

Metastyle: Widgetising Sharable Stylesheets via Linked Data

Xi Bai¹ and Armin Haller²

¹ BBC Media Services, Future Media, London W12 7TP, UK

² Australian National University, CSIRO ICT Centre, Canberra ACT 2601, Australia
`xi.bai@bbc.co.uk`, `armin.haller@csiro.au`

Abstract. Several formats for embedding metadata into Web pages have been proposed and assisting publishers in semantically enhancing their content. However, little work has been done to explore the potential of those formats to benefit Web page design and styling. On the other hand, due to different employed markup mechanisms, a stylesheet composed with a specific embeddable data format cannot be applied to another without tedious and time-consuming manually tailoring of selectors. This paper proposes an approach to helping users to generate and share so-called micro-themes with semantic attribute selectors, via which themes can be widgetised so as to be more readable and reusable compared with those using traditional selectors, thanks to unambiguous Linked Data URIs. Metastyle is a proof-of-concept implementation of this approach and it enables micro-themes, containing annotations derived from specific vocabularies, to be conveniently shared with users having their pages enhanced based on those vocabularies or subsets thereof.

1 Introduction

HTML embeddable data formats such as Microformats [9], RDFa [1] and Microdata [7] have helped publishers or webmasters to semantically enhance their online content with Web-compatible annotations, the number of which has been increased, driven mainly by early industrial adoptions from Yahoo! SearchMonkey, Google Rich Snippets and Facebook Open Graph. At the time of writing, *schema.org*¹, as a joint initiative by search engine vendors, has been used across over four million Internet domains [4]. Embeddable data formats mostly employ HTML attributes as anchors from where metadata are injected. For instance, RDFa uses commonly found (X)HTML attributes and also introduces a few new others, while Microdata invented a set of new attributes to allow publishers to annotate item properties as name-value pairs. Traditionally, HTML attributes such as `@class` and `@id` are often used as CSS selectors for styling HTML DOM elements. So far little attention has been paid to using metadata embedded via attributes as CSS selectors. Compared with the traditional way of selecting and styling HTML DOM elements, semantic attribute selectors have the

¹ <http://schema.org/>

advantage in targeting elements at the entity level based on the incorporated semantics (e.g., a person or a car, etc.) so as to be unambiguous and reusable. On the other hand, existing metadata-driven page styling approaches are still limited in reusability and hamper theme sharing since each theme normally targets a particular embeddable data format and cannot be applied to another format without proper human interventions, which are usually tedious and time-consuming jobs. In this paper, we explore the potential of harnessing embedded metadata to select HTML DOM elements within the page styling process and provide Web content publishers with a tool — Metastyle, which takes a specific vocabulary in RDF as a seed and generates a skeleton helping to create a so-called micro-theme dedicated to one or more Web entities and shared as widgets. A *micro-theme* is a stylesheet exclusively targeting a specific Web entity/resource. Moreover, Metastyle exposes an oEmbed² endpoint (`/oembed` generates responses in both JSON and XML based on URI extensions, the *format* parameter and the Accept header) and each shared micro-theme widget can be therefore embedded into oEmbed-aware clients easily and rendered through the widget URI. Micro-themes can be shared as templates and also discoverable via Metastyle. Statistics and ranking are provided as well, making better-designed micro-themes more findable than those with lower qualities. With respect to embeddable data format variations, Metastyle provides one-click transformation between micro-themes created based on different formats.

2 Metastyle At a Glance

Figure 1 depicts the framework of Metastyle which contains components targeting various functionalities covering several operations related to micro-themes.



Fig. 1. Metastyle Framework

² <http://oembed.com/>

Micro-Theme Editor allows users to create and customise micro-themes. There are four discovery (sub-)components (top-left) including searching, the gallery, theme grouping and top lists as well as a *Ranking* component to assist users in finding interesting micro-theme widgets. Each micro-theme widget has a view (generated by *Theme Viewer*) from which users can check statistics or conduct transformations on embeddable data formats (detailed in Section 4).

3 Micro-Theme Generation and Customisation

The semantic attribute selectors contained in the final micro-theme can be automatically generated as a skeleton by *Metastyle* based on the input vocabulary, the targeted resource type, the embeddable data format and the prefix (for RDFa only). Currently, HTML5 Microdata and RDFa Lite [8] are supported by *Metastyle* and alternative views for users to edit CSS by harnessing dynamic stylesheet languages are provided. *Sass*³ and *LESS*⁴ are supported and thanks to the features (e.g., variables, nesting, mixings, operators and functions, etc.) of those CSS preprocessors, micro-themes have become more compact and easier to compose and share as widgets, compared with those materialised in standard CSS. Moreover, compared with CSS rules with the flat structure, both *Sass* and *LESS* support nested selectors via which users can create more succinct and meaningful hierarchical styling rules corresponding to hierarchical annotations (e.g., nested `@itemscopes` or `@typeofs`) inside HTML pages. *Metastyle* follows the PREFIX.TYPE-PROPERTY convention to name selector variables (e.g., `schema.Person-name`) to avoid possible selector clash during micro-theme importing. Here, PREFIX stands for the prefix of a vocabulary namespace URI. *Metastyle* currently generates Microdata-based skeletons using `@itemscope`, `@itemtype` and `@itemprop`. Apart from them, `@itemid` must be used together with `@itemscope` and `@itemtype` [7], and *Metastyle* does not need to use `@itemid` in micro-theme skeletons since the latter two are already present in each selector. The Micro-theme skeleton generation is a vocabulary-driven process and at present, *Metastyle* is agnostic of the DOMs to which publishers will apply their final micro-themes. Thus, how to employ `@itemref` has not been explored but its potential will be looked at in future work. With a screenshot, Figure 2 illustrates the process of customising and sharing a generated micro-theme skeleton based on the exemplary markup of *Person* on *schema.org*.

4 Ranking, Searching and Reusability

Metastyle employs several ranking strategies for promoting the most interesting micro-theme widgets on both the entry page and the search result page. Rankings are supported according to published time, views and votes to help users to find desired micro-themes.

³ <http://sass-lang.com/>

⁴ <http://lesscss.org/>

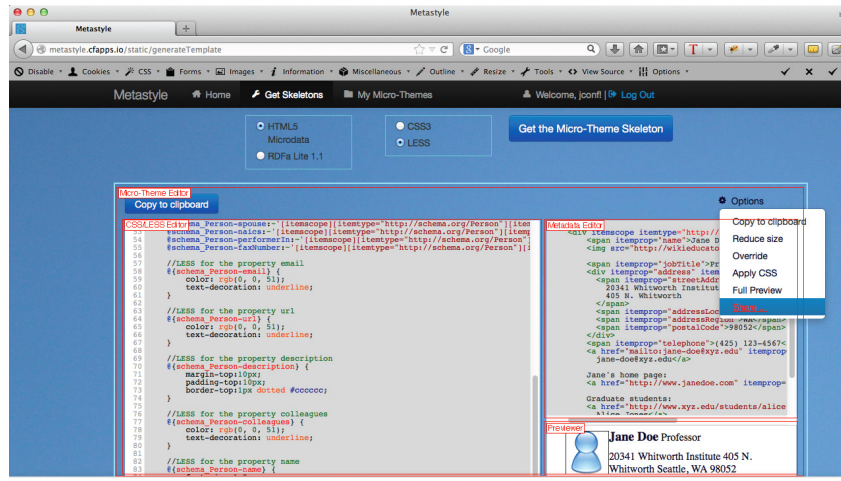


Fig. 2. Screenshot on the micro-theme skeleton generation and customisation

Suppose a user happens to have his/her homepage annotated using the *schema.org* vocabulary via Microdata and the searched micro-theme is however based on another embeddable format (e.g., RDFa Lite). Then the question is how can the user reuse it by simply doing a “lift-and-shift”? Here, Metastyle provides a one-click transformation solution targeting Microdata and RDFa Lite. We proposed a property-oriented approach to utilising the micro-theme transformation based on automatically generated combined-attribute selectors, each of which ends with an attribute taking an RDF statement’s predicate as its value. Note that each predicate can be attached to an attribute on the same HTML element where the entity (subject) is declared or on one of its descendants, and for Microdata-specific attributes, `@itemtype` should be used on the same element where `@itemscope` is declared [7]. In our approach, each selector therefore has one of the following formatting options (*T* stands for the an entity type; *P* stands for an entity property; `_` stands for a single whitespace character):

Entity Selector: `[itemscope][itemtype="TURI"]`
Property Selector: `[itemscope][itemtype="TURI"][itemprop="PURI"] |`
`[itemscope][itemtype="TURI"]_ [itemprop="PURI"]`

For RDFa, more care needs to be taken when dealing with vocabulary namespaces due to the possibly used `@prefix` and `@vocab` attributes, and thus, each selector will have one of the following formation options (*PRE* stands for the abbreviation of a specific namespace):

Entity Selector: `[typeof="PRE:TNAME"] |[typeof="TNAME"] |[typeof="TURI"]`
Property Selector: `[typeof="PRE:TNAME"] [property="PRE:PNAME"] |`
`[typeof="PRE:TNAME"]_ [property="PRE:PNAME"] |`
`[typeof="TNAME"] [property="PNAME"] |`
`[typeof="TNAME"]_ [property="PNAME"] |`
`[typeof="TURI"] [property="PURI"] |`
`[typeof="TURI"]_ [property="PURI"]`

Based on the above combined-attribute patterns for selectors, Metastyle established the *Selector Transformer* (see in Figure 1) with the duty of transforming one kind of selector formation to the other. For micro-themes in Sass or LESS (language detection can be done based on distinctive naming conventions on selector variables), their transformations will be fulfilled in a similar manner and only the names of selector variables will be transformed. Now the user can get an equivalent RDFa-based micro-theme from the one based on Microdata by simply clicking one button. With suggestions from Metastyle powered by *prefix.cc*⁵, he/she has a chance to choose “schema” as the proper prefix if needed by RDFa before the actual transformation occurs. Users also have an option to rollback to the original Microdata-based micro-theme as showcased in Figure 3.

```

18 margin-right: 20px;
19 border: 1px solid #333333;
20 width: 64px;
21 height: 40px;
22 }
23
24 /*style for the property homepage*/
25 [itemscope][itemtype="http://xmlns.com/foaf/0.1/Person"][itemprop="homepage"],
26 [itemscope][itemtype="http://xmlns.com/foaf/0.1/Person" [itemprop="homepage"],
27 [itemscope] [itemtype="http://xmlns.com/foaf/0.1/Person" [itemprop="homepage"],
28 [itemscope] [itemtype="http://xmlns.com/foaf/0.1/Person" [itemprop="homepage"] {
29   color: green;
30   text-decoration: underline;
31 }
32
33 /*style for the property homepage*/
34 [itemscope][itemtype="http://xmlns.com/foaf/0.1/Person" [itemprop="homepage"]:hover,
35 [itemscope][itemtype="http://xmlns.com/foaf/0.1/Person" [itemprop="homepage"]:hover,
36 [itemscope] [itemtype="http://xmlns.com/foaf/0.1/Person" [itemprop="homepage"]:hover,
37 [itemscope] [itemtype="http://xmlns.com/foaf/0.1/Person" [itemprop="homepage"]:hover {
38   color: green;
39   text-decoration: underline;
40 }
41
42 /*style for the property knows*/
43 [itemscope][itemtype="http://xmlns.com/foaf/0.1/Person" [itemprop="knows"],
44 [itemscope][itemtype="http://xmlns.com/foaf/0.1/Person" [itemprop="knows"],
45 [itemscope] [itemtype="http://xmlns.com/foaf/0.1/Person" [itemprop="knows"],

```

Copy to clipboard **Go back to CSS for Microdata**

Fig. 3. Micro-theme transformation

5 Related Work

A number of works on creating sharable themes (a *theme* is usually a bundle of highly coupled stylesheets, HTML documents and JavaScript documents if necessary) exist and some of them have been commercialised. A template called *microformats.css*⁶ was created and used for styling pages with hCard and hCalendar annotations in Microformats. Several other CSS templates have been developed for highlighting embedded Microformats (e.g., see in here⁷) and RDFa (e.g., see in here⁸). However, little attention has been paid to the theme shareability and none of these tools can provide a generic solution to address or mitigate the stylesheet repurposing problem. RDFaCE was invented for realising automatic content annotation (via, RDFa and Microdata) based on NLP outputs [6] and modularising/thematizing stylesheets has however not yet been fully investigated. Without being concerned about page styling, Callimachus [3] uses RDFa as a template language and page templates can be parsed into SPARQL

⁵ <http://prefix.cc/>

⁶ <http://www.mindgarden.de/downloads/microformats.css>

⁷ <http://www.hicksdesign.co.uk/journal/highlight-microformats-with-css>

⁸ <http://niklas1.github.com/rdfa-lab/css/rdfa.css>

queries whose final result bindings can be returned and propagated back to complete page construction. Other existing RDF-based tools such as ActiveR-aUL [5] can generate Web forms with embedded triples while their ways of applying stylesheets thereafter are *ad hoc*. Meanwhile, for existing metadata-driven styling methods, each of them usually targets a particular embeddable data format, and there is no easy way to (semi-)automatically migrate any of them to reuse themes on pages with other different formats with little or no human intervention.

6 Conclusions and Future Work

This paper showcases the potential of embeddable data formats to improve reusability and readability of stylesheets based on micro-themes targeting Web entities. Semantic attribute selectors used for constituting micro-themes can be automatically generated based on Web vocabularies/schemas in RDF and offered to publishers or webmasters as skeletons ready to customise and later share as widgets (also embeddable via oEmbed). Metastyle has been implemented as a service⁹ (with an introductory video clip¹⁰) and it will be further experimented with users having various degrees of knowledge about embeddable data formats and data models. Metastyle is being developed under our framework [2] which provides a one-stop solution to hypertext content publishing driven by embedded metadata derived from existing triples. Plugins (e.g., for WordPress and Drupal) are also under development to fulfil page representation missions thanks to existing support on metadata injection by several popular CMSs.

References

1. Adida, B., Birbeck, M., McCarron, S., Herman, I.: RDFa core 1.1, W3C recommendation. <http://www.w3.org/TR/rdfa-syntax/> (June 2012)
2. Bai, X., Haller, A., Klein, E., Robertson, D.: Metadata-driven hypertext content publishing and styling. In: Proceedings of WWW Companion'14. pp. 221–222. ACM (2014)
3. Battle, S., Wood, D., Leigh, J., Ruth, L.: The Callimachus project: RDFa as a web template language. In: Proceedings of COLD'12. CEUR-WS.org (2012)
4. Guha, R.V.: Light at the end of the tunnel. In: Proceedings of ISWC'13. pp. XI–XI. Springer (2013)
5. Haller, A., Groza, T.: A novel approach for interacting with Linked Open Data. In: Proceedings of WISE'11. pp. 338–340. Springer (2011)
6. Hellmann, S., Lehmann, J., Auer, S., Brümmer, M.: Integrating NLP using Linked Data. In: Proceedings of ISWC'13. pp. 98–113. Springer (2013)
7. Hickson, I.: HTML Microdata, W3C working draft. <http://www.w3.org/TR/microdata/> (October 2012)
8. Sporny, M.: RDFa Lite 1.1, W3C recommendation. <http://www.w3.org/TR/rdfa-lite/> (June 2012)
9. Suda, B.: Using Microformats. O'Reilly press (2006)

⁹ <http://metastyle.cfapps.io/>

¹⁰ <http://vimeo.com/baxtree/metastyle>