What is schema.org?

A Cetis Briefing Paper for LRMI.

By Phil Barker and Lorna M. Campbell

Schema.org is a joint initiative of the search engines Google, Bing, Yahoo and Yandex aimed at making it easier to index web pages in such a way that facilitates the building of sophisticated search services. Schema.org metadata may also be used for other applications e.g. in eBooks and as stand-alone metadata records.

This briefing describes schema.org for a technical audience. It is aimed at people who may want to implement schema.org markup in websites or other tools they build but who wish to know more about the technical approach behind schema.org and how to implement it. We also hope that this briefing will be useful to those who are evaluating whether to implement schema.org to meet the requirements of their own organization.

This briefing has been produced as part of the Learning Resource Metadata Initiative (LRMI), which is concerned with extending and applying schema.org to the description of educationally relevant properties of resources. Other briefings in this series will provide an in-depth overview of LRMI.

cetis
Centre for Educational Technology, Interoperability and Standards

What is Schema.org?

Schema.org is a joint initiative by Google, Yahoo, Microsoft Bing and the Russian search engine Yandex, which was launched in June 2011. The aim of the initiative is to help search engines to interpret information on web pages so that it can be used to improve the display of search results, making it easier for people to find the information they are looking for. To do this, content publishers insert machine readable information into the HTML of web pages that helps search engines understand the significance of the text on those pages. This information helps to identify which text is the title, which is the author's name, which link is to the publisher, etc. In other words, it allows human readable resource descriptions to double up as machine readable metadata, or what Google calls structured data.

Schema.org has two components:

- An ontology, i.e. a vocabulary for naming the types and characteristics of resources, their relationships with each other, and constraints on how to describe these characteristics and relationships.
- The expression of this information in machine readable formats such as microdata, RDFa Lite and JSON-LD.

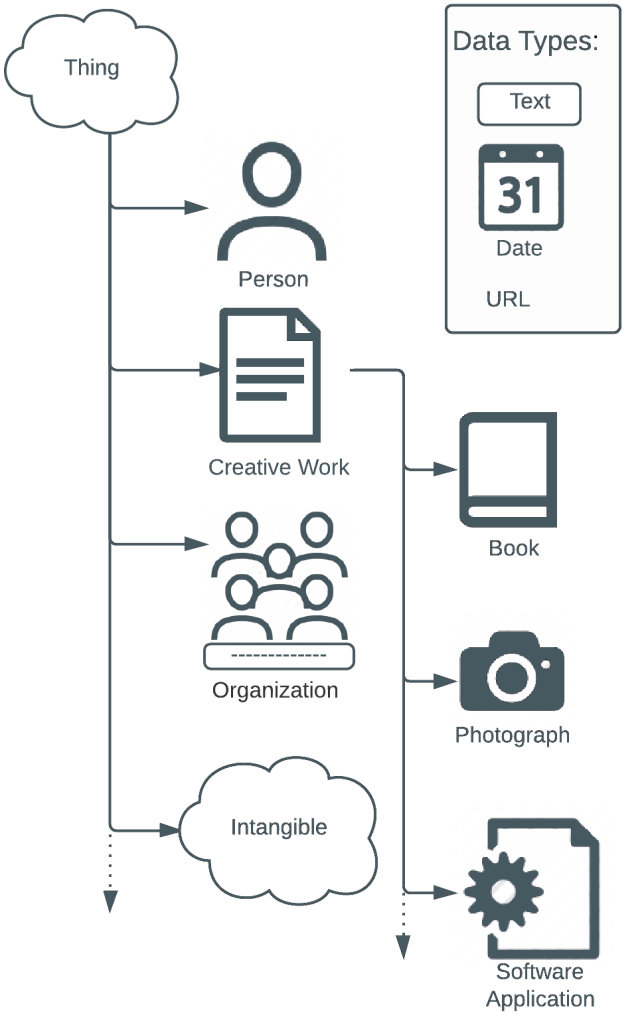


***Figure 1**. A small sample of the schema.org herarchy of types*

This Briefing follows the terminology used by schema.org microdata and refers to the resource being described as an item. The first step in describing an item is to classify it as a specific type of resource. This identifies what type of thing the item is, and for each type schema.org defines a set of properties that may be used to provide descriptions of the characteristics of the item. The schema.org documentation lists a hierarchical set of types and their properties. The top level of the hierarchy, the most generic type, is `Thing`, subtypes of this include `CreativeWork`, `Event`, `Intangible`, `Organization`, `Person`, `Place` and `Product`. Subtypes that are of particular interest to education include `ScholarlyArticle`, `Book`, `Review` and `WebPage` (all subtypes of `CreativeWork`), `EducationEvent` and `EducationalOrganization` (subtypes of `Event` and `Organization` respectively). Each type in the hierarchy may have its own set of properties, but properties are also inherited from the parent type. The main properties of the generic Thing, which are inherited by all other types, are

- **`description`** (Text): a short description of the item
- **`image`** (Url): URL of an image of the item
- **`name`** (Text): the name of the item
- **`url`** (Url): URL of the item

The text in parenthesis above indicates the expected type for the data provided

1. http://schema.org/docs/schemas.html

about that property (in terms of RDF Schema it is the range of the predicate), that is, the description and name can be free text, whereas the others must be URLs.

So, the `CreativeWork` type inherits all the properties of `Thing` listed above and provides about forty more, including

- **`about`** (Thing): the subject matter of the content
- **`author`** (Person or Organization): the author of this content
- **`dateCreated`** (Date): the date on which the CreativeWork was created
- **`publisher`** (Organization): the publisher of the creative work

Note that, with the exception of date (which should be in ISO 8601 date format), the information about these properties should all be provided as embedded items of other schema.org types, an example is provided below. In fact schema.org is designed to be forgiving to the extent that if information about the author is provided as a plain text string, an attempt should be made to interpret it.

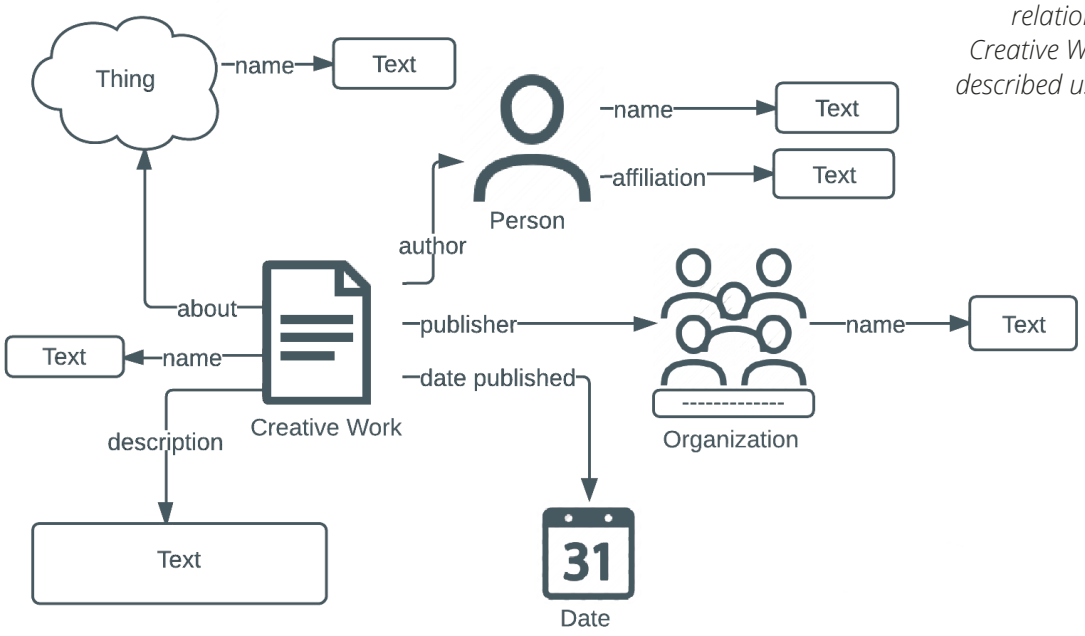


Figure 2. *Some of the relationships around a Creative Work that may be described using schema.org*

Uses of schema.org metadata

If schema.org metadata is available any search engine may use it to improve their search interface, for example:

- by distinguishing between different things with the same name (e.g. a book, a film and a game);
- by allowing the most relevant information to be displayed more prominently on the results page;
- by enabling results to be filtered by properties such as price, supplier or publication date;
- by providing links to more results about the same subject or from the same publisher.

An example of such an interface can be seen on Google Shopping, as illustrated in figure 3. (We must note that Google probably gets the data for creating such interfaces from many sources not just schema.org metadata.)

The metadata can also be used in the Google Custom Search Engine to build specialized search services based on schema.org types and properties.

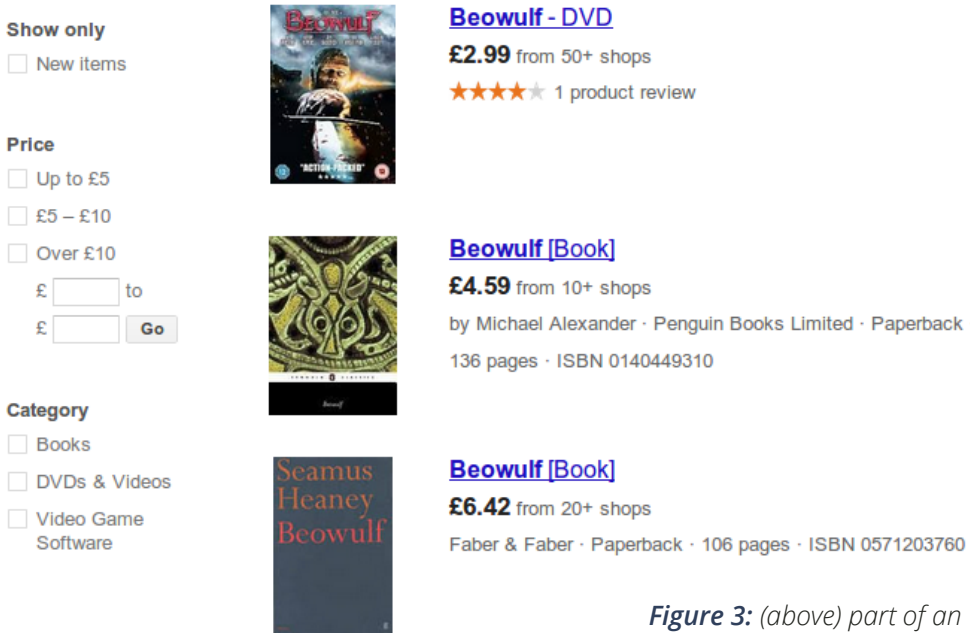
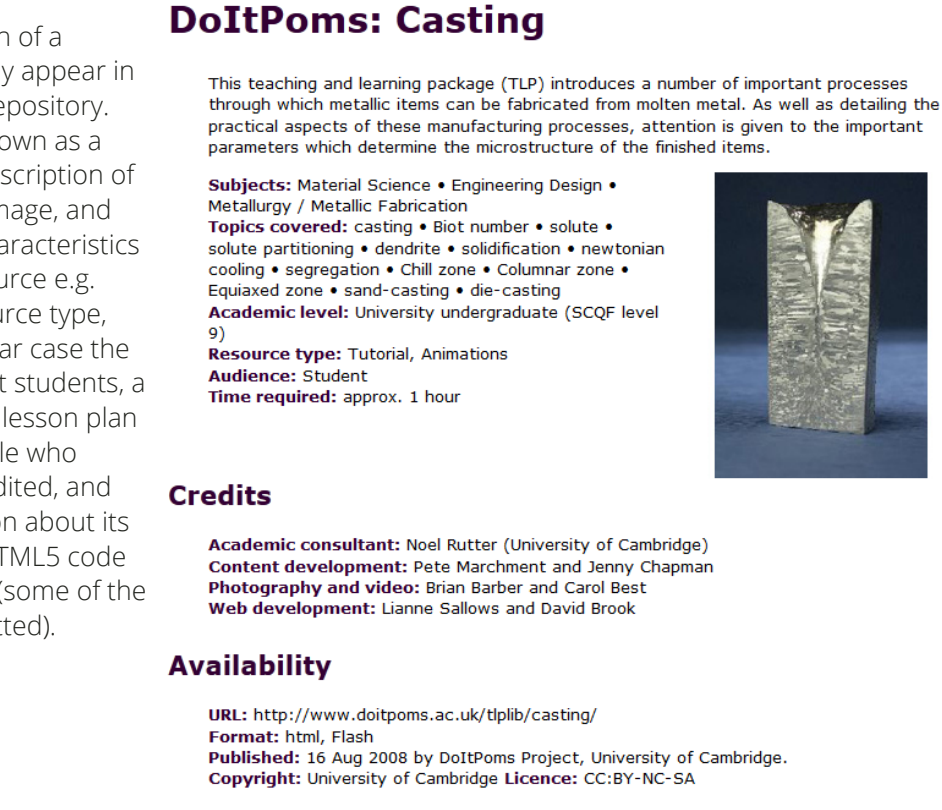


Figure 3: *(above) part of an enhanced search results page from Google*

Example

Most commonly schema.org metadata is encoded in HTML pages as microdata, which adds attributes to the HTML elements of the web page.

Figure 4 (right) is a description of a learning resource such as may appear in a catalogue listing or digital repository. The title of the resource is shown as a section heading, there is a description of the resource, an illustrative image, and information about various characteristics and relationships of the resource e.g. subject, academic level, resource type, audience, etc. (In this particular case the resource is a tutorial aimed at students, a different example might be a lesson plan aimed at teachers.) The people who created the resource are credited, and legal and technical information about its availability is provided. The HTML5 code for this page is shown below (some of the styling markup has been omitted).



DoItPoms: Casting

This teaching and learning package (TLP) introduces a number of important processes through which metallic items can be fabricated from molten metal. As well as detailing the practical aspects of these manufacturing processes, attention is given to the important parameters which determine the microstructure of the finished items.

Subjects: Material Science • Engineering Design • Metallurgy / Metallic Fabrication
Topics covered: casting • Biot number • solute • solute partitioning • dendrite • solidification • newtonian cooling • segregation • Chill zone • Columnar zone • Equiaxed zone • sand-casting • die-casting
Academic level: University undergraduate (SCQF level 9)
Resource type: Tutorial, Animations
Audience: Student
Time required: approx. 1 hour

Credits

Academic consultant: Noel Rutter (University of Cambridge)
Content development: Pete Marchment and Jenny Chapman
Photography and video: Brian Barber and Carol Best
Web development: Lianne Sallows and David Brook

Availability

URL: http://www.doitpoms.ac.uk/tlplib/casting/
Format: html, Flash
Published: 16 Aug 2008 by DoItPoms Project, University of Cambridge.
Copyright: University of Cambridge **Licence:** CC:BY-NC-SA

Figure 4: *(above) from a web page describing a learning resource about metal casting.*

```
<section>
    <header>
        <h1>DoItPoms: Casting</h1>
    </header>
    <p>This teaching and learning package (TLP) introduces a number of important processes through which metallic items can be fabricated from molten metal. As well as detailing the practical aspects of these manufacturing processes, attention is given to the important parameters which determine the microstructure of the finished items.</p>
    <img alt="picture of some cast aluminium" src="image.jpg">
    <p><strong>Subjects:</strong> Material Science • Engineering Design • Metallurgy / Metallic Fabrication
         <br /><strong>Topics covered:</strong> casting • Biot number • solute • solute partitioning • dendrite • solidification • newtonian cooling • segregation • Chill zone • Columnar zone • Equiaxed zone • sand-casting • die-casting
         <br /><strong>Academic level:</strong> University undergraduate (SCQF level 9)
         <br /><strong>Resource type:</strong> Tutorial, Animations
         <br /><strong>Audience:</strong> Student
         <br /><strong>Time required:</strong> approx. 1 hour
    </p>
    <h2>Credits</h2>
    <p><strong>Academic consultant:</strong>
                Noel Rutter (University of Cambridge)<br />
       <strong>Content development:</strong>
                Pete Marchment and Jenny Chapman<br/>
       <strong>Photography and video:</strong> Brian Barber and Carol Best<br />
       <strong>Web development:</strong> Lianne Sallows and David Brook<br />
    </p>
    <h2>Availability</h2>
    <p><strong>URL:</strong>
       <a href="http://www.doitpoms.ac.uk/tlplib/casting/">
            http://www.doitpoms.ac.uk/tlplib/casting/
       </a>
       <br /><strong>Format:</strong> html, Flash
       <br /><strong>Published:</strong> <time datetime="2008-08-16">16 Aug 2008</time> by DoItPoms Project, University of Cambridge.
       <br /><strong>Copyright:</strong> University of Cambridge
       <strong>Licence:</strong> CC:BY-NC-SA
    </p>
</section>
```

The whole description is wrapped in a `section` tag. To indicate in a machine readable way that this section provides a description of a single item, we may insert the microdata `itemscope` attribute, and we may use the `itemtype` attribute to specify the type of the item. In order to describe this example, probably the most appropriate type from the schema.org ontology to use as the value for `itemtype` is `WebApplication`, which allows us to specify the browser requirements of the resource. If the first HTML element is

```
<section    itemscope
            itemtype="http://schema.org/WebApplication">
```

Everything enclosed in this section will be interpreted as pertaining (directly or indirectly) to the description of the same resource, descriptions of other resources may be provided elsewhere on the page.

The microdata `itemprop` attribute is used to indicate which properties of the item are being described by the HTML. So to indicate that the text of the heading in line 3 is the title of this resource, we add the attribute `itemprop="name"` to the `h1` element.

```
3.  <h1 itemprop="name">DoItPoms: Casting</h1>
```

Similarly `itemprop="description"` can be added to the `p` tag on line 5. Adding `itemprop="image"` to the `img` tag will indicate that the URL specified by the `src` attribute of this tag is the URL for an image of this item; adding `itemprop="url"` to the `a` tag on line 25 marks the `href url` as the URL of the resource. Note, this may be (as it is in this case) a URL different to that of the page that carries the description.

We may want to indicate that the resource is about Materials Science, Engineering Design and Metallurgy / Metallic Fabrication (see line 7). The relevant item property is "about" but there is no html tag to add `itemprop="about"` into. The answer is to create one by inserting `span` elements: there are three distinct subjects and so we need three `span` elements:

```
7.   <p><strong>Subjects:</strong> <span itemprop="about">Material Science</span>
       • <span itemprop="about">Engineering Design</span> • <span
       itemprop="about">Metallurgy / Metallic Fabrication</span>
```

Note: the schema.org specification states that the `about` property is expected to be an item of type `Thing`. That means we should mark up each subject as the name of a `Thing` or some subtype of `Thing`, in this case it makes sense to class subject as `Intangible` for example:

```
<span itemprop="about"
       itemscope itemtype="http://schema.org/Intangible">
    <span itemprop="name">Material Science</span>
</span>
```

This is shown in figure 5, but in the example above we have made use of the short cut described in the the schema.org documentation:

> *We also expect that often, where we expect a property value of type Person, Place, Organization or some other subClassOf Thing, we will get a text string. In the spirit of "some data is better than none", we will accept this markup and do the best we can.*
>
> http://schema.org/docs/datamodel.html

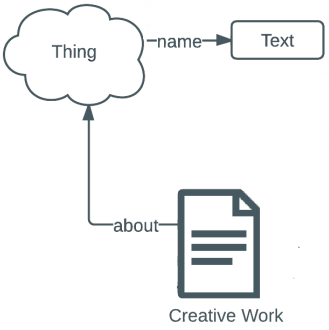


Figure 5: *a creative work about a thing with a name*

We have text for both the name and affiliation of the Academic Consultant who contributed to this resource (see line 17), so it makes sense to provide these as properties of an embedded item. This is achieved by adding an `itemscope` attribute with associated `itemtype="http://schema.org/Person"` to an enclosing HTML element as well as the `itemprop="contributor"` attribute. This creates a new item for describing a person, and the names and affiliation of this person can be provided:

```
15.  <strong>Academic consultant:</strong>
     <span itemprop="contributor"
           itemscope itemtype="http://schema.org/Person">
         <span itemprop="name">Noel Rutter</span>
        (<span itemprop="affiliation">
               University of Cambridge
         </span>)
     </span><br />
```

Note that the same value of `itemprop` now occurs in two places: i.e. name on lines 3 and 34, however these will be interpreted as the names of different items since properties pertain to the item of the most immediate `itemscope` in which they are enclosed. Many other parts of the text may be marked up to indicate the property they describe.

To test the result, we can use Google's Structured Data Testing Tool, which for this example shows the following information that has been extracted from the microdata (many subjects and contributors have been omitted):

Item
type: http://schema.org/webapplication
property:
name: DoItPoms: Casting
description: This teaching and learning package (TLP) introduces a number of important processes through which metallic items can be fabricated from molten metal. As well as detailing the practical aspects...
about: Material Science
about: Engineering Design
about: Metallurgy / Metallic Fabrication
...
contributor: ***Item 1***
...
url: http://www.doitpoms.ac.uk/tlplib/casting/
fileformat: text/html
fileformat: application/x-shockwave-flash
datepublished: 2008-08-16
publisher: DoItPoms Project, University of Cambridge
copyrightholder: University of Cambridge

Item 1
type: http://schema.org/person
property:
name: Noel Rutter
affiliation: University of Cambridge

Schema.org as JSON-LD

Thus far we have discussed the use of the schema.org vocabulary to mark up webpages using microdata, though the same can be achieved using the RDFa syntax. Using either of these syntaxes to embed metadata in web pages makes sense where the information is primarily being delivered as human readable documents, with schema.org being used to help with the computer-facilitated discovery of these documents. However, at times when such information is being transmitted from one computer to another it is not necessary for it to be human readable. Examples would include creating a catalogue by harvesting metadata from several sources and providing metadata for programmatic access via an API. JSON (JavaScript Object Notation), and specifically JSON-LD (JSON for Linking data) is a format that is eminently suited to such use[2]. Google supports JSON-LD for

2. http://json-ld.org/spec/latest/

provision of schema.org metadata[3] when embedded in webpages. Some services go further and allow schema.org metadata to be provided as stand alone JSON-LD documents, an example of such a service is the Learning Registry[4]. The Learning Registry is a technical infrastructure for capturing, connecting and sharing data about learning resources available online. One advantage of storing schema.org metadata in stand-alone JSON documents is that it means that metadata may be provided by people other other than the maintainers of web pages about the resource being described.

Trust, reliability and visibility

In the support document for their early work on schema.org metadata, then known as rich snippets, Google said this about hidden metadata:

> *In general, Google won't display any content in rich snippets that is not visible to [a] human user.*
>
> https://support.google.com/webmasters/answer/1093493#hidden

One of the potential advantages of the schema.org approach of marking up visible information as metadata over other ways of providing metadata is related to the trustworthyness and reliability of the data. Experience with early alternatives for providing metadata about web pages such as embedded <meta> tags in the header containing or linking to information intended for use by search applications suggested that such approaches yielded poor results. Sometimes the information provided was inaccurate due to deliberate attempts to mislead the search engines with data that suggested the web page was relevant to popular subjects; sometimes it was simply that the author of the content of a page had started with a template and hadn't changed the metadata in the header to match the content. The developer of one of the larger vocabularies to be incorporated into schema.org describes Google's (and other search engines') likely rationale for mistrusting hidden metadata this way[5]:

1. invisible markup invites spammers that try to manipulate the search engine,
2. a link to human-readable content allows [the combination of] structured data and the textual content for information extraction heuristics, and
3. the data quality is likely higher for visible content (since humans will complain otherwise).

> Martin Hepp, "JSON-LD: Finally, Google Honors Invisible Data for SEO"

There are some disadvantages to providing data using schema.org that stem from the constraint of marking up the text that is displayed. There may be a valid clash between the data that you want to provide a search application and what you want to display to users of your website. For example you may want to provide a URL to identify a resource unambiguously but without presenting users with a distracting link. The schema.org help documentation suggests the use of <link> or <meta> tags to embed this information but not to display it (see note below), with the caveat

> *This technique should be used sparingly. Only use meta with content for information that cannot otherwise be marked up.*
>
> http://schema.org/docs/gs.html#advanced_missing

3. https://support.google.com/webmasters/answer/4620709
4. http://learningregistry.org/
5. http://blog.heppresearch.com/2014/03/24/json-ld-finally-google-honors-invisible-data-for-seo/

Note: Google's guidelines on using `<link>` and `<meta>` tags to provide alternative machine readable values.

Use the `<link>` element to provide a URL to identify a resource but not to show a distracting link:

```
<div itemscope itemtype="http://schema.org/Book">
  <span itemprop="name">The Catcher in the Rye</span>—
  <link itemprop="url" href="http://en.wikipedia.org/wiki/The_Catcher_in_the_Rye" />
  by <span itemprop="author">J.D. Salinger</span>
</div>
```

Use the `<meta>` tag to provide information that is implicit in the content of the page

```
 <div itemprop="reviews"
      itemscope itemtype="http://schema.org/AggregateRating">
    <img src="four-stars.jpg" />
    <meta itemprop="ratingValue" content="4" />
    <meta itemprop="bestRating" content="5" />
    Based on <span itemprop="ratingCount">25</span> user ratings
  </div>
```

More subtly, putting metadata within the tree of elements that make up an HTML document places that information into a hierarchical structure which contrasts with the more flexible graphs that semantic web languages such as RDF allow. In the post quoted above, Martin Hepp describes this as "violating the principle of 'separation of concerns' – you have to align a given HTML tree structure with a given data structure, dictated by the schema.org ontology". Thus, as in the simple example provided above, all the information about the person who is a contributor to a creative work all sits within an HTML element that specifies that the information relates to the contributor property of that work. This becomes problematic when one wants to provide extensive detailed information about items that are used to give values for properties such as contributor, especially where the same information is repeated. Imagine four people have contributed to the resource and all four work for the same organisation: the nesting hierarchy means that all the information about the organisation has to be repeated for each person. Furthermore, imagine that some of these people have contributed to several resources described on the same page: the amount of redundant repetition of data multiplies. Provided an item (in this case the person who made the contribution or their affiliation) is uniquely identified a search application *may* aggregate information about that item from several sources, in which case there would be no need to repeat the information in the document—but equally, they may not. There are mechanisms for jumping out of the nested hierarchy, for example the `itemref` property in microdata allows links to be made between items described in separate sections on a webpage[6], however it is unclear how well this is supported within schema.org. As Martin Hepp points out in the blog post quoted above, the recent adoption by Google of a JSON-LD serialization of schema.org metadata that may be embedded within a webpage without being displayed is a step away from the insistence that the data marked-up with schema.org should be visible to humans. However the conditions under which Google will trust this metadata are not known.

6. http://www.w3.org/TR/microdata/#attr-itemref

Conclusion

The schema.org sponsors hope their effort will result in simplicity for those wishing to provide data. The schema.org front page claims:

> *A shared markup vocabulary makes it easier for webmasters to decide on a markup schema and get the maximum benefit for their efforts. So, in the spirit of sitemaps.org, search engines have come together to provide a shared collection of schemas that webmasters can use.*
>
> http://schema.org/

The aim is to have a single source where a web developer can find all the documentation required to provide information for the big search engines. In principle, the success of schema.org can be judged by two factors: the number of websites providing metadata using schema.org markup and the services provided by search engines that exploit this metadata. Neither of these is easy to judge from outwith the search engine companies, however the indications are positive.

Large scale surveys of the web are difficult, and the results for a new initiative with as much backing as schema.org are likely to date rapidly. A survey by Bizer *et al*[7] based on data from 40 million websites in 2012, a few months after schema.org's launch, found that schema.org was used by about 30% of the top 1000 sites. In November 2013 Google were finding schema.org structured data in over 5 million websites[8].

It is equally difficult to know what use Google and other big search engines are making of schema.org metadata. What is clear is that any use will be alongside metadata taken from other sources, e.g. the Google Knowledge Graph, data extraction from web crawling and book digitization. The example from Google shopping above illustrates how this information might be presented. Their continuing development of schema.org, for example of the JSON-LD serialization and their use of it in tools such as the Custom Search Engine[9], indicates that Google continue to see value in it.

There is a chicken and egg element to these two indicators: web developers will only put more effort into implementing schema.org if they see big search engines using it; big search engines will only use it if they see many websites implementing it. The process can be bootstrapped on both sides. Adoption by providers is eased by support of schema.org by content management systems such as WordPress and Drupal, and extensions providing this support are being developed, as are several stand alone tag generators. In addition, services that demonstrate the value of implementing schema.org can becreated using tools such as Google Custom Search Engine.

7 http://link.springer.com/chapter/10.1007%2F978-3-642-41338-4_2 (pdf at http://dws.informatik.uni-mannheim.de/fileadmin/lehrstuehle/ki/pub/Bizer-etal-DeploymentRDFaMicrodataMicroformats-ISWC-InUse-2013.pdf)

8 http://semanticweb.com/schema-org-chat-googles-r-v-guha_b40607

9 https://support.google.com/customsearch/answer/4544182?hl=en

Links and Resources

The GetSchema.org wiki provides links to many useful resources and tools relevant to schema.org. A few are listed below, starting with the relevant specification documents.

Further Information: schema.org and related specs

The normative documents for various specifications and standards related to schema.org are listed below with an indication of their formal status.

Schema.org

Home page http://schema.org/
The schema hierarchy of types http://schema.org/docs/schemas.html
Getting started with schema.org http://schema.org/docs/gs.html

HTML5

W3C Candidate Recommendation: http://www.w3.org/TR/html5/

HTML Microdata

W3C Working Group Note: http://www.w3.org/TR/microdata/

RDFa Lite 1.1

W3C Recommendation: http://www.w3.org/TR/2012/REC-rdfa-lite-20120607/

JSON JavaScript Object Notation)

Introducing JSON: http://json.org/

JSON-LD (JSON for Linking Data)

Home page: http://json-ld.org/

Tools that help create schema.org structured data

Several tools exist that will assist in the creation of schema.org structured data, these are useful for a variety or purposes from producing illustrative snippets to managing websites with automatically generated schema.org markup embedded.

Google's Structured Data Markup Helper

https://support.google.com/webmasters/answer/3069489?hl=en

Microdata Generator

http://microdatagenerator.org/

Schema-creator by Raven

Standalone web form http://schema-creator.org/
WordPress plugin http://schema-creator.org/wordpress.php

Other WordPress plugins

http://wordpress.org/plugins/tags/schemaorg

Drupal schema.org module

https://drupal.org/project/schemaorg

Tools that check, display or use schema.org structured data

Google Structured Data Testing Tool
http://www.google.com/webmasters/tools/richsnippets

Yandex Structured data validator
http://webmaster.yandex.com/microtest.xml

Bing's Markup Validator
http://www.bing.com/toolbox/markup-validator

SEO moves microdata parser
http://tools.seomoves.org/microdata/

GetSchema.org Microdata to RDF Extractor
http://getschema.org/microdataextractor/about

Microdata.reveal Chrome browser plugin
https://chrome.google.com/webstore/detail/microdatareveal/olapakiakkblfdaajcifgldandnikpdh

Google Custom Search
https://developers.google.com/custom-search/

About this briefing

Title: What is Schema.org?
Authors: Phil Barker and Lorna M. Campbell
Date: 05/06/2014 (revised 06/06/2014)
URI: http://publications.cetis.ac.uk/2014/960

This briefing was produced by Cetis for LRMI under contract to Creative Commons.

About LRMI

The Learning Resource Metadata Initiative is funded by the Bill & Melinda Gates Foundation, and jointly lead by Creative Commons and the Association of Educational Publishers—now the 501(c)(3) arm of the Association of American Publishers—with the aim of making it easier to publish, discover, and deliver high quality educational resources on the web. With input from a wide range of organisations, from both the open and commercial spheres, involved in publishing and using educational resource LRMI successfully proposed additions to schema.org (an initiative of Google, Yahoo and Bing) allowing the description of educationally important properties of resources to be marked-up in web pages in a manner that is easily understood by search engines. This enables people to create search engines that support the filtering search results based on criteria such as their match to a specific part of a curriculum, or the age of the students, or one of several other characteristics.

http://www.lrmi.net/

About Cetis

Cetis is the Centre for Educational Technology, Interoperability and Standards. Our staff are globally recognised as leading experts on education technology innovation, interoperability and technology standards. For over a decade Cetis has provided impartial strategic, technical and pedagogical advice on educational technology and standards to funding bodies, standards agencies, government, institutions and commercial partners.

Cetis are active in the development and implementation of open standards and have been instrumental in developing and promoting the adoption of technology and standards for course advertising, open education resources, assessment, and student data management, opening new markets and creating opportunities for innovation. Our work includes a wide range of activities from representation at national standardisation bodies, facilitation of online and face-to-face events to production of a range of formal and informal publications.

http://www.cetis.ac.uk