

Information systems modeling

Tomasz Kubik



Publishing resources on the Web



Cool URIs for the Semantic Web

W3C Interest Group Note 03 December 2008

This version:

<http://www.w3.org/TR/2008/NOTE-cooluris-20081203/>

Latest version:

<http://www.w3.org/TR/cooluris/>

Previous version:

<http://www.w3.org/TR/2008/NOTE-cooluris-20080331/>

Editors:

[Leo Sauermann](#) (DFKI GmbH)

[Richard Cyganiak](#) (DERI, NUI Galway and Freie Universität Berlin)

Contributors:

[Danny Ayers](#) (Talis Information Ltd.)

[Max Völkel](#) (FZI Karlsruhe)

Please refer to the [errata](#) for this document, which may include some corrections.

Copyright © 2008 W3C® (MIT, ERCIM, Keio), All Rights Reserved. W3C [liability](#), [trademark](#) and [document use](#) rules apply.

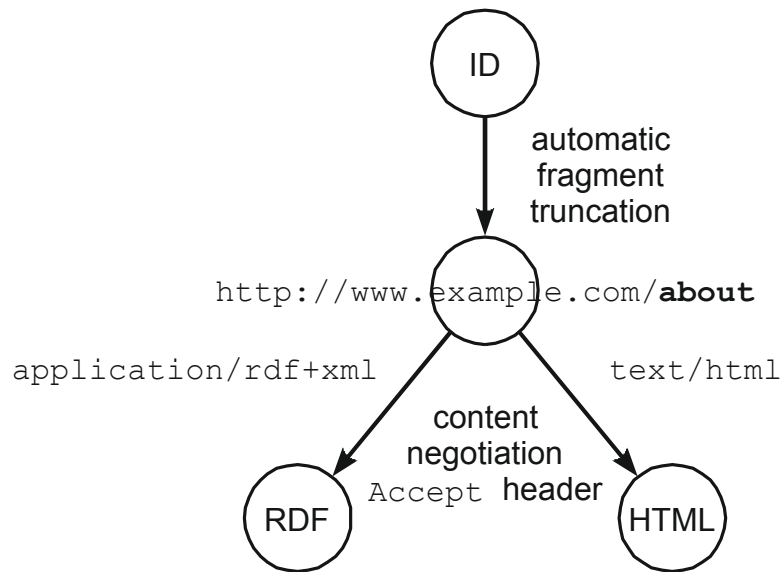
Abstract

The *Resource Description Framework* RDF allows users to describe both Web documents and concepts from the real world—people, organisations, topics, things—in a computer-processable way. Publishing such descriptions on the Web creates the *Semantic Web*. URIs (Uniform Resource Identifiers) are very important, providing both the core of the framework itself and the link between RDF and the Web. This document presents guidelines for their effective use. It discusses two strategies, called *303 URIs* and *hash URIs*. It gives pointers to several Web sites that use these solutions, and briefly discusses why several other proposals have problems.

Hash URI

with content negotiation:

`http://www.example.com/about#john`



Content-Location:

`http://www.example.com/about.rdf`

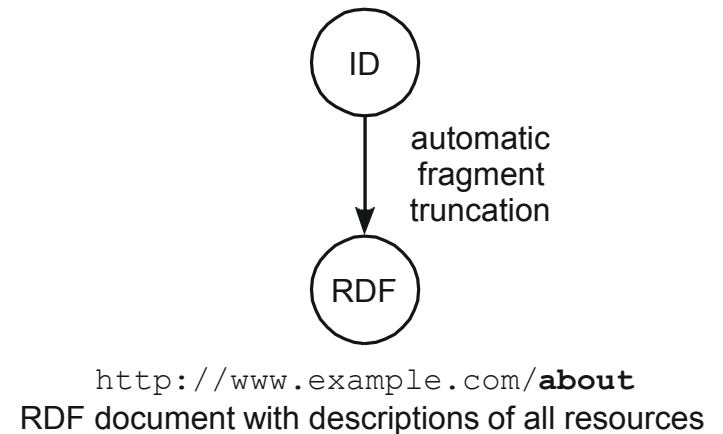
Content-Location:

`http://www.example.com/about.html`

HTML document with descriptions of all resources

without content negotiation:

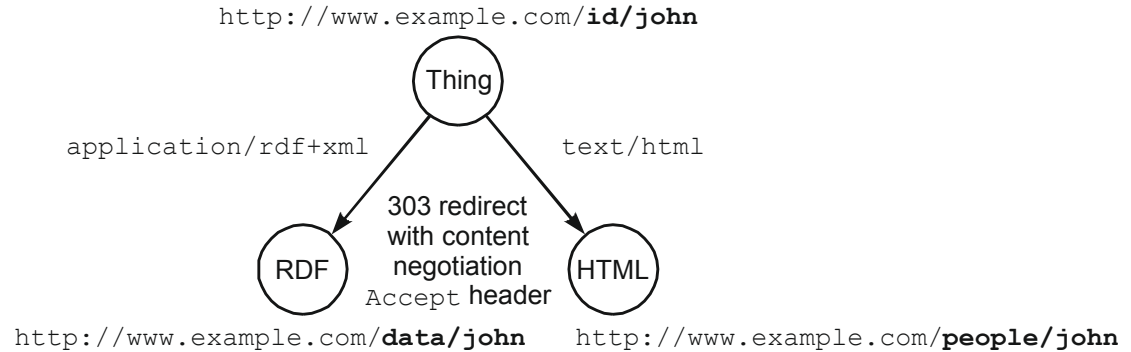
`http://www.example.com/about#john`



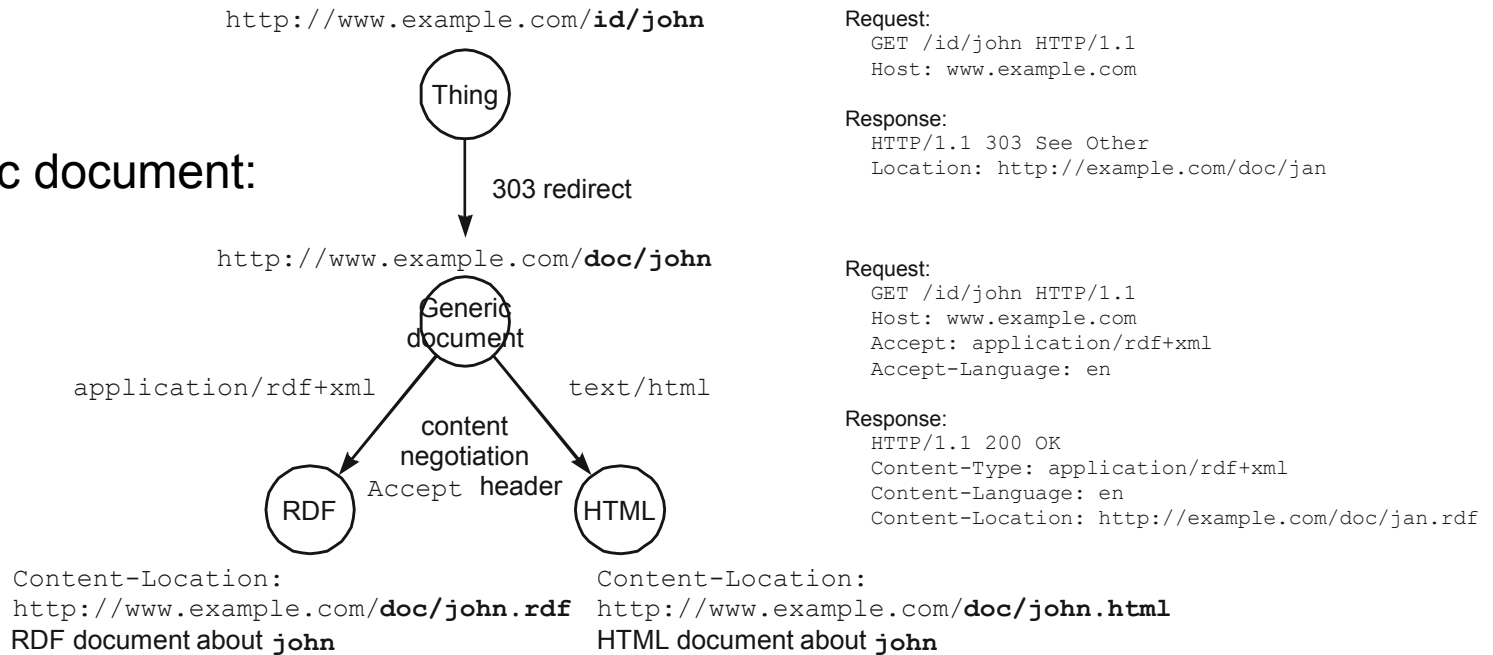
https://developer.mozilla.org/en-US/docs/Web/HTTP/Content_negotiation

303 forwarding

different documents:



one generic document:



Publishing linked data

LINKED DATA PATTERNS

[Next](#)

Linked Data Patterns

A pattern catalogue
for modelling,
publishing, and
consuming Linked
Data

Leigh Dodds Ian Davis

2012-05-31

<https://patterns.dataincubator.org/book/>

Hierarchical URIs

How should URIs be assigned to a group of resources that form a natural hierarchy?

Context

It is often the case that a collection of resources may form a natural hierarchy. E.g. the chapters within a book, or the departments within an organization. Reflecting this strict hierarchy within the URI structure makes those URIs more hackable allowing users/developers to "navigate" up the hierarchy by pruning the URI.

Solution

Where a natural hierarchy exists between a set of resources use [Patterned URIs](#) that conform to the following pattern:

```
:collection/:item/:sub-collection/:item
```

E.g. in a system which is publishing data about individual books and their chapters, we might use the following identifier for chapter 1 of a specific book:

```
/books/12345/chapters/1
```

The `/chapters` URI will naturally reflect to the collection of all chapters within a specific book. The `/books` URI maps to the collection of all books within a system, etc.

Example(s)

The discogs dataset in dataincubator uses hierarchical uris of the form:

```
http://discogs.dataincubator.org/release/22530/track/1-01
```

Discussion

This technique is best suited to scenarios where the items in the sub-collections (chapters) are always associated with a single parent item. Other relationships might exist, e.g. the chapter may be included in another but the chapter is always associated with at least one book: they do not exist in isolation. In circumstances where this doesn't hold true, then it is best to just use simple Patterned URIs.

The same applies to circumstances where the hierarchy may change over time.

Remark:

It is quite similar to resource-oriented architecture (ROA) <https://www.developer.com/web-services/overview-of-resource-oriented-architectures/>

<https://patterns.dataincubator.org/book/hierarchical-uris.html>

Autodiscovery

How can people find the underlying linked data for a given web page?

Context

Currently the Web is predominately built from interlinked HTML pages. Linking directly to linked data documents from HTML presents the risk of confusion for non-technical audiences. However the publisher requires that the underlying data be discoverable by linked data aware tools and indexable by search engines.

Solution

When publishing a web page derived from linked data include a link element in the head of the web page pointing to the original data.

```
<link rel="meta" type="application/rdf+xml" title="Raw Data" href="http://example.com/data.rdf"/>
```

Example(s)

The FOAF Vocabulary recommends linking a homepage to an equivalent FOAF profile using the link element.

The Semantic Radar Firefox plugin uses the autodiscovery pattern to detect the presence of linked data related to the web page the user is viewing.

Discussion

Until web browsers become fully linked data aware it may not be satisfactory to link directly to linked data pages from the body of an HTML page. HTML provides the link element to allow publishers to include links to information and resources that may be relevant in addition to the main content of the page. Web browsers may choose to display these links in their user interface. Web search engines can use these links to discover additional information that may make their search more relevant for the user.

<https://patterns.dataincubator.org/book/autodiscovery.html>

RDFa

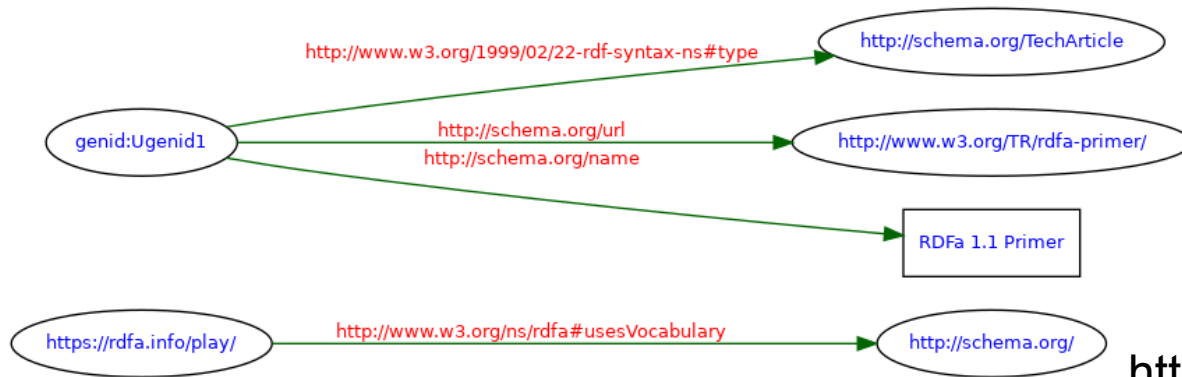
- RDFa is a structured data markup language used to embed semantical metadata in web documents, see:
 - *RDFa Core 1.1 - Third Edition. Syntax and processing rules for embedding RDF through attributes*, <https://www.w3.org/TR/rdfa-core/>
 - *HTML+RDFa 1.1 - Second Edition. Support for RDFa in HTML4 and HTML5*, <https://www.w3.org/TR/html-rdfa/>.
 - *RDFa 1.1 Primer - Third Edition. Rich Structured Data Markup for Web Documents*, <https://www.w3.org/TR/rdfa-primer/>
- Web pages enriched with RDFa annotations offer structured data that can be consumed by search engines
- The content of such web pages can be verified with the use of <http://rdfa.info/play/> service or structured data testing tools (see <https://developers.google.com/search/docs/advanced/structured-data>)

RDFa, RDFa Lite

<https://rdfa.info/>

- RDF graphs might be embedded through the use of **simple attributes** in existing markup languages such as HTML.
 - @about indicates subject IRI reference
 - @property, @rel, or @rev represent predicates
 - @resource, @src, or @href represent objects which are IRI references
 - @content represents objects that are literals (literals are also the content of the elements in question, with an optional datatype expressed using @datatype, and an optional language expressed using a Host Language-defined mechanism such as @xml:lang).
- In this way HTML document can become a holder for RDF data.

```
<span vocab="http://schema.org/" typeof="TechArticle">  
  <a property="url" href="http://www.w3.org/TR/rdfa-primer/">  
    <span property="name">RDFa 1.1 Primer</span></a>.  
</span>
```



CheetSheet:

- syntax attributes: @prefix, @vocab.
- subject attributes: @about.
- predicate attributes: @property, @rel, @rev.
- resource attributes: @resource, @href, @src.
- literal attributes: @datatype, @content, @xml:lang or @lang.
- macro attributes: @typeof, @inlist.

<https://www.w3.org/TR/rdfa-lite/>
<https://www.w3.org/TR/rdfa-primer/>

Microdata

- Used to nest metadata within existing content on web pages through attributes
 - @itemscope
 - creates the Item and indicates that descendants of this element contain information about it.
 - @itemtype
 - describes the item and its properties context by a valid URL of a vocabulary
 - @itemid
 - Indicates a unique identifier of the item.
 - @itemprop
 - Indicates that its containing tag holds the value of the specified item property. The property's name and value context are described by the item's vocabulary. Properties values usually consist of string values, but can also use URLs using the a element and its href attribute, the img element and its src attribute, or other elements that link to or embed external resources.
 - @itemref
 - Properties that are not descendants of the element with the itemscope attribute can be associated with the item using this attribute. Provides a list of element ids (not itemids) with additional properties elsewhere in the document.

Welcome to the microformats wiki! (http://microformats.org/wiki/Main_Page)
W3C Working Draft on microdata (<https://www.w3.org/TR/microdata/>)
<https://html.spec.whatwg.org/multipage/microdata.html>

RDFa vs Microdata

RDFa Lite 1.1	Microdata 1.0
<pre><div vocab="http://schema.org/" typeof="Product"> Dell UltraSharp 30" LCD Monitor </div></pre>	<pre><div itemscope itemtype="http://schema.org/Product"> Dell UltraSharp 30" LCD Monitor </div></pre>

Mapping microdata to RDF:

- @itemprop is replaced with @property.
- @itemscope is dropped.
- @itemtype is replaced with @typeof.

Only two features of Microdata aren't supported by RDFa Lite:

- @itemref and @itemscope.

microformats, microformats 2

- A set of simple, open data formats built upon existing and widely adopted standards
- Designed to openly publish contacts, events, reviews, recipes, and other structured information on the web
- Relay on the use of `class` and `rel` attributes
- **Webmention** standard uses microformats to provide a way in which messages and comments can be sent from one site to another
- `microformats.org` maintains the official registry of HTML `rel` values

As plain text:

Barnaby Walters

With HTML:

```
<a href="http://waterpigs.co.uk">Barnaby Walters</a>
```

With classic microformats:

```
<span class="vcard"><a class="fn n url" href="http://waterpigs.co.uk">Barnaby Walters </a> </span>
```

With microformats 2:

```
<a class="h-card" href="http://waterpigs.co.uk">Barnaby Walters</a>
```

<http://microformats.org>, http://microformats.org/wiki/Main_Page-pl
<https://www.w3.org/TR/webmention/>
<https://microformats.org/wiki/existing-rel-values>
<https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/rel>

Microformats vocabularies

- hCard
 - <https://microformats.org/code/hcard/creator>
- hCalendar
 - <https://microformats.org/code/hcalendar/creator>
- rel-license
- rel-nofollow
- rel-tag
- VoteLinks
- XFN - XFN creator
- XMDP
- XOXO
 - RFC2426

<http://microformats.org/profile/hcard>

<http://microformats.org/wiki/hcard>

The screenshot shows the hCard Creator interface. On the left, there is a form titled "hCard-o-matic" with input fields for: given name, middle name, family name, organization, street, city, state/province, postal code, country name, phone, email, uri, photo url, AIM screenname, YIM screenname, Jabber screenname, and Categories (comma separated). Below the form are "Reset" and "Build It!" buttons. On the right, there is a "code" section displaying the generated HTML code for an hCard. Below the code is a "preview" section showing a rendered hCard with the text "This hCard created with the hCard creator".

The screenshot shows the hCalendar Creator interface. On the left, there is a form titled "hCalendar-o-matic" with input fields for: summary, event title, location, uri, start (date and time), end (date and time), TZ (time zone), and description. There are also "Reset" and "Build It!" buttons. On the right, there are two sections: "code" showing the generated HTML code for an hCalendar event, and "compact code" showing a more compact version of the code. Below the code is a "preview" section showing a rendered hCalendar event with the text "May 5, 2004 - event title" and "This hCalendar event brought to you by the hCalendar Creator".

Well known vocabularies

- According to best practices the semantical annotations should be declared with the use of well known vocabularies. These include:
 - schema.org:
 - project home page, <https://schema.org/>,
 - downloads: <https://schema.org/docs/datamodel.html>, <http://topbraid.org/schema/>
 - goodRelations, which derives from schema.org:
 - project home page <http://www.heppnetz.de/projects/goodrelations/> and
 - *GoodRelations Language Reference*, <http://www.heppnetz.de/ontologies/goodrelations/v1.html>
 - downloads: <http://www.heppnetz.de/ontologies/goodrelations/v1.owl>
 - Dublin Core (ISO Standard 15836:2009, ANSI/NISO Standard Z39.85-2012)
 - project home page, <http://dublincore.org/>
 - *Dublin Core Metadata Element Set, Version 1.1: Reference Description*, <http://www.dublincore.org/documents/dces/>
 - downloads: <http://dublincore.org/2012/06/14/dcelements.rdf>, <http://dublincore.org/2012/06/14/dcterms.rdf>
 - foaf:
 - *FOAF Vocabulary Specification 0.99*, <http://xmlns.com/foaf/spec/>
- Creative Commons (licensing)
 - <http://creativecommons.org/ns>
 - <http://creativecommons.org/schema.rdf>

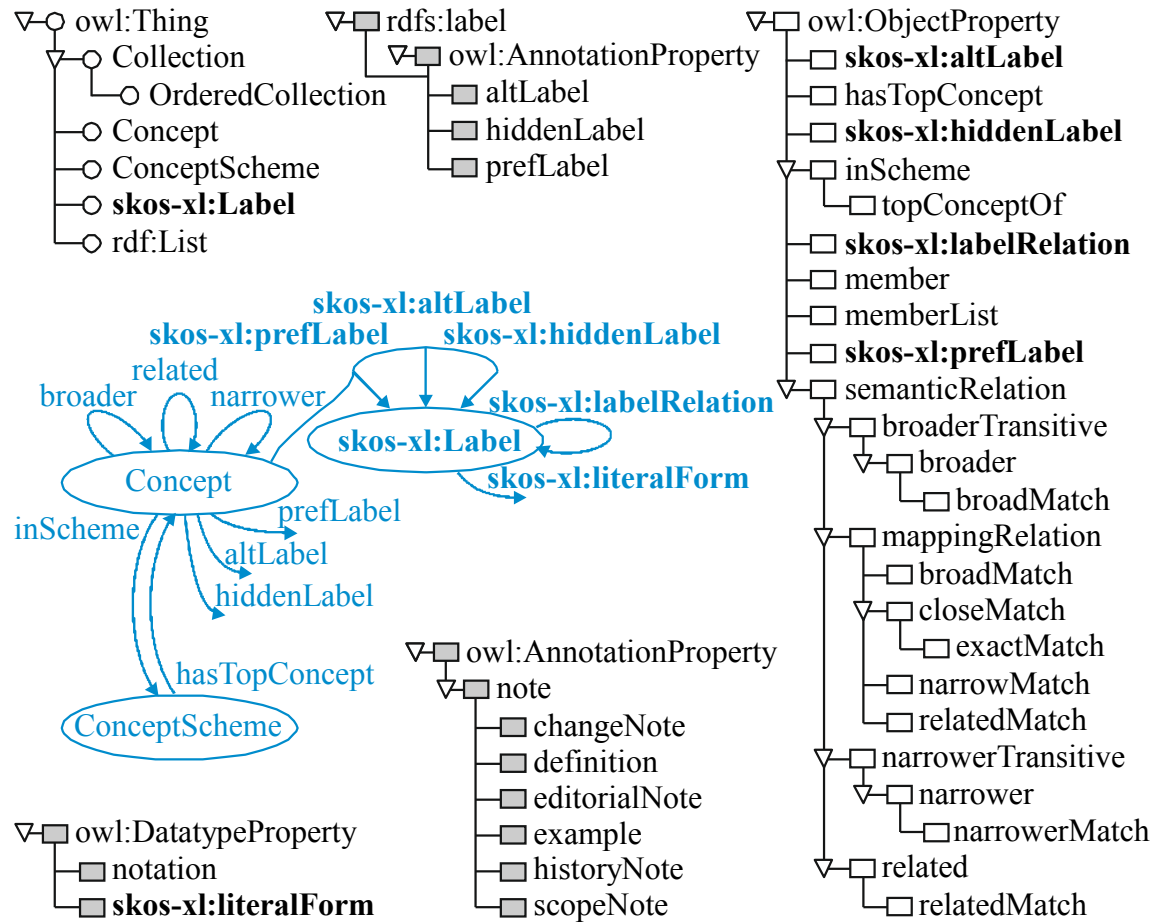
Publishing medatata as JSON-LD

- JSON-LD is an extension of JSON oriented at serialization of semantical graphs, see
 - JSON-LD Playground, <https://json-ld.org/playground/>
 - JSON for Linking Data, <https://json-ld.org/>
- One of the simplests method for embedding structured data inside web pages relies on using `<script>` node as a placeholder of the whole graph encoded in JSON-LD.

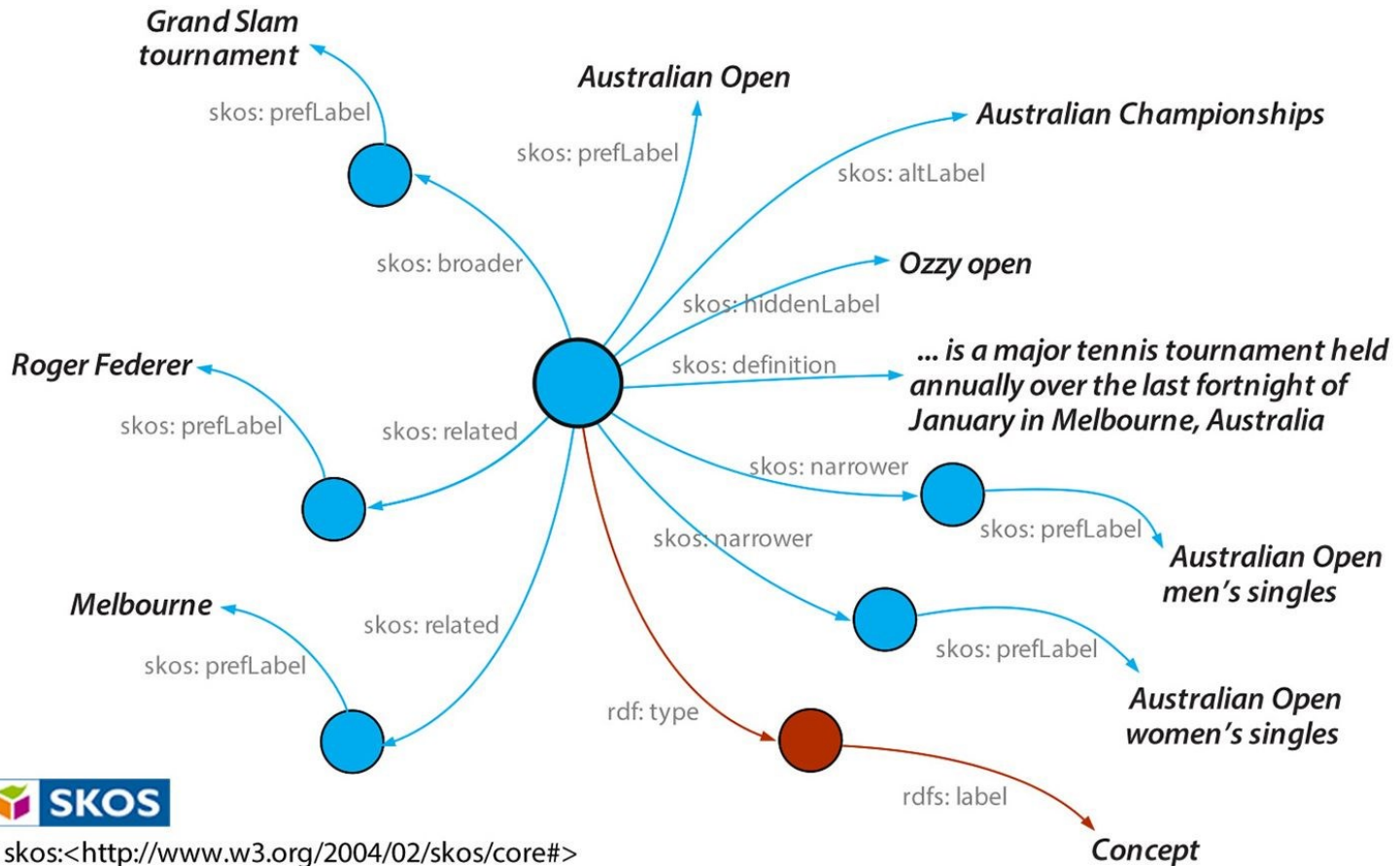
SKOS

- Simple Knowledge Organisation System

- <https://www.w3.org/TR/skos-reference/>
- <http://www.w3.org/2009/08/skos-reference/skos.rdf>

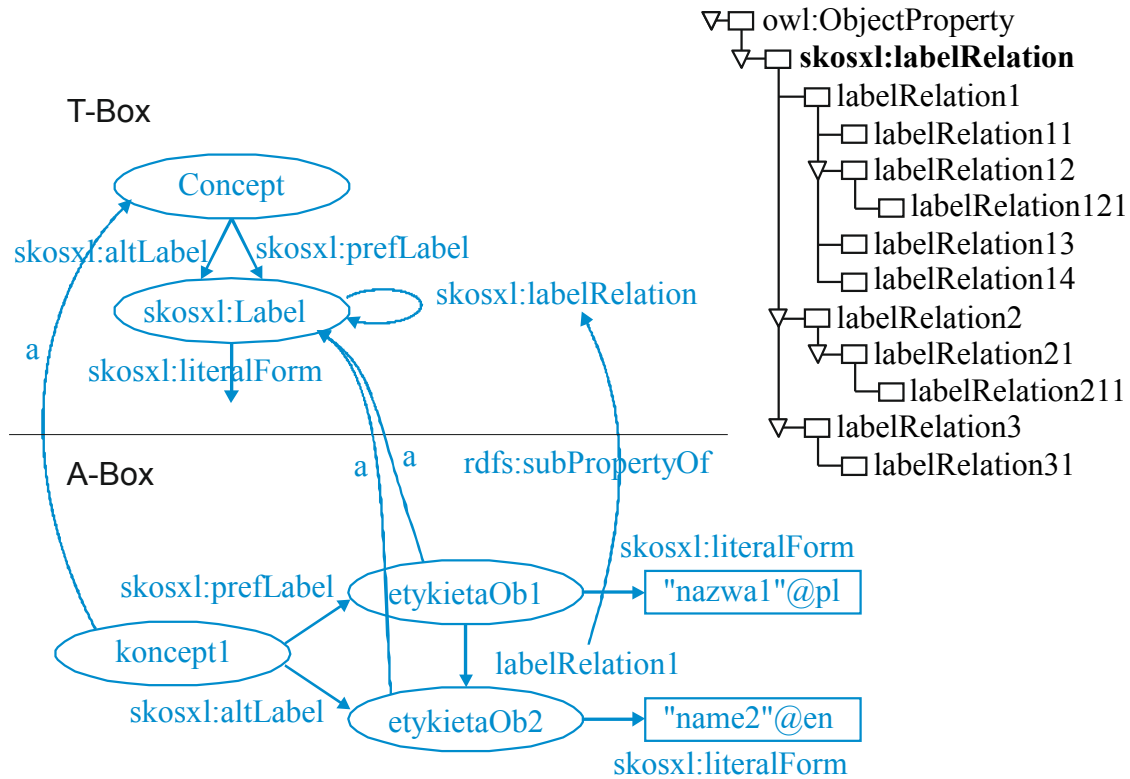


SKOS example

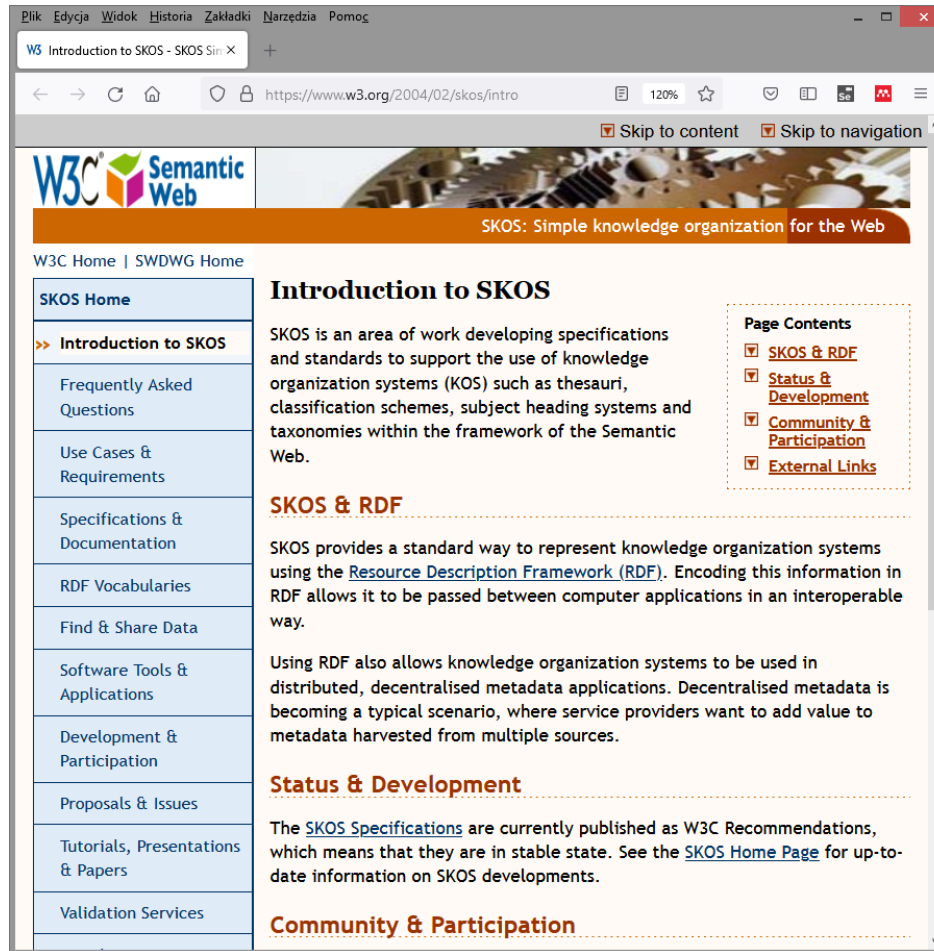


<https://www.w3.org/TR/skos-reference/>

SKOS XL



SKOS resources



The screenshot shows a web browser window displaying the SKOS introduction page. The browser's address bar shows the URL <https://www.w3.org/2004/02/skos/intro>. The page features the W3C Semantic Web logo and a navigation menu on the left. The main content area is titled "Introduction to SKOS" and includes a "Page Contents" sidebar with links to "SKOS & RDF", "Status & Development", "Community & Participation", and "External Links".

W3C Home | SWDVG Home

SKOS Home

- >> **Introduction to SKOS**
- Frequently Asked Questions
- Use Cases & Requirements
- Specifications & Documentation
- RDF Vocabularies
- Find & Share Data
- Software Tools & Applications
- Development & Participation
- Proposals & Issues
- Tutorials, Presentations & Papers
- Validation Services

Introduction to SKOS

SKOS is an area of work developing specifications and standards to support the use of knowledge organization systems (KOS) such as thesauri, classification schemes, subject heading systems and taxonomies within the framework of the Semantic Web.

SKOS & RDF

SKOS provides a standard way to represent knowledge organization systems using the [Resource Description Framework \(RDF\)](#). Encoding this information in RDF allows it to be passed between computer applications in an interoperable way.

Using RDF also allows knowledge organization systems to be used in distributed, decentralised metadata applications. Decentralised metadata is becoming a typical scenario, where service providers want to add value to metadata harvested from multiple sources.

Status & Development

The [SKOS Specifications](#) are currently published as W3C Recommendations, which means that they are in stable state. See the [SKOS Home Page](#) for up-to-date information on SKOS developments.

Community & Participation

Page Contents

- SKOS & RDF
- Status & Development
- Community & Participation
- External Links

<https://www.w3.org/2004/02/skos/intro>

SKOS tools

The image displays two screenshots of the PoolParty Thesaurus Server interface. The left screenshot shows a hierarchical tree view of concepts. A context menu is open over the 'Products' category, showing options: 'Create Top Concept...', 'Import', 'Export', 'Apply Custom Class...', and 'Delete'. The right screenshot shows a detailed view of the 'PoolParty Thesaurus Server' concept. The interface includes a header with the title 'PoolParty Thesaurus Server' and a URL 'http://vocabulary.semantic-web.at/samweb/1629'. Below the header, there are tabs for 'Details', 'Notes', 'Documents', 'Linked Data', 'Triples', 'Visualization', 'Quality Report', and 'History'. The 'Details' tab is active, showing a table of properties: 'isUsedBy' and 'hasVersion' with plus signs for expansion, and 'uses' with a plus sign. A list of related concepts is shown below the table, including '04.01 Applications (30)', '04.02 Services (6)', '04.03 Software systems (6)', 'App store (4)', 'Clustering Tools (3)', 'Database management systems (4)', 'Operating systems (6)', 'Semantic technologies (12)', 'Alignment Tools (8)', 'Crawler (2)', 'Inference engine (5)', 'Lookup Service (4)', 'Ontology server (2)', 'ONKI Ontology Library Server (0)', 'PoolParty Thesaurus Server (2)', 'PoolParty Linked Data Harvester (0)', 'PoolParty quality checker (0)', 'SPARQL tools (5)', 'Search technologies (4)', and 'Semantic web technologies (0)'. The 'PoolParty Thesaurus Server' concept is highlighted in orange.

<https://conf-doc.poolparty.biz/pp6/poolparty-overview/poolparty-skos-rdf-and-uris/skos-and-poolparty>

Reasoning in OWL

see: <http://tomasz.kubik.staff.iar.pwr.wroc.pl/dydaktyka/InformationSystemsModeling/2020/CadinalityTest.pdf>

```
@prefix : <http://www.example.org/ontologies/cnt#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <http://www.example.org/ontologies/cnt> .

<http://www.example.org/ontologies/cnt> rdf:type owl:Ontology .
#####
# Object Properties
#####
:hasBrother rdf:type owl:ObjectProperty .

#####
# Data properties
#####
:hasHeight rdf:type owl:DatatypeProperty ;
    rdfs:range xsd:integer .

#####
# Classes
#####

:Any rdf:type owl:Class ;
    owl:equivalentClass [ rdf:type owl:Restriction ;
        owl:onProperty :hasBrother ;
        owl:someValuesFrom owl:Thing
    ] .
:Average rdf:type owl:Class ;
    owl:equivalentClass [ owl:intersectionOf ( [ rdf:type owl:Restriction ;
        owl:onProperty :hasBrother ;
        owl:minCardinality "3"^^xsd:nonNegativeInteger
    ]
        [ rdf:type owl:Restriction ;
        owl:onProperty :hasBrother ;
        owl:maxCardinality "4"^^xsd:nonNegativeInteger
    ]
    );
    rdf:type owl:Class
].
:Few rdf:type owl:Class ;
```

```
owl:equivalentClass [ owl:intersectionOf ( [ rdf:type owl:Restriction ;
        owl:onProperty :hasBrother ;
        owl:minCardinality "1"^^xsd:nonNegativeInteger
    ]
        [ rdf:type owl:Restriction ;
        owl:onProperty :hasBrother ;
        owl:maxCardinality "2"^^xsd:nonNegativeInteger
    ]
    );
    rdf:type owl:Class
].

:Many rdf:type owl:Class ;
    owl:equivalentClass [ rdf:type owl:Restriction ;
        owl:onProperty :hasBrother ;
        owl:minCardinality "5"^^xsd:nonNegativeInteger
    ] .

:Person rdf:type owl:Class .

#####
# Individuals
#####
:i1 rdf:type owl:NamedIndividual ,
    :Person .
:i2 rdf:type owl:NamedIndividual ,
    :Person .
:i3 rdf:type owl:NamedIndividual ,
    :Person .
:i4 rdf:type owl:NamedIndividual ,
    :Person .
:i5 rdf:type owl:NamedIndividual ,
    :Person .
:i6 rdf:type owl:NamedIndividual ,
    :Person .
:t1 rdf:type owl:NamedIndividual ,
    :Person ;
    :hasBrother :i1 ,
        :i2 ,
        :i3 .
```

Closing the world in SPARQL

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX cnt: <http://www.example.org/ontologies/cnt#>
CONSTRUCT {?s a cnt:Few}

WHERE {
    SELECT ?s (count(?p) AS ?numBrothers)
        WHERE {
            ?s cnt:hasBrother ?p
        } GROUP BY ?s
        HAVING (?numBrothers > 0 && ?numBrothers <= 2)
}
```

Remarks on SPARQL

- Specification and overview of SPARQL features and use cases can be found in:
 - SPARQL 1.1 Query Language, <https://www.w3.org/TR/sparql11-query/>
 - SPARQL 1.1 Overview, <https://www.w3.org/TR/sparql11-overview/>
- SPARQL is not only an RDF query language but also a protocol built on the top of HTTP, offering two operations: query and update, see:
 - SPARQL 1.1 Protocol, <https://www.w3.org/TR/sparql11-protocol/>).
- SPARQL endpoints are not RESTfull edpoints
 - they accept GET request delivered in `query` attribute (not in the path) thus the queries must be html encoded before use
 - update/delete requests are not handled with the use of PUT/DELETE HTTP methods
 - responses may be encoded in different ways (ttl, json, json-ld, csv - depends on sparql endpoint abilities)
 - there is no mapping similar to ORM which would be used to convert received responses into objects in chosen programmin language (the model is a graph model, not relational one).