

Data Interoperability and Semantics

< Part 3. Data schemas and semantics >

ICM – Toolbox Engineering and Interoperability of Software Systems – Course unit on Data Interoperability and Semantics

M1 Cyber Physical and Social Systems – Course unit on Data Interoperability and Semantics

Maxime Lefrançois <https://maxime-lefrancois.info>

Course unit URL: <https://ci.mines-stetienne.fr/cps2/course/data>

Data Interoperability and Semantics

Outline

- < Part 3. Data schemas and semantics >
 - Part 3.1. Data schemas
 - Part 3.1.1. XML Schema
 - Part 3.1.2. JSON Schema
 - Part 3.2. Semantics
 - Part 3.2.1. Heterogeneities and data conflicts
 - Part 3.2.2. Controlled vocabularies and ontologies
 - Part 3.2.3. Resource Description Framework
 - Part 3.2.4. RDFa: Rich structured data markup for web documents
 - Part 3.2.5. JSON-LD: JSON for Linking Data

ICM – Computer Science Major – Course unit on Data Interoperability and Semantics

M1 Cyber Physical and Social Systems – Course unit on Data Interoperability and Semantics

Maxime Lefrançois <https://maxime-lefrancois.info>

Course unit URL: <https://ci.mines-stetienne.fr/cps2/course/data>

Data Interoperability and Semantics

< Part 3. Data schemas and semantics >

Part 3.1. Data schemas

ICM – Toolbox Engineering and Interoperability of Software Systems – Course unit on Data Interoperability and Semantics

M1 Cyber Physical and Social Systems – Course unit on Data Interoperability and Semantics

Maxime Lefrançois <https://maxime-lefrancois.info>

Course unit URL: <https://ci.mines-stetienne.fr/cps2/course/data>

Data Interoperability and Semantics

< Part 3. Data schemas and semantics >

Part 3.1. Data schemas

Part 3.1.1. XML Schema

ICM – Toolbox Engineering and Interoperability of Software Systems – Course unit on Data Interoperability and Semantics

M1 Cyber Physical and Social Systems – Course unit on Data Interoperability and Semantics

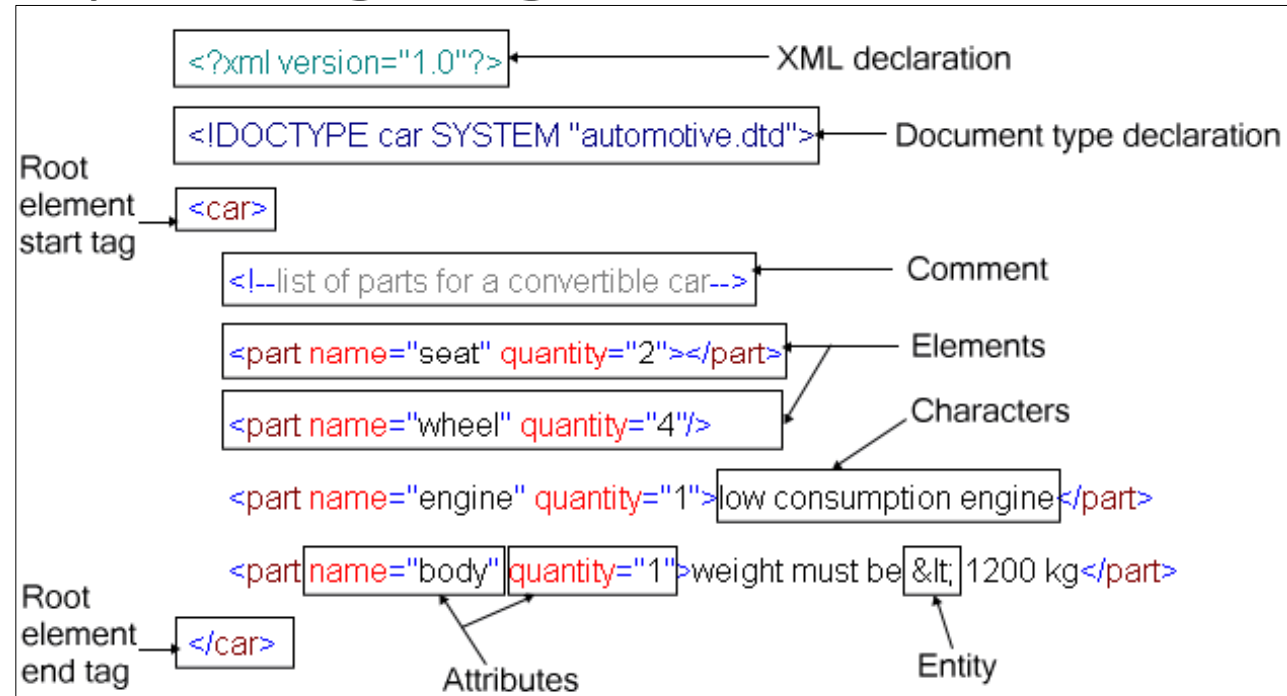
Maxime Lefrançois <https://maxime-lefrancois.info>

Course unit URL: <https://ci.mines-stetienne.fr/cps2/course/data>

Extensible Markup Language

XML (file format)

Filename extension	.xml
Internet media type	application/xml text/xml [1]
Uniform Type Identifier (UTI)	public.xml
UTI conformation	public.text
Magic number	<?xml
Developed by	World Wide Web Consortium
Type of format	Markup language
Extended from	SGML
Extended to	Numerous languages, including XHTML · RSS · Atom · KML
Standard	1.0 (Fifth Edition) [2] (November 26, 2008; 12 years ago) 1.1 (Second Edition) [2] (August 16, 2006; 15 years ago)
Open format?	Yes



- v1.0 in 1998, still extensively used in many verticals
- numerous formats based on XML (418 registered on IANA)
https://en.wikipedia.org/wiki/List_of_XML_markup_languages
 application/atom+xml application/rdf+xml ...
- verbosity, complexity and redundancy

Extensible Markup Language

XML (file format)

Filename extension	.xml
Internet media type	application/xml text/xml [1]
Uniform Type Identifier (UTI)	public.xml
UTI conformation	public.text
Magic number	<?xml
Developed by	World Wide Web Consortium
Type of format	Markup language
Extended from	SGML
Extended to	Numerous languages, including XHTML · RSS · Atom · KML
Standard	1.0 (Fifth Edition)  (November 26, 2008; 12 years ago) 1.1 (Second Edition)  (August 16, 2006; 15 years ago)
Open format?	Yes

Characters and escaping

- unicode implementations: `<?xml version="1.0" encoding="UTF-8"?>`
- escaping characters: `<`; `'<' - &`; `'&' - ❤`; `'♥'` - etc.

Syntactical correctness

- *well formed vs ill-formed*
- one root tag
- correct nesting
- tag names (approx) start with letter, then alphanumeric or `'.'`

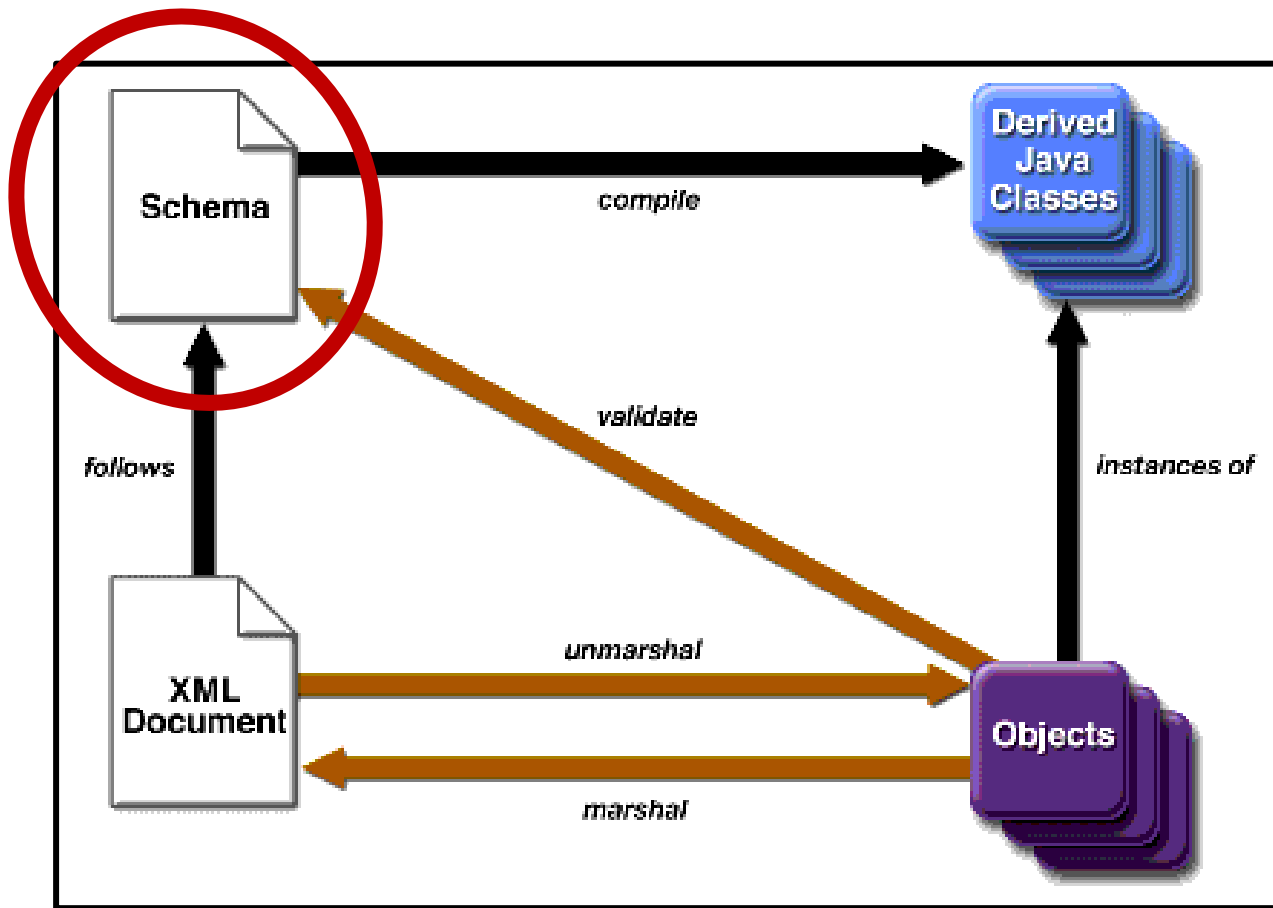
Schemas and validation

- *valid vs invalid*
- DTD, or XML Schema

Namespaces

- `xmlns:ns1="http://example.org/ns1"`
`xmlns:ns2="http://example.org/ns2"`
- allows to use different schemas together: `<ns1:Tag>` `<ns2:Tag>`

XML – OOP data binding

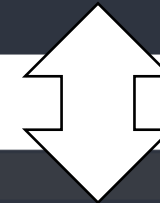


XML-OOP data binding

ex Java: <https://zetcode.com/java/jaxb/>

Employee.java

```
@XmlElement(name = "employee")
@XmlAccessorType(XmlAccessType.FIELD)
public class Employee implements Serializable
{
    private Integer id;
    private String firstName;
    private String lastName;
}
```



employee.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<employee>
    <firstName>Lokesh</firstName>
    <id>1</id>
    <lastName>Gupta</lastName>
</employee>
```

Java Architecture for XML Binding (JAXB) example

<https://howtodoinjava.com/jaxb/jaxb-annotations/>

XML Schema

- Document Type Definition (DTD, 2008)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

link to some .dtd document



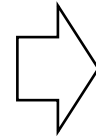
[...]

```
<!ELEMENT html (head, body)>
<!ELEMENT p (#PCDATA | p | ul | dl | table | h1|h2|h3)*>
```

[...]

```
<!ATTLIST img
  src      CDATA      #REQUIRED
  id       ID         #IMPLIED
  sort     CDATA      #FIXED "true"
  print    (yes | no) "yes"
>
```

[...]



```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<!-- the XHTML document body starts here-->
<html xmlns="http://www.w3.org/1999/xhtml">
  ...
</html>
```

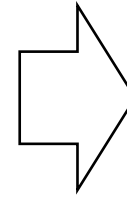
HTML 5 : no more link to a DTD

XML Schema

- Regular Language for XML Next Generation (RELAX NG, 2001) <https://relaxng.org/>

```
<element name="addressBook" xmlns="http://relaxng.org/ns/structure/1.0">
  <zeroOrMore>
    <element name="card">
      <element name="name">
        <text/>
      </element>
      <element name="email">
        <text/>
      </element>
    </element>
  </zeroOrMore>
</element>
```

RELAX NG document



```
<addressBook>
  <card>
    <name>John Smith</name>
    <email>js@example.com</email>
  </card>
  <card>
    <name>Fred Bloggs</name>
    <email>fb@example.net</email>
  </card>
</addressBook>
```

Valid XML document

XML Schema

- XML Schema (W3C, 2003)

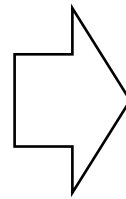
The only one you should use (if you need to)

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

  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="heading" type="xs:string"/>
        <xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

XML Schema document



```
<?xml version="1.0"?>

<note
  xmlns="https://www.w3schools.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="https://www.w3schools.com/xml note.xsd">
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

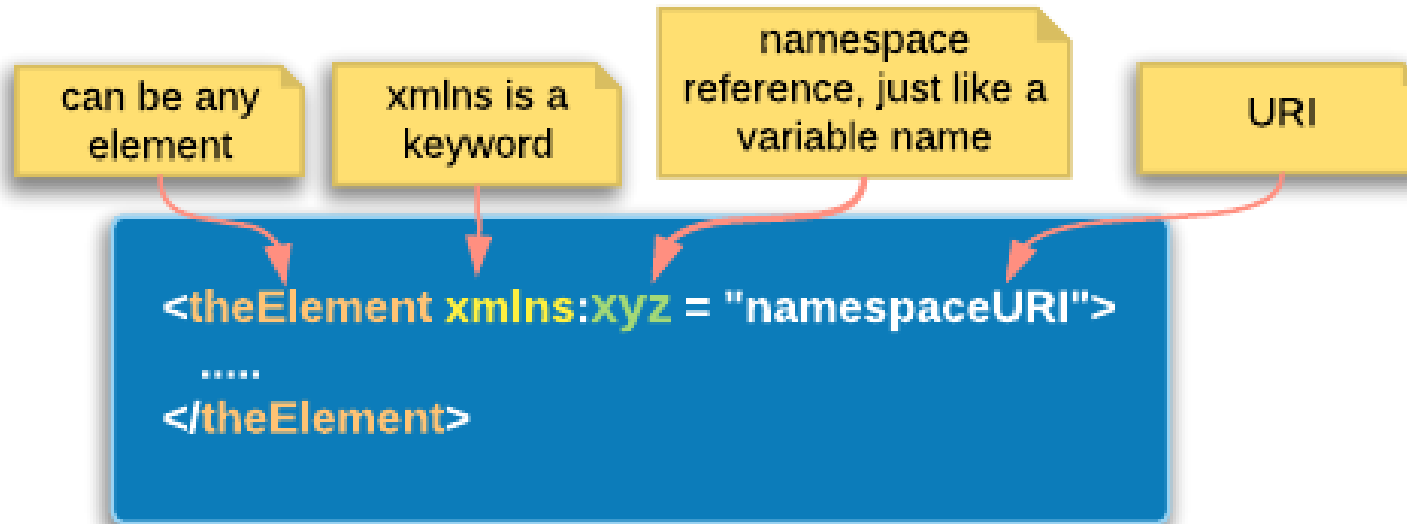
Valid XML document

XML Schema

- Tutorials
 - https://www.tutorialspoint.com/xml/xml_schemas.htm
 - https://www.w3schools.com/xml/schema_intro.asp
 - ...
- What to do with XML Schemas
 - validate documents
 - generate forms
 - generate classes (any OOP language)
 - ...

Multiple XML Schemas ?

- How to combine different XML schemas in a single file?
- Using XML namespaces



Example: A document with XHTML + MathML + SVG

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC
  "-//W3C//DTD XHTML 1.1 plus MathML 2.0 plus SVG 1.1//EN"
  "http://www.w3.org/2002/04/xhtml-math-svg/xhtml-math-svg-flat.dtd">
<html xmlns      = "http://www.w3.org/1999/xhtml"
      xmlns:svg   = "http://www.w3.org/2000/svg">

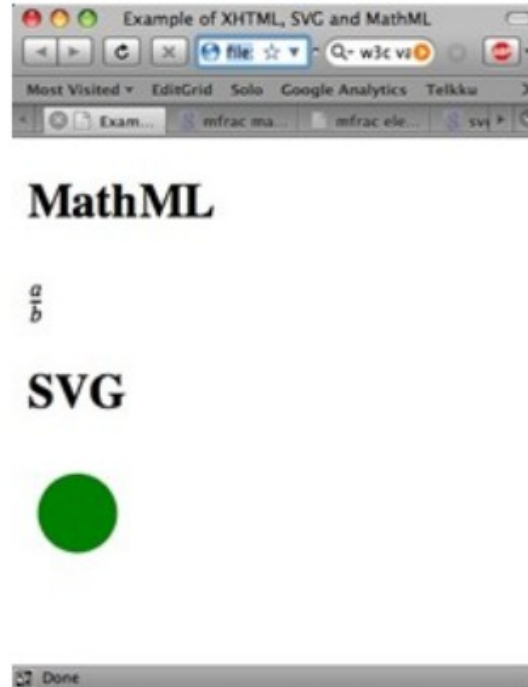
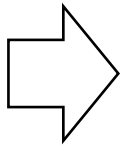
  <head>
    <title>Example of XHTML, SVG and MathML</title>
  </head>
  <body>

    <h2>MathML</h2>
    <p>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <mfrac>
          <mi>a</mi>
          <mi>b</mi>
        </mfrac>
      </math>
    </p>

    <h2>SVG</h2>

    <p>
      <svg:svg width="50px" height="50px">
        <svg:circle cx="25px" cy="25px" r="20px" fill="green"/>
      </svg:svg>
    </p>

  </body>
</html>
```



Data Interoperability and Semantics

< Part 3. Data schemas and semantics >

Part 3.1. Data schemas

Part 3.1.2. JSON Schema

ICM – Toolbox Engineering and Interoperability of Software Systems – Course unit on Data Interoperability and Semantics

M1 Cyber Physical and Social Systems – Course unit on Data Interoperability and Semantics

Maxime Lefrançois <https://maxime-lefrancois.info>

Course unit URL: <https://ci.mines-stetienne.fr/cps2/course/data>

JavaScript Object Notation

JavaScript Object Notation



Filename extension	.json
Internet media type	application/json
Type code	TEXT
Uniform Type Identifier (UTI)	public.json
Type of format	Data interchange
Extended from	JavaScript
Standard	STD 90 (RFC 8259), ECMA-404, ISO/IEC 21778:2017
Open format?	Yes
Website	json.org

```

JSON Object → {
  "company": "mycompany",
  "companycontacts": { ← Object Inside Object
    "phone": "123-123-1234",
    "email": "myemail@domain.com"
  },
  "employees": [ ← JSON Array
    {
      "id": 101,
      "name": "John",
      "contacts": [ ← Array Inside Array
        "email1@employee1.com",
        "email2@employee1.com"
      ]
    },
    {
      "id": 102, ← Number Value
      "name": "William",
      "contacts": null ← Null Value
    }
  ]
}

```

JSON Schema

JSON Schema Validator

newtonsoft.com

An online, interactive JSON Schema validator. Supports JSON Schema Draft 3, Draft 4, Draft 6 and Draft 7.

[View source code](#)

Select schema:

Empty schema

```
1 {
2   "$schema": "http://json-schema.org/draft-07/schema#",
3
4   "definitions": {
5     "person": {
6       "type": "object",
7       "properties": {
8         "name": { "type": "string" },
9         "children": {
10          "type": "array",
11          "items": { "$ref": "#/definitions/person" },
12          "default": []
13        }
14      }
15    }
16  },
17
18  "type": "object",
19
20  "properties": {
21    "person": { "$ref": "#/definitions/person" }
22  }
23 }
```

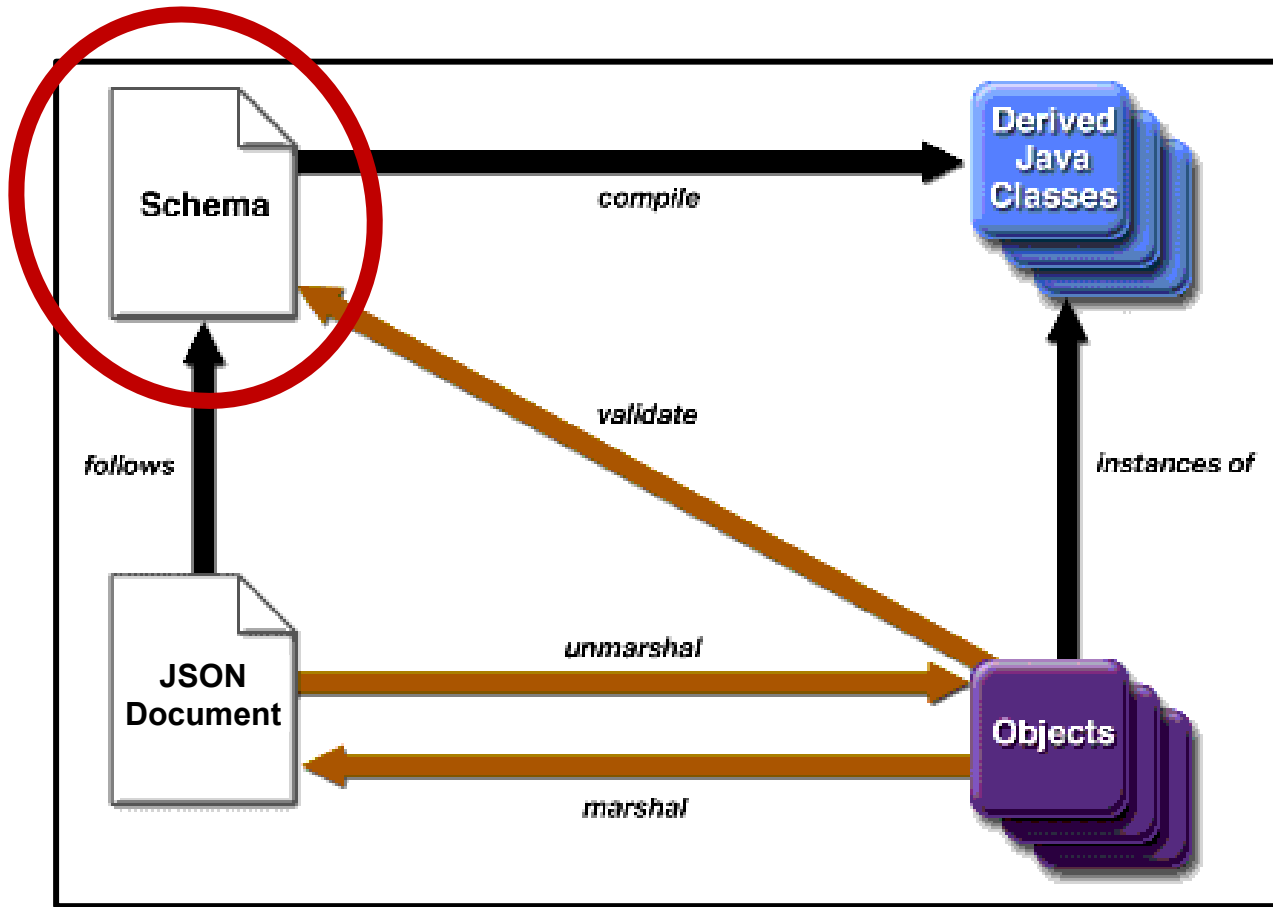


Input JSON:

```
1 {
2   "personsd fsd": {
3     "nameklk": "Elizabeth",
4     "childrensss": [
5       {
6         "nameess": "Charles",
7         "childrenss": [
8           {
9             "nameess": "William",
10            "children": [
11              { "nameesss": "George" },
12              { "name": "Charlotte" }
13            ]
14          },
15          {
16            "nameesss": "Harry"
17          }
18        ]
19      }
20    ]
21  }
22 }
```



JSON – OOP data binding



JSON-OOP data binding



JSON Schema Specification (latest)

<https://json-schema.org/draft/2020-12/json-schema-validation.html>

Table of Contents

- 1. [Introduction](#)
- 2. [Conventions and Terminology](#)
- 3. [Overview](#)
- 4. [Interoperability Considerations](#)
 - 4.1. [Validation of String Instances](#)
 - 4.2. [Validation of Numeric Instances](#)
 - 4.3. [Regular Expressions](#)
- 5. [Meta-Schema](#)
- 6. [A Vocabulary for Structural Validation](#)
 - 6.1. [Validation Keywords for Any Instance Type](#)
 - 6.1.1. [type](#)
 - 6.1.2. [enum](#)
 - 6.1.3. [const](#)
 - 6.2. [Validation Keywords for Numeric Instances \(number and integer\)](#)
 - 6.2.1. [multipleOf](#)
 - 6.2.2. [maximum](#)
 - 6.2.3. [exclusiveMaximum](#)
 - 6.2.4. [minimum](#)
 - 6.2.5. [exclusiveMinimum](#)
 - 6.3. [Validation Keywords for Strings](#)
 - 6.3.1. [maxLength](#)
 - 6.3.2. [minLength](#)
 - 6.3.3. [pattern](#)
 - 6.4. [Validation Keywords for Arrays](#)
 - 6.4.1. [maxItems](#)
 - 6.4.2. [minItems](#)
 - 6.4.3. [uniqueItems](#)
 - 6.4.4. [maxContains](#)
 - 6.4.5. [minContains](#)
 - 6.5. [Validation Keywords for Objects](#)
 - 6.5.1. [maxProperties](#)
 - 6.5.2. [minProperties](#)
 - 6.5.3. [required](#)
 - 6.5.4. [dependentRequired](#)
- 7. [Vocabularies for Semantic Content With "format"](#)
 - 7.1. [Foreword](#)
 - 7.2. [Implementation Requirements](#)
 - 7.2.1. [Format-Annotation Vocabulary](#)
 - 7.2.2. [Format-Assertion Vocabulary](#)
 - 7.2.3. [Custom format attributes](#)
 - 7.3. [Defined Formats](#)
 - 7.3.1. [Dates, Times, and Duration](#)
 - 7.3.2. [Email Addresses](#)
 - 7.3.3. [Hostnames](#)
 - 7.3.4. [IP Addresses](#)
 - 7.3.5. [Resource Identifiers](#)
 - 7.3.6. [uri-template](#)
 - 7.3.7. [JSON Pointers](#)
 - 7.3.8. [regex](#)
- 8. [A Vocabulary for the Contents of String-Encoded Data](#)
 - 8.1. [Foreword](#)
 - 8.2. [Implementation Requirements](#)
 - 8.3. [contentEncoding](#)
 - 8.4. [contentMediaType](#)
 - 8.5. [contentSchema](#)
 - 8.6. [Example](#)
- 9. [A Vocabulary for Basic Meta-Data Annotations](#)
 - 9.1. ["title" and "description"](#)
 - 9.2. ["default"](#)
 - 9.3. ["deprecated"](#)
 - 9.4. ["readOnly" and "writeOnly"](#)
 - 9.5. ["examples"](#)
- 10. [Security Considerations](#)
- 11. [References](#)
 - 11.1. [Normative References](#)
 - 11.2. [Informative References](#)
- [Appendix A. Keywords Moved from Validation to Core](#)
- [Appendix B. Acknowledgments](#)
- [Appendix C. ChangeLog](#)
- [Authors' Addresses](#)

JSON Schema

Examples:

- <https://json-schema.org/learn/miscellaneous-examples.html>
- <https://www.fiware.org/developers/data-models/>
- <https://oneiota.org/>

FIWARE/data-models



Code and specifications to support harmonized data models

36

Contributors

3

Issues

83

Stars

100

Forks

openconnectivityfoundation/
IoTDataModels



Repository for data models found in www.oneiota.org

20

Contributors

0

Issues

9

Stars

8

Forks

JSON Schema

```
{
  "$schema": "https://json-schema.org/draft/2019-09/schema#",
  "$id": "http://my-paintings-api.com/schemas/painting-schema.json",
  "type": "object",
  "title": "Painting",
  "description": "Painting information",
  "additionalProperties": true,
  "required": ["name", "artist", "dimension", "description", "tags"],
  "properties": {
    "name": {
      "type": "string",
      "description": "Painting name"
    },
    "artist": {
      "type": "string",
      "maxLength": 50,
      "description": "Name of the artist"
    },
    "description": {
      "type": ["string", "null"],
      "description": "Painting description"
    },
    "dimension": {
      "$ref": "#/$defs/dimension"
    },
    "tags": {
      "type": "array",
      "items": {
        "$ref": "#/$defs/tag"
      }
    }
  },
  "$defs": {
    "tag": {
      "type": "string",
      "enum": ["oil", "watercolor", "digital", "famous"]
    },
    "dimension": {
      "type": "object",
      "title": "Painting dimension",
      "description": "Describes the dimension of a painting in cm",
      "additionalProperties": true,
      "required": ["width", "height"],
      "properties": {
        "width": {
          "type": "number",
          "description": "Width of the product",
          "minimum": 1
        },
        "height": {
          "type": "number",
          "description": "Height of the product",
          "minimum": 1
        }
      }
    }
  }
}
```

JSON

```
{
  "name": "Mona Lisa",
  "artist": "Leonardo da Vinci",
  "description": null,
  "dimension": {
    "height": 53.0,
    "width": 77.0
  },
  "tags": ["oil", "famous"]
}
```

Validation does not fall if undefined properties are given

Required root level properties

max 50 characters allowed

string or null allowed

Only predefined values allowed

Only positive numbers allowed

Data Interoperability and Semantics

< Part 3. Data schemas and semantics >

Part 3.2. Semantics

ICM – Toolbox Engineering and Interoperability of Software Systems – Course unit on Data Interoperability and Semantics

M1 Cyber Physical and Social Systems – Course unit on Data Interoperability and Semantics

Maxime Lefrançois <https://maxime-lefrancois.info>

Course unit URL: <https://ci.mines-stetienne.fr/cps2/course/data>

Data Interoperability and Semantics

< Part 3. Data schemas and semantics >

Part 3.2. Semantics

Part 3.2.1. Heterogeneities and data conflicts

ICM – Toolbox Engineering and Interoperability of Software Systems – Course unit on Data Interoperability and Semantics

M1 Cyber Physical and Social Systems – Course unit on Data Interoperability and Semantics

Maxime Lefrançois <https://maxime-lefrancois.info>

Course unit URL: <https://ci.mines-stetienne.fr/cps2/course/data>

DEMO: Compare JSON documents

<https://samples.openweathermap.org/data/2.5/weather?id=2172797&appid=b6907d289e10d714a6e88b30761fae22>

and

<https://www.prevision-meteo.ch/services/json/lausanne>

and

<https://github.com/smart-data-models/dataModel.Weather/blob/master/WeatherForecast/examples/example-normalized.json>

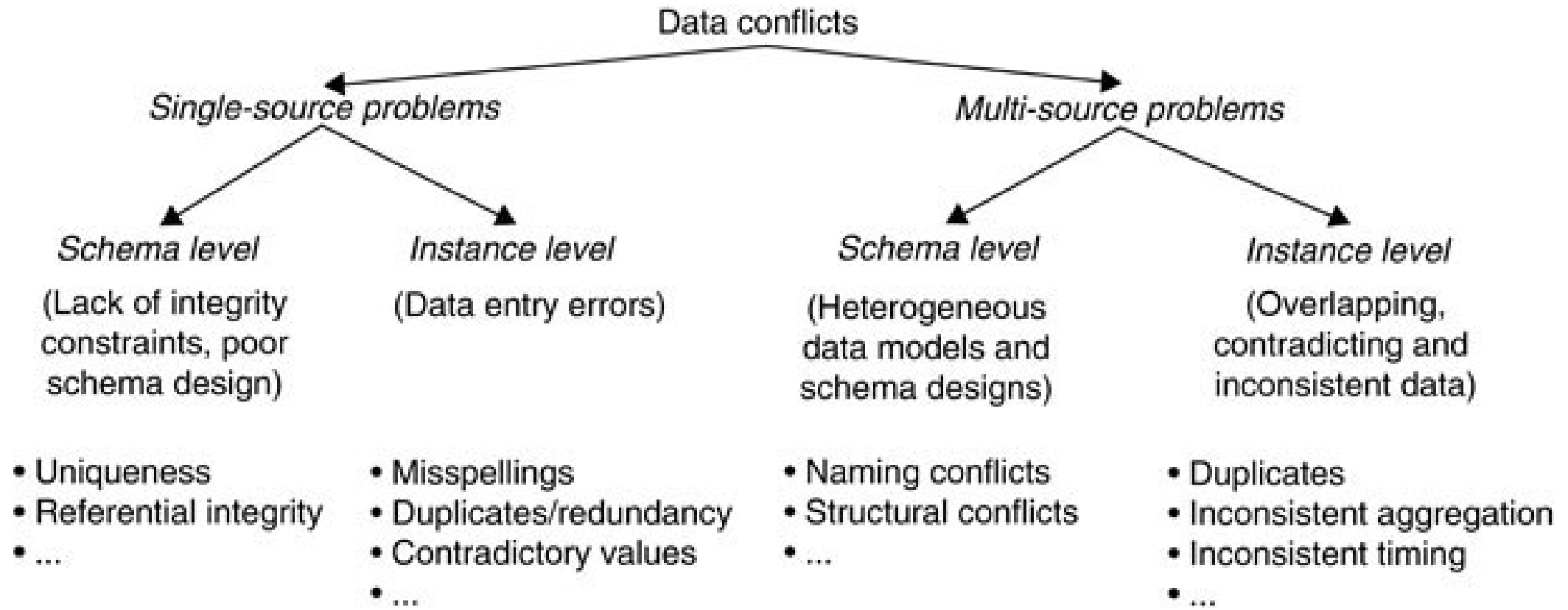
DEMO: Compare JSON documents

Different modeling choices were made, which make these two services completely non-interoperable:

- the lat/long coordinates: string vs number
- the UNIX timestamps vs dates and times
- the choice of keys and the semantics (meaning) of the values
- the units of temperature, pressure, wind speed, ...
- the semantics of wind direction
- the value for "icon": "03n" - (if we follow our nose on the website, we may figure out it refers to <http://openweathermap.org/img/w/03n.png>)
- the country codes ISO 3166-1 ALPHA-2 and ISO 3166-1 ALPHA-3 (example of Australia and Austria)

Data conflicts

*Data conflicts are **deviations between data intended to capture the same state of a real-world entity**. Data with conflicts are often called “dirty” data and can mislead analysis performed on it.*



Data Interoperability and Semantics

< Part 3. Data schemas and semantics >

Part 3.2. Semantics

Part 3.2.2. Controlled vocabularies, thesauri, taxonomies

ICM – Toolbox Engineering and Interoperability of Software Systems – Course unit on Data Interoperability and Semantics

M1 Cyber Physical and Social Systems – Course unit on Data Interoperability and Semantics

Maxime Lefrançois <https://maxime-lefrancois.info>

Course unit URL: <https://ci.mines-stetienne.fr/cps2/course/data>

Controlled vocabularies

... an established list of standardized terminology for use in indexing and retrieval of information

— OECD

... an organized arrangement of words and phrases used to index content and/or retrieve content through browsing or searching

— Getty institute

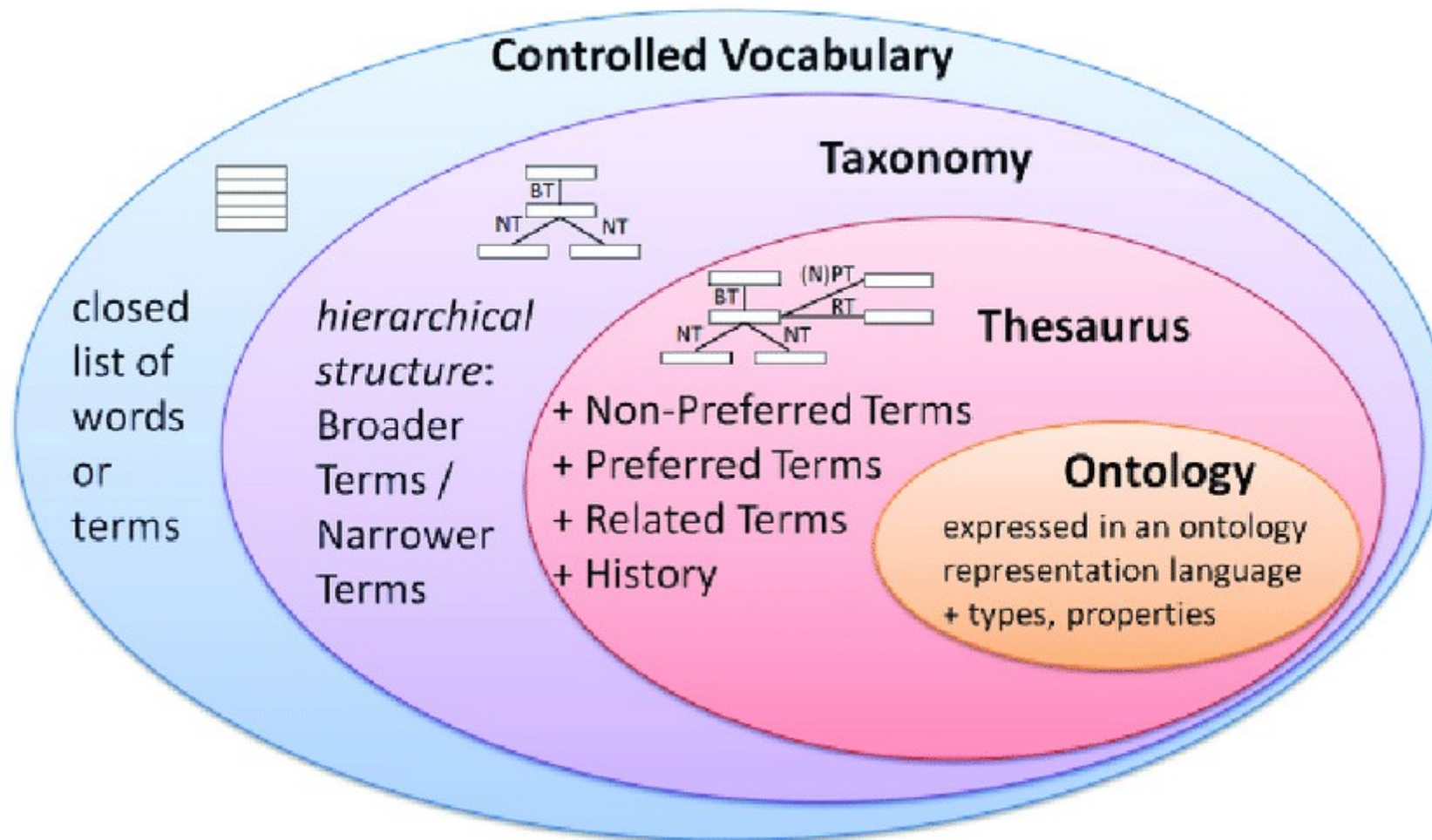
... an standardized – yet dynamic – set of terms and phrases authorized for use in an indexing system to describe a subject area or information domain

— SCIP

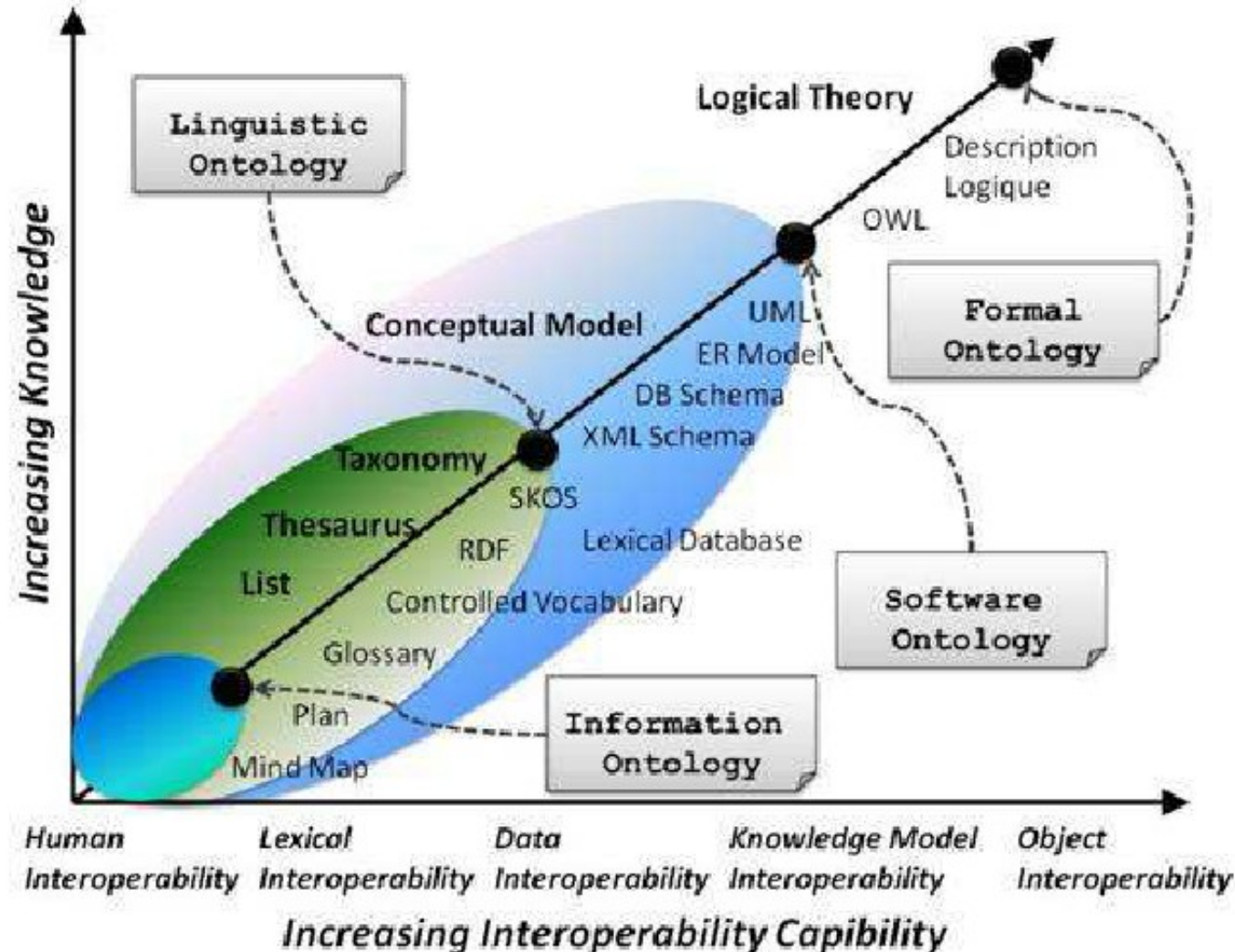
vocabulary for which the entries, i.e. **definition/term** pairs, are controlled by a **Source Authority** based on a **rulebase** and **process** for addition/deletion of entries

— ISO/IEC 15944-5:2008(en)
Information technology — Business Operational View

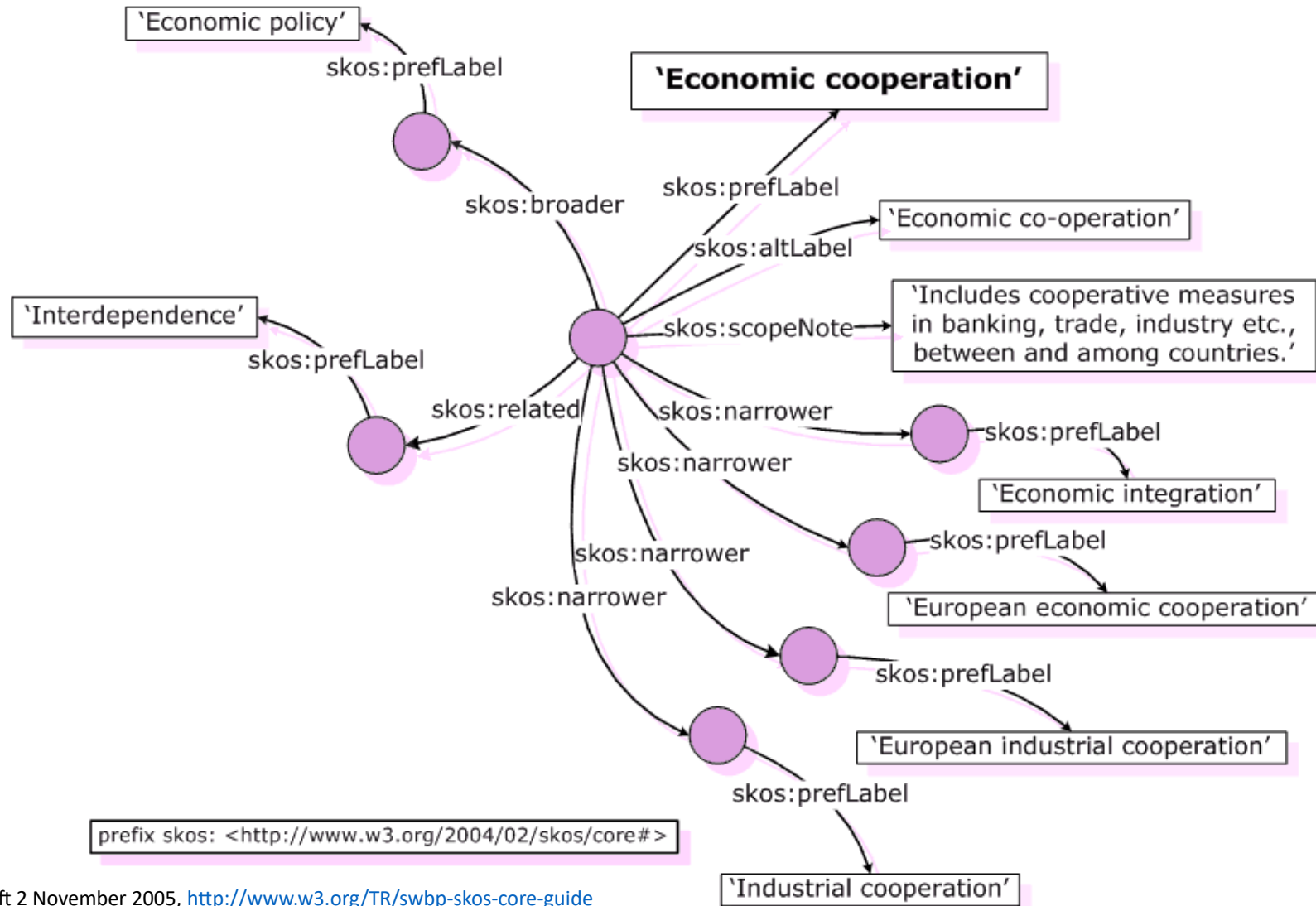
Controlled vocabulary, taxonomy, thesaurus, ontology



Controlled vocabulary, taxonomy, thesaurus, ontology



SKOS Simple Knowledge Organization System



Examples of thesauri, taxonomies, ...

UNESCO Thesaurus



The UNESCO Thesaurus is a **controlled and structured list of terms used in subject analysis and retrieval of documents and publications** in the fields of education, culture, natural sciences, social and human sciences, communication and information.

<https://skos.um.es> › unescothes

UNESCO Thesaurus - SKOS

GEMET

General
Multilingual
Environmental
Thesaurus

<https://www.eionet.europa.eu/gemet/en/webservices/>

nasa/dictionaries

A collection of NASA "dictionaries", including thesauri, taxonomies and ontologies.



2 Contributors 2 Issues 51 Stars 31 Forks

<https://github.com/nasa/dictionaries>

Project Haystack

Using Standard Markup for...

Site

Equipment

Point

```
id: @2180b666-430b2363
site
dis: 'Gaithersburg'
geoAddr: '18212 Montgomery Village Ave,
Gaithersburg, MD'
geoCity: 'Gaithersburg'
geoCoord: C(39.154824,-77.209002)
geoCountry: 'US'
geoPostalCode: '20879'
geoState: 'MD'
tz: 'New_York'
```

```
id: @2180b666-7032054c
equip
dis: 'RTU-1'
ahu
hvac:
rooftop
siteRef: @2180b666-430b2363
elecMeterRef: @2180b666-7032054d
```

→ To elecMeter entity

```
id: @218a0616-0b5e382b
point
dis: 'Discharge air temperature'
sensor
air
temp
discharge
his
unit: 'F'
equipRef: @2180b666-7032054c
siteRef: @2180b666-430b2363
```

<https://project-haystack.org/>

Data Interoperability and Semantics

< Part 3. Data schemas and semantics >

Part 3.2. Semantics

Part 3.2.3. Resource Description Framework

ICM – Toolbox Engineering and Interoperability of Software Systems – Course unit on Data Interoperability and Semantics

M1 Cyber Physical and Social Systems – Course unit on Data Interoperability and Semantics

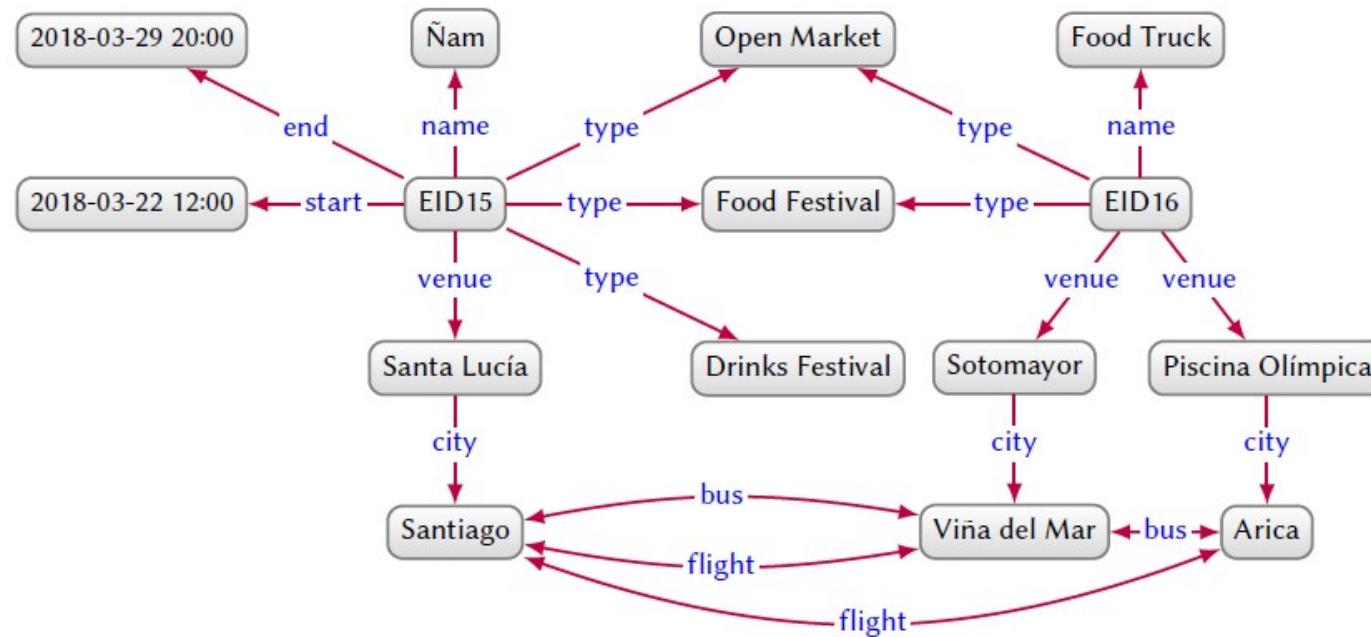
Maxime Lefrançois <https://maxime-lefrancois.info>

Course unit URL: <https://ci.mines-stetienne.fr/cps2/course/data>

RDF- Resource Description Framework

RDF

Resource Description Framework	
Abbreviation	RDF
Status	Published, W3C Recommendation
Year started	1996; 25 years ago
First published	February 28, 1996; 25 years ago
Latest version	1.1 (Recommendation) February 25, 2014; 7 years ago
Organization	World Wide Web Consortium (W3C)
Committee	RDF Working Group
Editors	Richard Cyganiak, David Wood, Markus Lanthaler
Base standards	URI
Related standards	RDFS, OWL, RIF, RDFa
Domain	Semantic Web
Website	www.w3.org/TR/rdf11-concepts/



An RDF Graph

Vertices:

Resources: Things. Uniquely identified by URIs

Literals: Unicode string + Datatype identified by URI. *out-degree=0*

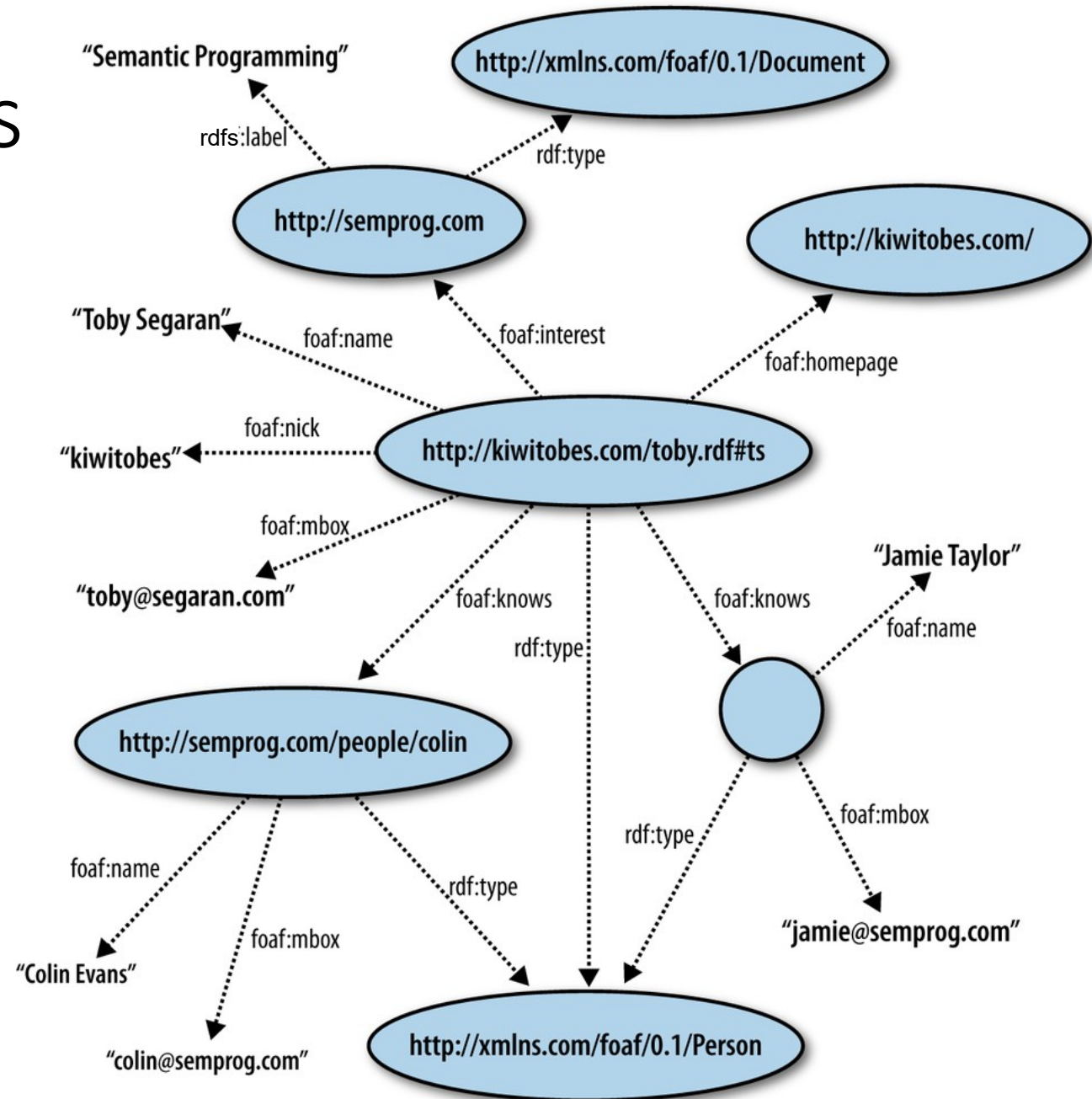
Edges:

Relationships: connections between vertices. Uniquely identified by URIs



CURIE = Compact URIs

```
@prefix rdf:
  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs:
  <http://www.w3.org/2000/01/rdf-schema#>.
@prefix foaf:
  <http://xmlns.com/foaf/0.1/>.
@prefix schema:
  <http://schema.org/>.
...
```



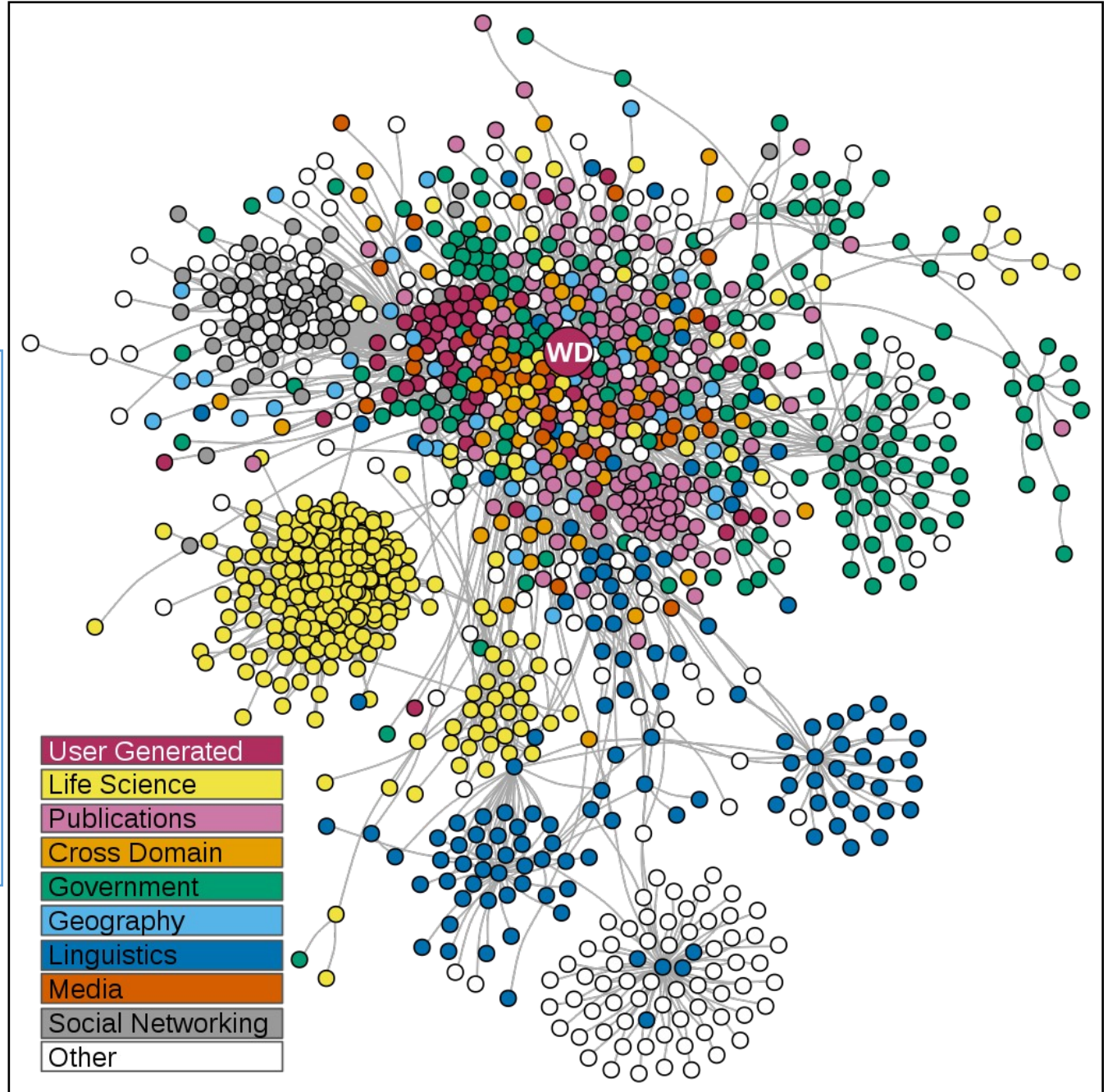
The Linked Data

It's possible to use URIs defined in an external graph

- to use a reference identification system
- to augment a graph
anyone can say anything about anything

One can specify vertice co-reference with some special edges: owl:sameAs from W3C OWL vocabulary

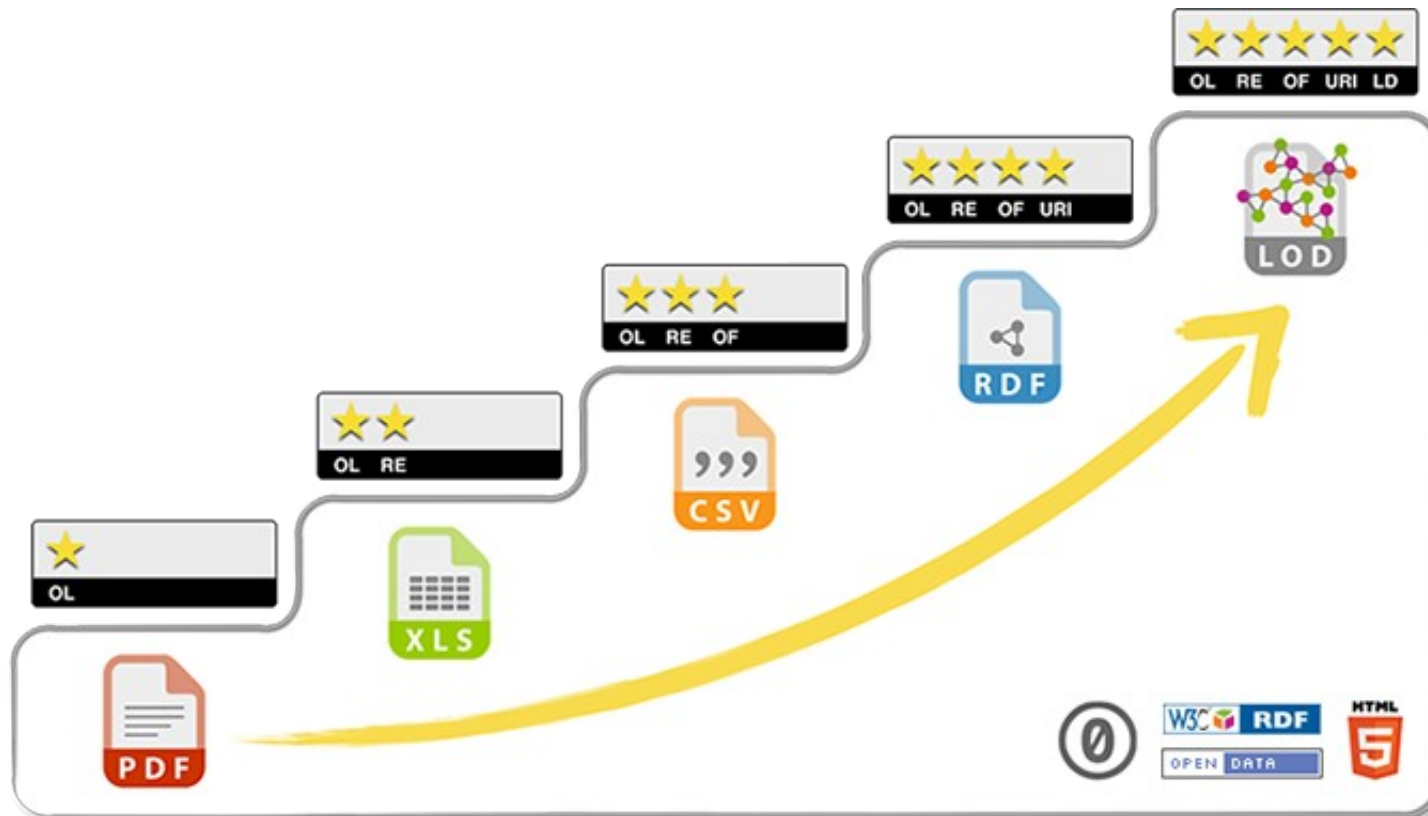
chile:Santiago - owl:sameAs → geo:SantiagoDeChile



Linked Open Data cloud visualisation: each node is a RDF dataset. Links indicate the existence of external identification links. Source: wikipedia

5 ★ OPEN DATA

[Tim Berners-Lee](#), the inventor of the Web and Linked Data initiator, suggested a [5-star deployment scheme](#) for Open Data.



<https://5stardata.info/en/>

Examples of RDF vocabularies & ontologies

The logo for schema.org, featuring the text "schema.org" in white on a red background.

Over 10 million sites use Schema.org to markup their web pages and email messages. Many applications from Google, Microsoft, Pinterest, Yandex and others already use these vocabularies to power rich, extensible experiences.

Documentation

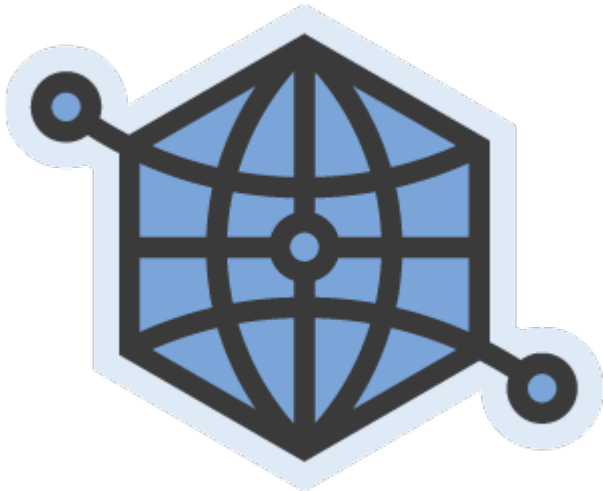
Here is some of the documentation available on this site:

- [Getting Started](#): A simple introduction to microdata and using schema.org for marking up your site.
- [Schemas](#): The actual schemas, arranged in a hierarchy, with a page for each item in the schema.
- [The full type hierarchy](#): The full type hierarchy, in a single file.
- [Frequently asked questions](#)
- [Data model](#): a brief note on the data model used, etc.
- [Extension Mechanism](#): The extension mechanism that can be used to extend the schemas
- [Schema.org Discussion Group](#): Forum for finding answers to questions, etc.
- [Feedback form](#): Please give us feedback, report bugs, etc.

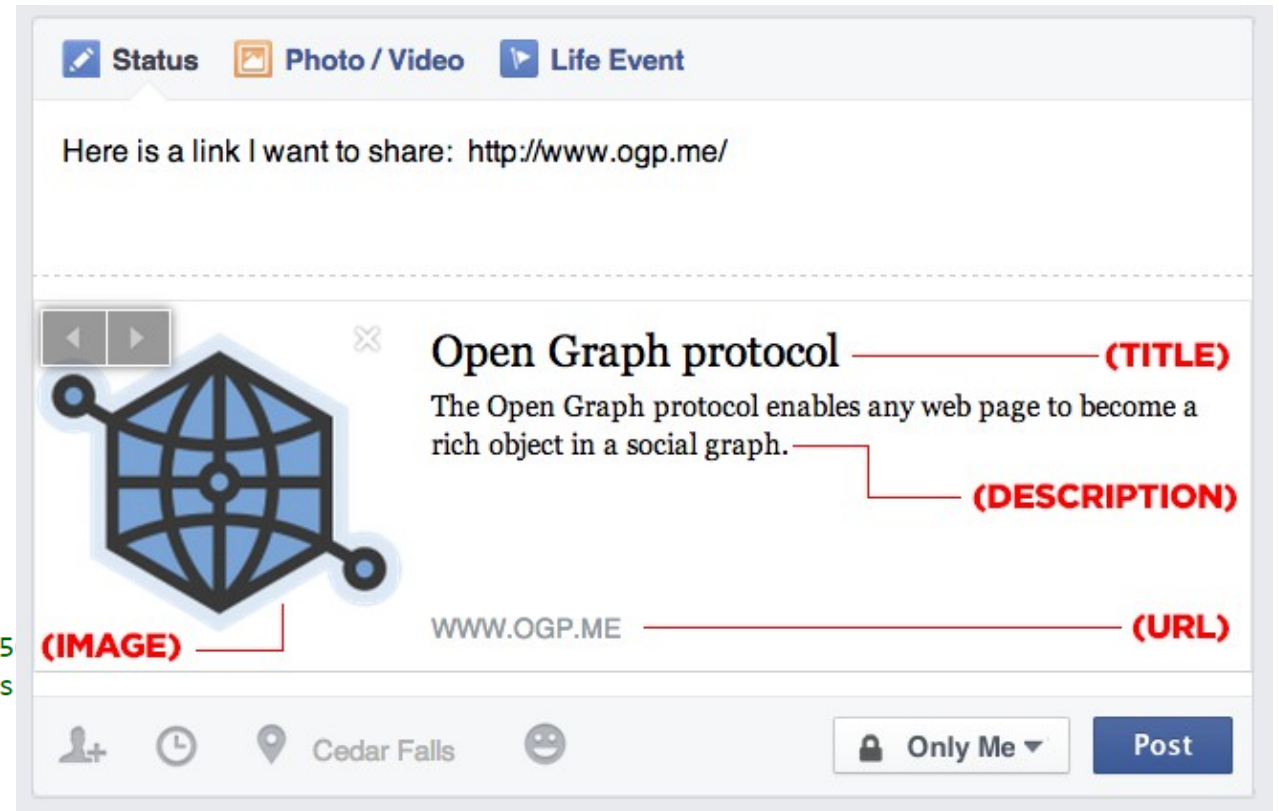
Founded by Google, Microsoft, Yahoo and Yandex, 2011

Test with <https://validator.schema.org/> or <https://developers.google.com/search/docs/advanced/structured-data>

Examples of RDF vocabularies & ontologies



```
<html prefix="og: https://ogp.me/ns#">
<head>
<title>The Rock (1996)</title>
<meta property="og:title" content="The Rock" />
<meta property="og:type" content="video.movie" />
<meta property="og:url" content="https://www.imdb.com/title/tt01175
<meta property="og:image" content="https://ia.media-imdb.com/images
...
</head>
...
</html>
```



Open Graph Protocol <https://ogp.me/> (Facebook, 2010), test with <https://developers.facebook.com/tools/debug/>

Data Interoperability and Semantics

< Part 3. Data schemas and semantics >

Part 3.2. Semantics

Part 3.2.4. RDFa: Rich structured data markup for web documents

ICM – Toolbox Engineering and Interoperability of Software Systems – Course unit on Data Interoperability and Semantics

M1 Cyber Physical and Social Systems – Course unit on Data Interoperability and Semantics

Maxime Lefrançois <https://maxime-lefrancois.info>

Course unit URL: <https://ci.mines-stetienne.fr/cps2/course/data>

RDFa Resource Description Framework in Attributes

See RDFa 1.1 Primer - Rich Structured Data Markup for Web Documents
- <https://www.w3.org/TR/rdfa-primer/>

```
<html prefix="og: https://ogp.me/ns#">
<head>
<title>The Rock (1996)</title>
<meta property="og:title" content="The Rock" />
<meta property="og:type" content="video.movie" />
<meta property="og:url" content="https://www.imdb.com/title/tt0117500/" />
<meta property="og:image" content="https://ia.media-imdb.com/images/rock.jpg" />
...
</head>
...
</html>
```

```
<div typeof="foaf:Person" xmlns:foaf="http://xmlns.com/foaf/0.1/">
  <p property="foaf:name">
    Alice Birpemsrick
  </p>
  <p>
    Email: <a rel="foaf:mbox"
href="mailto:alice@example.com">alice@example.com</a>
  </p>
  <p>
    Phone: <a rel="foaf:phone" href="tel:
+1-617-555-7332">+1 617.555.7332</a>
  </p>
</div>
```

RDFa Resource Description Framework in Attributes

See RDFa 1.1 Primer - Rich Structured Data Markup for Web Documents
- <https://www.w3.org/TR/rdfa-primer/>

The Goal

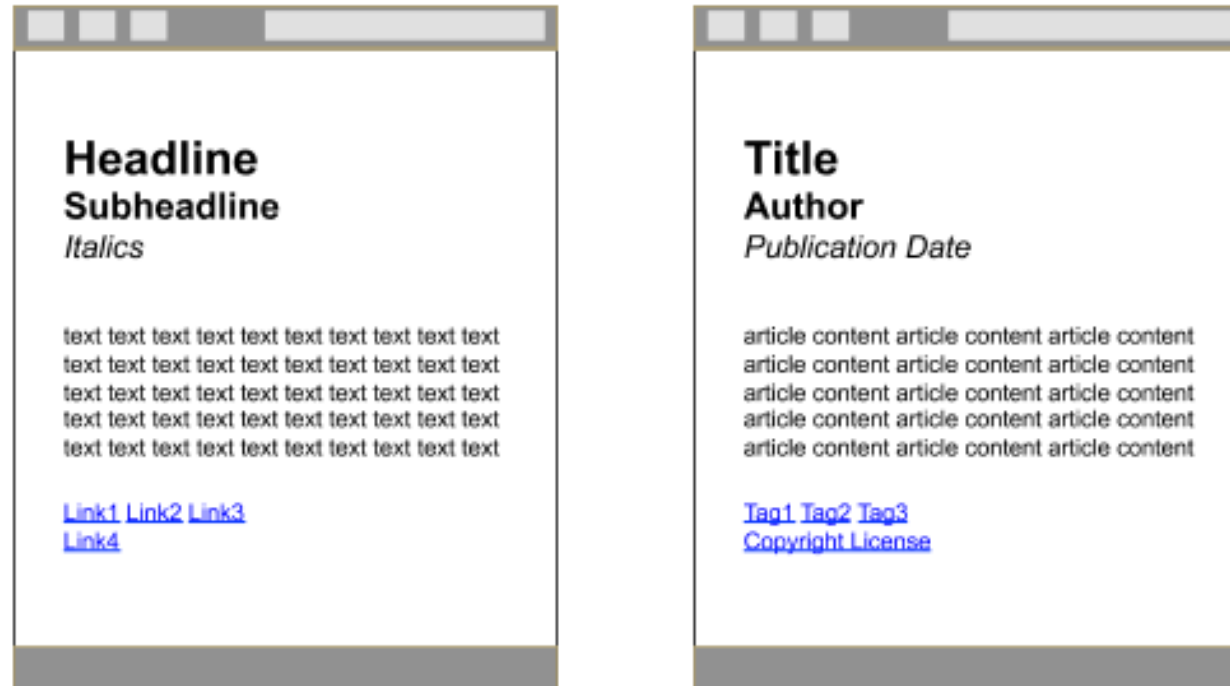
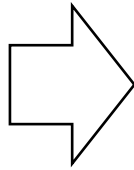


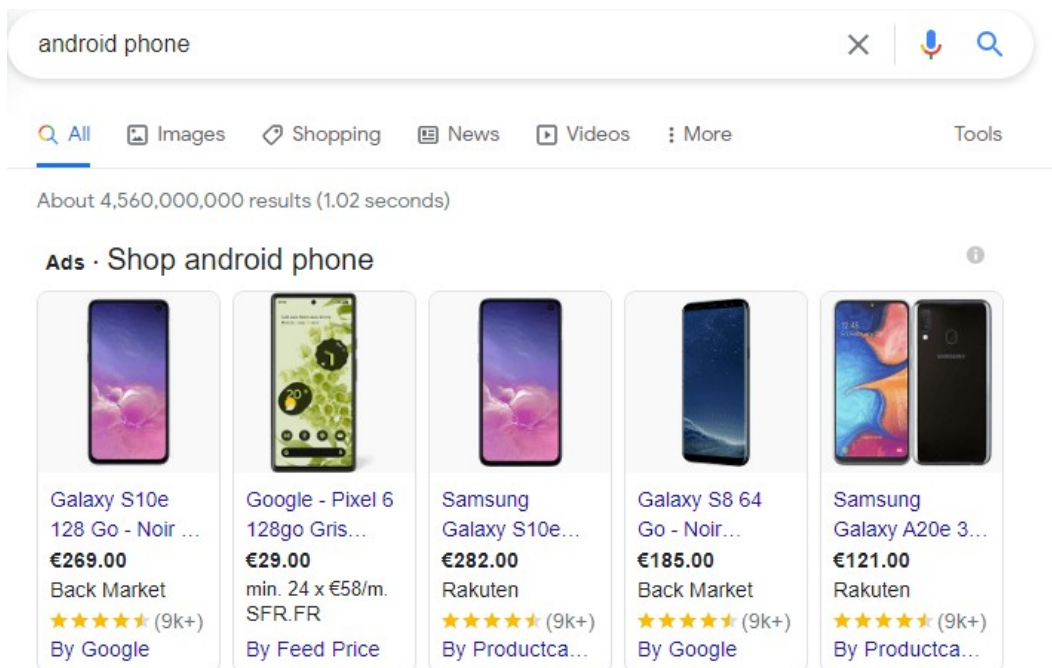
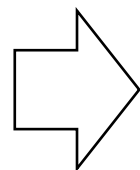
Figure 1: On the left, what browsers see. On the right, what humans see. Can we bridge the gap so that browsers see more of what we see?

RDFa Resource Description Framework in Attributes

See RDFa 1.1 Primer - Rich Structured Data Markup for Web Documents

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

The Result



Google can extract structured data from e.g., e-commerce websites, and “understand” its meaning

Data Interoperability and Semantics

< Part 3. Data schemas and semantics >

Part 3.2. Semantics

Part 3.2.5. JSON-LD: JSON for Linking Data

ICM – Toolbox Engineering and Interoperability of Software Systems – Course unit on Data Interoperability and Semantics

M1 Cyber Physical and Social Systems – Course unit on Data Interoperability and Semantics

Maxime Lefrançois <https://maxime-lefrancois.info>

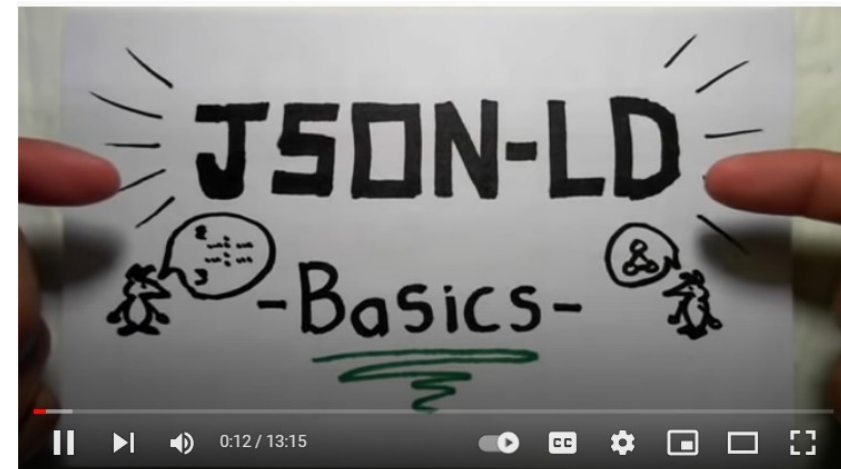
Course unit URL: <https://ci.mines-stetienne.fr/cps2/course/data>

JavaScript Object Notation for Linked Data

See JSON-LD 1.1 - A JSON-based Serialization for Linked Data - <https://www.w3.org/TR/json-ld11/>

JSON-LD

Filename extension	.jsonld
Internet media type	application/ld+json
Type of format	Semantic Web
Container for	Linked Data
Extended from	JSON
Standard	JSON-LD 1.1 ↗ / JSON-LD 1.1 API ↗
Open format?	Yes



What is JSON-LD?

127,225 views • Jun 19, 2012

1.8K 1.8K DISLIKE SHARE SAVE ...

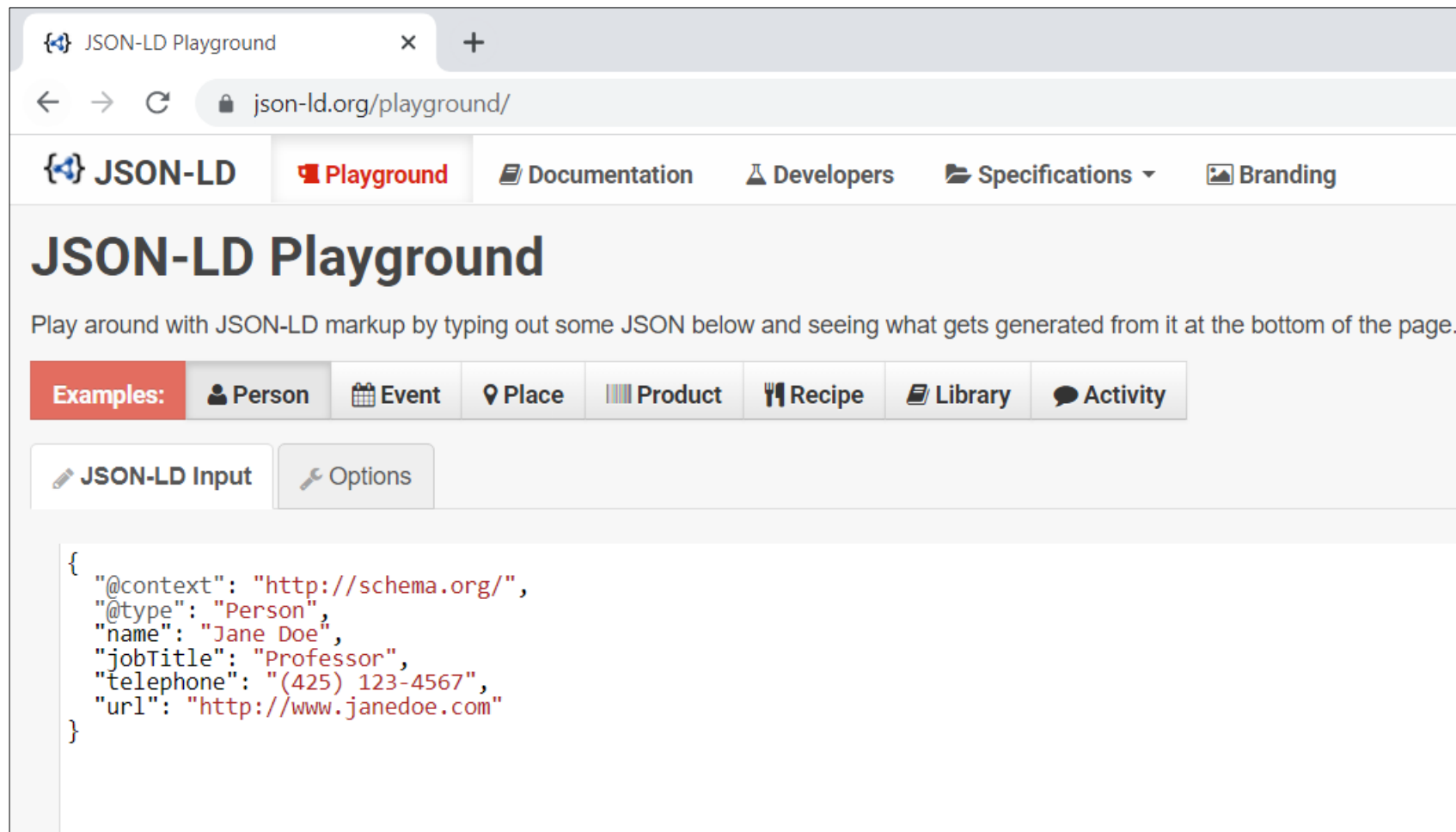


Manu Sporny
3.51K subscribers

SUBSCRIBE

<https://json-ld.org/learn.html>

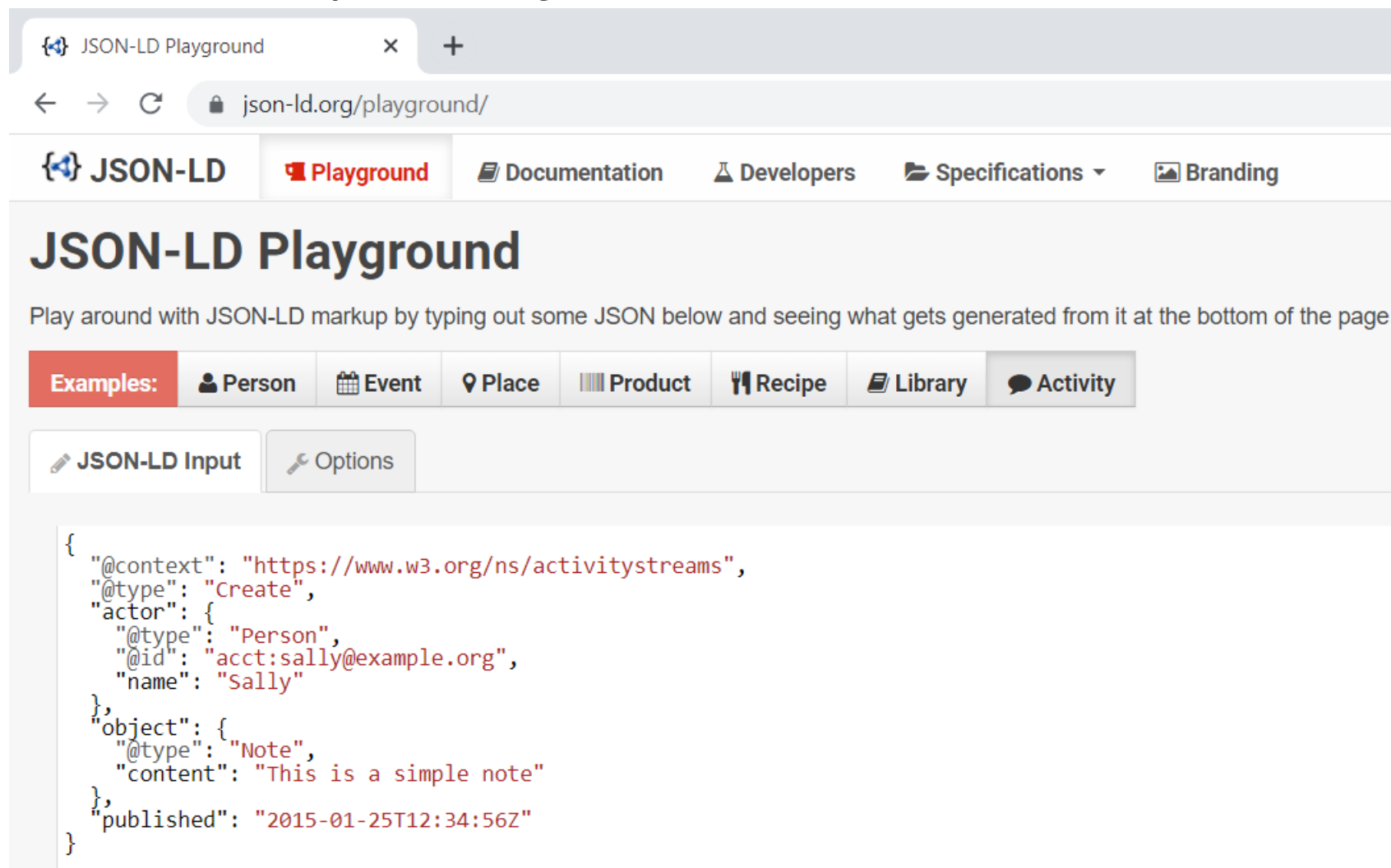
JavaScript Object Notation for Linked Data



The screenshot shows the JSON-LD Playground interface. The browser tab is titled "JSON-LD Playground" and the address bar shows "json-ld.org/playground/". The navigation bar includes links for "JSON-LD", "Playground" (active), "Documentation", "Developers", "Specifications", and "Branding". The main heading is "JSON-LD Playground". Below it, a text prompt says: "Play around with JSON-LD markup by typing out some JSON below and seeing what gets generated from it at the bottom of the page." A row of example buttons includes "Examples:", "Person" (selected), "Event", "Place", "Product", "Recipe", "Library", and "Activity". Below the buttons are two tabs: "JSON-LD Input" (active) and "Options". The input area contains the following JSON-LD code:

```
{
  "@context": "http://schema.org/",
  "@type": "Person",
  "name": "Jane Doe",
  "jobTitle": "Professor",
  "telephone": "(425) 123-4567",
  "url": "http://www.janedoe.com"
}
```

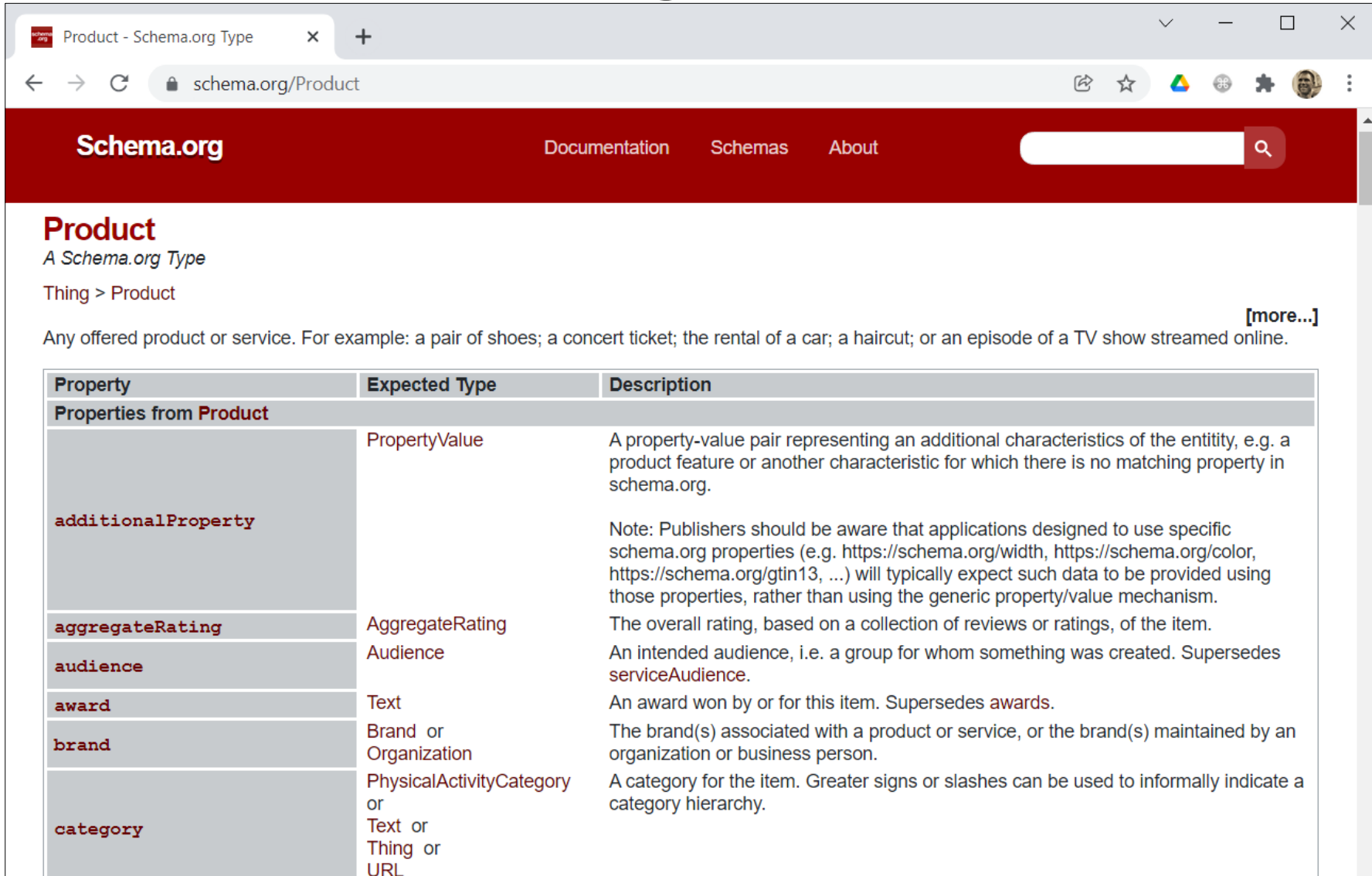
JavaScript Object Notation for Linked Data



The screenshot shows the JSON-LD Playground interface. At the top, there's a browser tab labeled 'JSON-LD Playground' and a URL bar with 'json-ld.org/playground/'. The navigation bar includes links for 'JSON-LD', 'Playground' (active), 'Documentation', 'Developers', 'Specifications', and 'Branding'. Below the navigation bar, the title 'JSON-LD Playground' is displayed, followed by a description: 'Play around with JSON-LD markup by typing out some JSON below and seeing what gets generated from it at the bottom of the page.' A row of buttons for 'Examples:' includes 'Person', 'Event', 'Place', 'Product', 'Recipe', 'Library', and 'Activity'. Below these, there are two tabs: 'JSON-LD Input' (active) and 'Options'. The main content area shows a JSON-LD input example:

```
{
  "@context": "https://www.w3.org/ns/activitystreams",
  "@type": "Create",
  "actor": {
    "@type": "Person",
    "@id": "acct:sally@example.org",
    "name": "Sally"
  },
  "object": {
    "@type": "Note",
    "content": "This is a simple note"
  },
  "published": "2015-01-25T12:34:56Z"
}
```

JSON-LD and schema.org

A screenshot of a web browser displaying the Schema.org 'Product' page. The browser's address bar shows 'schema.org/Product'. The page has a red header with the 'Schema.org' logo and navigation links for 'Documentation', 'Schemas', and 'About'. A search bar is located on the right side of the header. The main content area is titled 'Product' and includes the subtitle 'A Schema.org Type' and a breadcrumb 'Thing > Product'. A description of 'Product' is provided, followed by a table of properties. The table has three columns: 'Property', 'Expected Type', and 'Description'. The properties listed are 'additionalProperty', 'aggregateRating', 'audience', 'award', 'brand', and 'category'. Each property row includes its name, the expected data type, and a detailed description of its use and any supersedes relationships.

Product - Schema.org Type

schema.org/Product

Schema.org

Documentation Schemas About

Product

A Schema.org Type

Thing > Product

[more...]

Any offered product or service. For example: a pair of shoes; a concert ticket; the rental of a car; a haircut; or an episode of a TV show streamed online.

Property	Expected Type	Description
Properties from Product		
additionalProperty	PropertyValue	A property-value pair representing an additional characteristics of the entity, e.g. a product feature or another characteristic for which there is no matching property in schema.org. Note: Publishers should be aware that applications designed to use specific schema.org properties (e.g. https://schema.org/width , https://schema.org/color , https://schema.org/gtin13 , ...) will typically expect such data to be provided using those properties, rather than using the generic property/value mechanism.
aggregateRating	AggregateRating	The overall rating, based on a collection of reviews or ratings, of the item.
audience	Audience	An intended audience, i.e. a group for whom something was created. Supersedes serviceAudience .
award	Text	An award won by or for this item. Supersedes awards .
brand	Brand or Organization	The brand(s) associated with a product or service, or the brand(s) maintained by an organization or business person.
category	PhysicalActivityCategory or Text or Thing or URL	A category for the item. Greater signs or slashes can be used to informally indicate a category hierarchy.

Product - Schema.org Type

+

schema.org/Product

🔗

☆

🌐

⚙️

👤

⋮

Examples

Example 1 

No Markup

Microdata

RDFa

JSON-LD

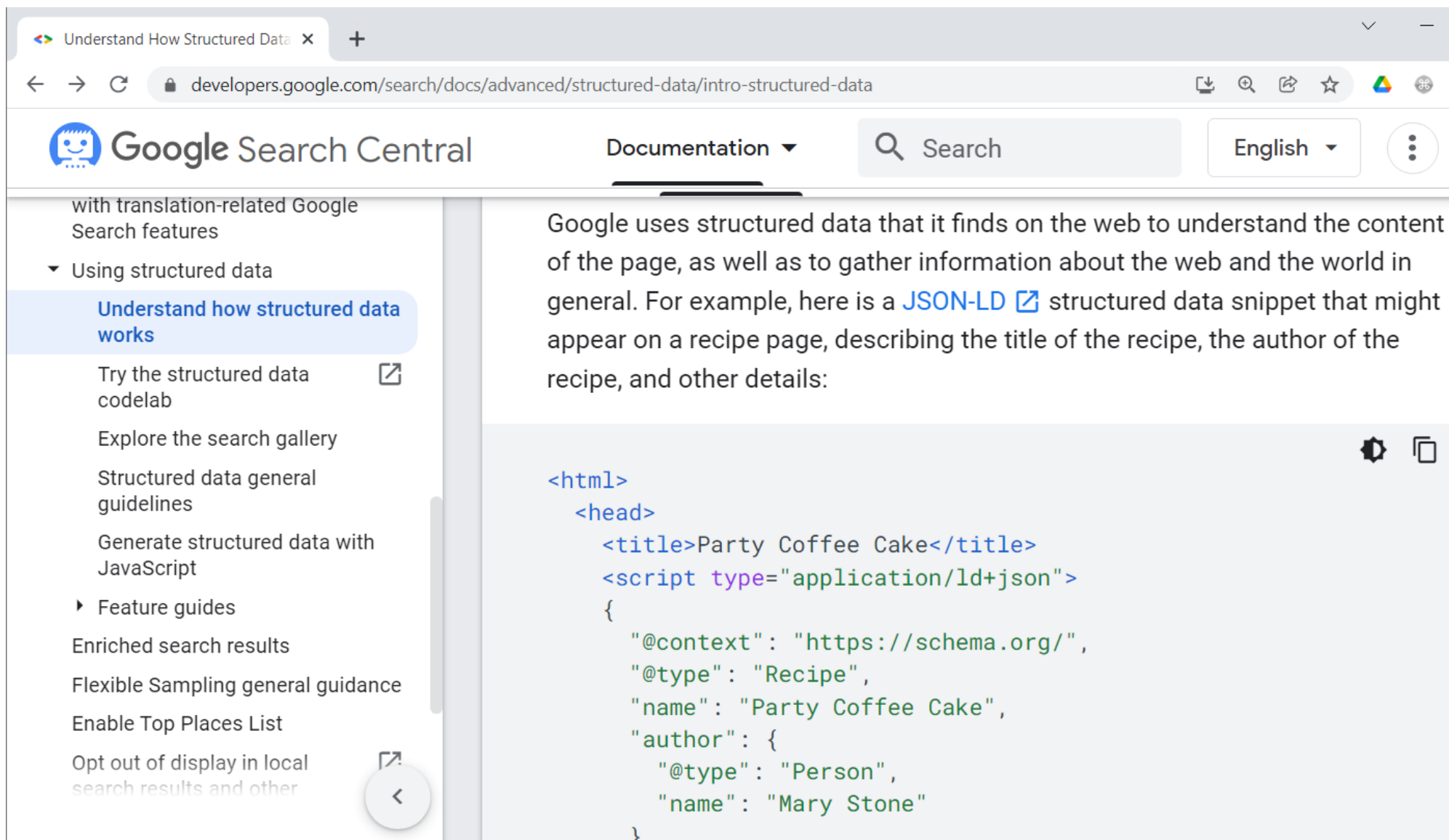
Structure

Example encoded as JSON-LD in a HTML script tag.

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "Product",
  "aggregateRating": {
    "@type": "AggregateRating",
    "ratingValue": "3.5",
    "reviewCount": "11"
  },
  "description": "0.7 cubic feet countertop microwave. Has six preset cooking categories and co
  "name": "Kenmore White 17\" Microwave",
  "image": "kenmore-microwave-17in.jpg",
  "offers": {
    "@type": "Offer",
    "availability": "https://schema.org/InStock",
    "price": "55.00",
    "priceCurrency": "USD"
  },
  "review": [
    {
      "@type": "Review",
      "author": "Ellie",
      "datePublished": "2011-04-01",
      "reviewBody": "The lamp burned out and now I have to replace it.",
      "name": "Not a happy camper".
```

47

JSON-LD and Google



The screenshot shows the Google Search Central documentation page for structured data. The browser address bar displays the URL: `developers.google.com/search/docs/advanced/structured-data/intro-structured-data`. The page header includes the Google Search Central logo, a 'Documentation' dropdown menu, a search bar, and a language selector set to 'English'.

The left sidebar contains a navigation menu with the following items:

- with translation-related Google Search features
- Using structured data
 - Understand how structured data works** (highlighted)
 - Try the structured data codelab
 - Explore the search gallery
 - Structured data general guidelines
 - Generate structured data with JavaScript
- Feature guides
 - Enriched search results
 - Flexible Sampling general guidance
 - Enable Top Places List
 - Opt out of display in local search results and other

The main content area contains the following text:

Google uses structured data that it finds on the web to understand the content of the page, as well as to gather information about the web and the world in general. For example, here is a [JSON-LD](#) structured data snippet that might appear on a recipe page, describing the title of the recipe, the author of the recipe, and other details:

```
<html>
<head>
  <title>Party Coffee Cake</title>
  <script type="application/ld+json">
  {
    "@context": "https://schema.org/",
    "@type": "Recipe",
    "name": "Party Coffee Cake",
    "author": {
      "@type": "Person",
      "name": "Mary Stone"
    }
  }
</script>
</head>
</html>
```


JSON-LD and Bing

Mes sites

Tableau de bord

> Configurer mon site

Sitemaps

Suggérer des URL

Ignorer les paramètres d'URL

Contrôle de l'analyse

Liens profonds

Bloquer des URL

Désavouer des liens

Ciblage géographique

Vérifier la possession du site

Pages connectées

Utilisateurs

> Rapports et données

Trafic sur vos pages

Adéquation à l'utilisation sur appareil mobile de la page

Explorateur d'indexation

Recherche par mots clés

Rapports SEO

Liens entrants

Information sur l'analyse



Site

www.internet-formation.fr

Valdateur de balisage

Saisissez un lien URL et notre outil vous informera sur tout le balisage structuré trouvé sur la page. Ce balisage sera ensuite probablement utilisé pour afficher des extraits enrichis de vos URL dans Bing.

VALIDER

Ressources

[HTML Microdata](#)

[RDFa](#)

[Open Graph](#)

[Microformats](#)

[Schema.org](#)

[Json-LD](#)

Json-LD

```
{
  "@context": "http://schema.org",
  "@type": "Organization",
  "name": "Internet-Formation : centre de formation et agence web",
  "url": "https://www.internet-formation.fr",
  "logo": "https://www.internet-formation.fr/img/logo-IF.png",
  "contactPoint": [
    {
      "telephone": "+336-76-34-29-91",
      "address": {}
    }
  ],
  "@context": "http://schema.org",
  "@type": "WebSite",
  "name": "Internet-Formation : centre de formation et agence web",
  "alternateName": "Internet-Formation",
  "url": "https://www.internet-formation.fr",
  "@context": "http://schema.org",
  "@type": "Person",
  "name": "Mathieu Chartier",
  "jobTitle": "Consultant web et webmaster",
  "affiliation": "Internet-Formation",
  "sameAs": []
}
```

Data Interoperability and Semantics

</ Part 3. Data schemas and semantics >

ICM – Toolbox Engineering and Interoperability of Software Systems – Course unit on Data Interoperability and Semantics

M1 Cyber Physical and Social Systems – Course unit on Data Interoperability and Semantics

Maxime Lefrançois <https://maxime-lefrancois.info>

Course unit URL: <https://ci.mines-stetienne.fr/cps2/course/data>