in RDF



- **individuals:** Tomas Vitvar identified by <http://vitvar.com#me>
- **kinds of things:** Person identified by **#Person**
 - *properties of those things, e.g., mailbox, identified by #mailbox*
 - *values of those properties, e.g. <mailto:tomas@vitvar.com>*
 - + *values of other data types such as strings, integers, dates, etc.*

References in statements

- **URI identifies**
 - *network-accessible things (electronic documents) → URL*
 - *things that are not network-accessible, such as human beings*
 - *abstract concepts that do not physically exist, such as "fullname"*
 - **RDF uses URI references to identify subjects, predicates, objects**
- **URI references (or URIref in short)**
 - *URI with an optional fragment identifier*
 - <http://www.w3.org/2000/10/swap/pim/contact#fullname>
 - **RDF resource is anything that can be identified with URIref**
 - *a set of URIrefs is called a **RDF vocabulary***
- **Literals**
 - *character strings to represent property values*
 - *can only be assigned to objects in RDF*
(in other words, objects can be either URIrefs or literals)
 - *they cannot be assigned to subjects or properties*
 - *two kinds: **plain literals** and **typed literals***

RDF Serializations – Triples Notation

- Triples notation
 - *list of all triples from RDF graph*
 - *the full triples notation requires that URI references be written out completely (in angled brackets)*
 - *very long documents, some URIrefs need to be repeated*
- Simplicity for examples
 - *QNames without angle brackets*
 - *Common prefixes and namespaces:*
 - *example*

Kinds of Things

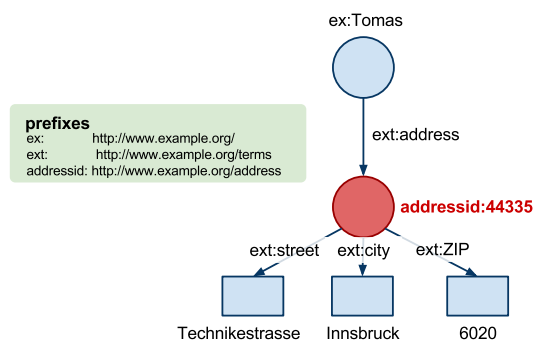
- Property **rdf:type**
 - *defines a type of a resource*
 - *corresponds to "is a member of" relationship*
 - **ext:Person** understood as a class
 - *however, RDF language does not define its semantics*
 - *RDF Schema language provides additional vocabulary for class semantics*

Overview

- Microformats
- Microdata
- RDF and RDFa
 - *Structured Property Values*
 - *Encoding RDF in XML (RDF/XML)*
 - *RDF-in attributes (RDFa)*
- OpenGraph Protocol

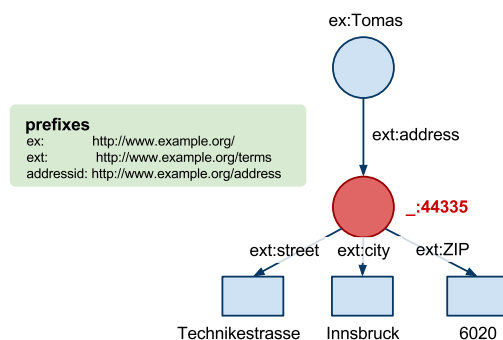
Structured Property Values

- Consider real-world complex structures
 - *Tomas works at Technikestrasse 21a, 6020 Innsbruck, Austria*
 - *One option to describe this using RDF:*
 - *But this is not often sufficient, such statements usually need to be recored as a structure, i.e. a street, a city, ZIP, ...*
- *describe Tomas's **address as a resource** that has a URIref*



Blank Nodes

- Does every structure need to have a URIref?
 - When referenced from outside of the graphs yes, otherwise not
- Blank nodes
 - Nodes that do not need to be referenced from outside of the graph
 - No need for URIref, they are only used within the graph
- Blank node identifier
 - local within a graph: `_:LocalID`, must be unique within the graph
 - two blank nodes in two graphs with the same IDs are not the same!



Modeling with Blank Nodes

- N-ary relationships
 - In fact, a blank node is a way to model an n-ary relationships
 - A blank node breaks down an n-ary to binary relationships
 - 3-ary relationship between Tomas and (Technikestr, Innsbruck, 6020)
 - Tomas – Technikestr, Tomas – Innsbruck, Tomas – 6020
- Unidentified things
 - not always good to use URIs such as e-mails to identify people
 - e-mails may change, disappear, ...
 - sometimes no need to assign unique ids to people
 - Example
 - the author of the book is `mailto:tomas@vitvar.com`, as opposed to it is a person with e-mail `mailto:tomas@vitvar.com`
 - A person is an **abstract concept** that can be modeled using a blank node

Untyped and Typed Literals

- Untyped Literals
 - *No information about how to interpret a value of the plain literal*
 - *a programme must have a knowledge how to interpret the value*
- Typed literals
 - *pairing a string with a URIref that identifies a particular datatype*
(*xsd:* refers to <http://www.w3.org/2001/XMLSchema#>)
 - *RDF does not define its own data types (except `rdf:XMLLiteral`)*
 - *no need to map external to native ones*
 - *RDF uses external data types defined in XML Schema*
 - *not all are suitable, only basic ones such as `string`, `integer`, `date`*

Overview

- Microformats
- Microdata
- RDF and RDFa
 - *Structured Property Values*
 - *Encoding RDF in XML (RDF/XML)*
 - *RDF-in attributes (RDFa)*
- OpenGraph Protocol

Basic Rules

- Representation of RDF in XML language
- Example RDF triple
 - a page `index.html` was created on August 16, 1999
- RDF/XML representation
 - We can interpret a RDF statement as:
a **description** that is **about** a subject of the statement
 - XML element (QName) of the description is the **predicate**
 - a value of the element is the **object**
 - **URIs** must be written out when in attribute values

Multiple Statements and Typed Literals

- Example RDF triples
- RDF/XML representation
 - a description may combine all properties for a single subject but there also can be a description for every subject (such representations are the same)

Blank Nodes

- Example RDF triples
- RDF/XML representation
 - A node with id `editor332` can be referenced from within the RDF graph, not outside of the RDF graph

Overview

- Microformats
- Microdata
- RDF and RDFa
 - *Structured Property Values*
 - *Encoding RDF in XML (RDF/XML)*
 - *RDF-in attributes (RDFa)*
- OpenGraph Protocol

RDFa

- Embedding RDF data in XHTML
 - *XHTML only, is extensible, HTML not*
 - *RDFa defines a number of extension attributes*
 - *Parsers may recognize RDFa annotations in HTML too*
 - *RDFa is generic to embed arbitrary RDF data*
 - *however, only standard (commonly agreed) vocabularies make sense*
- W3C Recommendations:
 - *RDFa Specification* [↗](#)
 - *RDFa Primer* [↗](#)

Property and Object Values as Resources

- Creating a property using **rel** attribute
 - *assume, following text is at <http://blog.vitvar.com/?p=107>*
 - *This corresponds to the RDF triple*
 - *When the subject is not explicitly stated, then the subject is the URL of the XHTML page being described*

Property and Object Values as Literals

- Creating a property using **property** attribute
 - *RDFa defines a **property** extension attribute*
 - *assume, following text is at <http://blog.vitvar.com/?p=107>*
 - *This corresponds to the RDF triple*
- Typed literals
 - *RDFa defines a **datatype** extension attribute*
- Alternative content
 - *RDFa defines **content** extension attribute*
 - *replaces the object value that is in the element's value*

Subject

- Creating a subject using **about** attribute
 - *RDFa defines **about** extension attribute*
 - *Let the following text is at <http://blog.vitvar.com/?p=107>*
 - *This corresponds to the RDF triple*
 - *Also possible to use multiple subjects on a single page*

Types and Blank Nodes

- Types
 - RDFa defines **typeof** extension attribute
 - corresponds to **rdf:type** property
- Blank node
 - When annotation has **typeof** but not **about**
 - blank node, that is, a node without a subject
 - I know Peter who has e-mail petr@novak.cz

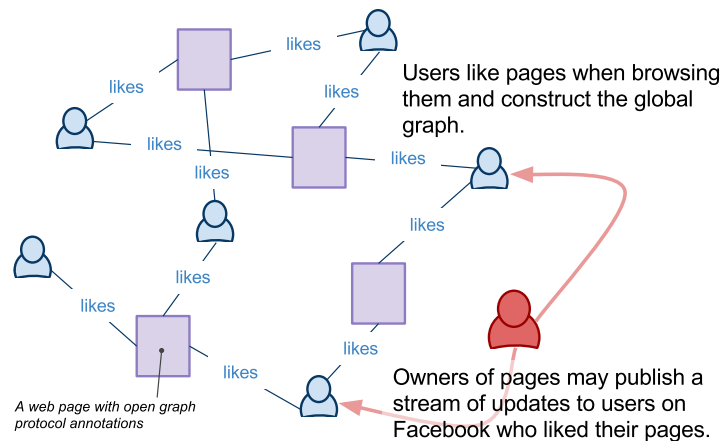
Overview

- Microformats
- Microdata
- RDF and RDFa
- **OpenGraph Protocol**

OpenGraph Protocol

- Global Social Graph

- *important adoption of RDFa, see [Open Graph Protocol](#)*
- *defines meta-data for pages' description so that it can be easily included in a global graph connecting people and pages through "likes" (a person – likes – a page)*



Page Annotations

- Open Graph protocol main properties

- *a page is the subject in the RDF triple*
- **og:title** – *title of the page*
- **og:type** – *type of the content (e.g., movie)*
- **og:image** – *URL of the image for the page*
- **og:url** – *a canonical URL of the page to be used as its permanent ID in the graph*

- HTML page annotation RDFa example

Publishing updates

- Ownership
 - Page must be associated with a Facebook application
 - using **fb:app_id** meta tag
 - Owners can publish a stream of updates using the Facebook Graph API [🔗](#)
- Getting access
- Publishing updates